When Reasoning Meets Its Laws

Junyu Zhang [13*‡] **Yifan Sun** [1*] **Tianang Leng** [3*] **Jingyan Shen** [4*]
Liu Ziyin [25†] **Paul Pu Liang** [2†] **Huan Zhang** [1†]

[1]University of Illinois Urbana-Champaign [2]Massachusetts Institute of Technology
[3]University of Pennsylvania [4]New York University [5]NTT Research

Abstract

Despite the superior performance of Large Reasoning Models (LRMs), their reasoning behaviors are often counterintuitive, *e.g.*, excessive reasoning on simple questions or insufficient reasoning on complex ones, leading to suboptimal reasoning capabilities. This paper presents the Laws of Reasoning (LoRe), a unified framework that formalizes intrinsic reasoning patterns in ideal LRMs. We first propose *compute law* with the hypothesis that the reasoning compute should scale linearly with question complexity. Beyond compute, we extend LoRe with a supplementary *accuracy law*, positing exponential decay of accuracy with increasing complexity. Since the complexity is difficult to quantify in practice, we approximate these hypotheses via two tractable properties, *monotonicity* and *compositionality*. We therefore introduce LoRe-Bench, a benchmark that systematically measures these properties for large reasoning models. Evaluation shows that most reasoning models exhibit reasonable monotonicity but lack compositionality. In response, we develop an effective finetuning approach that enforces compute-law compositionality. Extensive empirical studies demonstrate that better compliance with compute laws yields consistently improved reasoning performance on multiple benchmarks, and uncovers synergistic gains across properties and laws. Project page: https://lore-project.github.io/.

1 Introduction

Large Reasoning Models (LRMs) such as OpenAI o1 (Jaech et al., 2024) have demonstrated unprecedented progress in approaching human-like reasoning capabilities. Despite their strong performance on solving complex problems, even powerful LRMs exhibit *abnormal behaviors* that deviate from typical human reasoning patterns. Humans generally adapt their thinking based on problem complexity (Newell et al., 1972). In contrast, as illustrated in Fig. 1, DeepSeek-R1 (Guo et al., 2025) tends to generate longer reasoning on a simpler sub-problem than on its complex composite question (1197 vs. 817 tokens), while achieving lower accuracy (38% vs. 50%)[1].

We also identify this unexpected phenomenon across a wide range of reasoning models, as shown in Fig. 5. This is primarily because prevalent training paradigms generally overlook the high variability of Chain-of-Thought (CoT) (Wei et al., 2022) data during the training phase. These CoT data are heuristically curated by human annotators or generated through online rollout (Schulman et al., 2017; Shao et al., 2024), rarely constrained by explicit principles, *e.g.*, how much thinking budget to allocate for a given problem (Wu et al., 2025). Hence, such approaches fail to guide models toward an optimal thinking strategy. It will lead to *inefficient allocation of computation*—either overthinking (Chen

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

*Equal contribution. †Equal mentorship. ‡Work partially done during the internship at MIT.
Correspondence to: junyuz6@engineering.upenn.edu, ppliang@mit.edu, huan@huan-zhang.com.
[1]For each problem, we generated multiple samples to account for randomness.

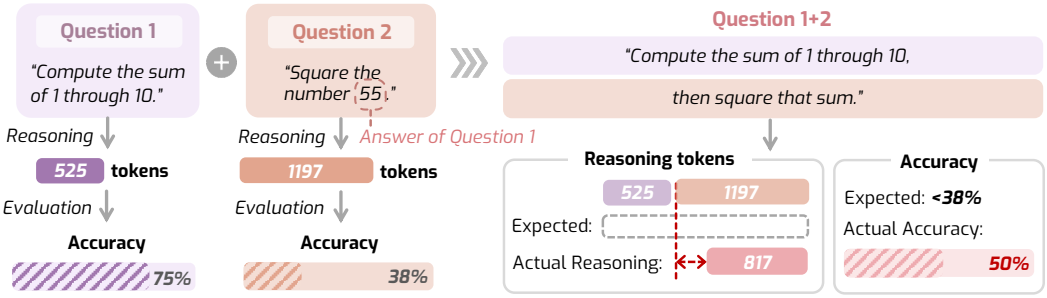


Figure 1: **Illustrative example with DeepSeek-R1** on (a) a summation question, (b) a squaring question, and (c) their composition ("sum, then square"). The model allocates ~300 *more* reasoning tokens to solve the squaring question than to the composite question, with a 12.5% *accuracy drop*.

et al., 2024b; Sui et al., 2025) or underthinking (Su et al., 2025; Wang et al., 2025), which in turn harms the performance (Stechly et al., 2024; Zhou et al., 2025a).

To overcome this limitation, one line of work focuses on adaptive *post-training* techniques, including supervised fine-tuning with variable-length CoT (Aggarwal & Welleck, 2025; Team et al., 2025). Another line of work modulates reasoning at *test time* (Muennighoff et al., 2025; Fan et al., 2025; Zhang et al., 2025b). While many attempts have been made to control reasoning, existing methods lack a unified theoretical framework for regulating optimal reasoning and still behave undesirably in our studies. Therefore, beyond empirical methods, several key challenges remain: **(Q1)** *Can we theoretically formalize model reasoning to ensure desirable behavior?* (Section 2) **(Q2)** *How can we evaluate whether popular LRMs follow these proposed principles?* (Section 3) **(Q3)** *Does enforcing these principles further improve general reasoning capabilities?* (Section 4, 5)

To fill this gap, we introduce **the Laws of Reasoning** (LoRe), which systematically formalize the relationship between complexity and optimal reasoning behaviors in *ideal* LRMs. The LoRe framework comprises a core *compute law* and a complementary *accuracy law*. Given the practical challenges of measuring these hypotheses, the two fundamental laws are approximated via two tractable properties of optimal reasoning models, monotonicity and compositionality.

We then evaluate whether current LRMs follow the laws by developing LoRe-Bench, **a comprehensive benchmark** that examines monotonicity and compositionality in LRMs. While LoRe-Mono is a synthetic benchmark with curated complexity orderings for evaluating monotonicity, LoRe-Compo is constructed from MATH500 (Lightman et al., 2023) to measure compositionality. Our results show that current models exhibit monotonicity but *lack compositionality*, even for strong baselines.

In response, we propose **a simple yet effective fine-tuning approach** to enforce the compute-law compositionality. From validation experiments, we present three key insights: (1) the compositionality of reasoning compute can be greatly improved with *simple fine-tuning* approach; (2) Enforcing compositionality *generally leads to better reasoning capability*; (3) *Synergistic effects* emerge, yielding broader improvements across different properties and laws.

2 The Laws of Reasoning

We introduce the Laws of Reasoning (LoRe), a unified framework that formalizes the relationship between question complexity and optimal reasoning behaviors for *ideal* models. Specifically, we focus on two key aspects, reasoning compute and accuracy, which are fundamental to understanding how models scale, generalize, and allocate computation when solving complex problems. Section 2.1 formulates the key concepts of reasoning. In Section 2.2, we present the central *compute law*, with a hypothesis that the reasoning budget should scale proportionally with complexity. In Section 2.3, we introduce the complementary *accuracy law*, which posits that overall accuracy should decay exponentially with increasing complexity. See Fig. 2 for an illustration of the framework.

2.1 Problem Formulation

Notation. Let $x \in \mathcal{X} \subseteq \mathcal{V}^*$ denote a question, where $\mathcal{V}^*$ is the space of finite-length sequences over a vocabulary $\mathcal{V}$. Let $M_\theta \in \mathcal{M}$ denote an *autoregressive* large reasoning model. LRMs adopt the *thinking-then-answering* paradigm (Guo et al., 2025; Abdin et al., 2025; Comanici et al., 2025), where

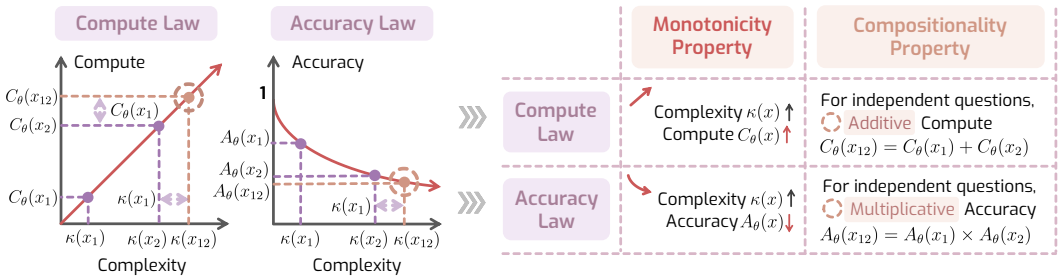


Figure 2: **Overview of the LoRe Framework.** We present the compute law with the complementary accuracy law: compute scales linearly ↗ and accuracy decays exponentially ↘ with question complexity. Our framework approximates these laws using two properties: *monotonicity* and *compositionality*. Specifically, for the compute law, monotonicity captures that more complex questions require more compute, while compositionality ◌ implies additive compute for independent questions.

the model M_θ first generates a reasoning chain $r \in \mathcal{R} \subseteq \mathcal{V}^*$ with probability $p_\theta(r \mid x)$ and then an answer $y \in \mathcal{Y} \subseteq \mathcal{V}^*$ with probability $p_\theta(y \mid x, r)$. We assume a fixed decoding strategy by default and denote the model's output by $o = (r, y) \in \mathcal{O} \subseteq \mathcal{V}^*$. We define the composition of two questions x_1 and x_2 as their concatenation with a connector prompt $c \in \mathcal{V}^*$[2], *i.e.*, $x_1 \oplus x_2 = \text{concat}(x_1, c, x_2)$.

Definition 1 (Complexity)**.** Let a *unit-cost primitive step* denote a single valid transition of a fixed deterministic Turing machine (Turing, 1937), and let τ be any finite sequence of primitive steps with length $\ell(\tau) \in \mathbb{N}$. Let $v(x, \tau) \in \{0, 1\}$ be a binary verifier that accepts (x, τ) if and only if τ is a valid solution sequence for x. The *complexity* of $x \in \mathcal{X}$ is

$$\kappa(x) \triangleq \min\{\, \ell(\tau) : \, v(x, \tau) = 1 \,\} \in \mathbb{N} \cup \{\infty\},$$

with $\kappa(x) = \infty$ if no valid solution sequence exists.

Unlike difficulty measured by empirical failure rates (Yu et al., 2026; Wen et al., 2025; Sun et al., 2026a) or estimated using psychometric models (Truong et al., 2025; Sun et al., 2026b; Zhou et al., 2026), our notion of complexity refers to the *minimal number of unit-cost primitive steps*, *i.e.*, the length of the shortest correct solution. Conceptually, $\kappa(x)$ can be well-defined via a binary verifier and a fixed deterministic Turing machine. However, computing $\kappa(x)$ is generally intractable, as verifying the *minimal* solution length requires a global search over a potentially exponential space.

For a given model, its test-time reasoning compute is directly proportional to the number of reasoning tokens generated. We therefore quantify reasoning compute as follows.

Definition 2 (Reasoning Compute)**.** The *reasoning compute* on question x is defined as the expected number of reasoning tokens generated by the model:

$$C_\theta(x) \triangleq \mathbb{E}_{r \sim p_\theta(\cdot|x)}[\ell(r)],$$

where $\ell(r)$ denotes the length (in tokens) of the reasoning chain r.

Definition 3 (Reasoning Accuracy)**.** The *reasoning accuracy* is defined as the probability that the model, when generating a reasoning chain and an answer given input x, produces a final answer that matches the ground truth. Formally,

$$A_\theta(x) \triangleq \mathbb{E}_{(r,y) \sim p_\theta(\cdot|x)} \left[\mathbf{1}\left\{\text{ans}(y) = a^\star(x)\right\}\right] \in [0, 1],$$

where $a^\star(x)$ denotes the correct answer to x, and $\text{ans}(y)$ extracts the final answer from y.

2.2 Compute Law

We hypothesize that, if a reasoning model allocates its reasoning compute efficiently, the amount of compute is expected to scale proportionally with complexity in approximation, *i.e.*, $C_\theta(x) \propto \kappa(x)$:

Hypothesis 1 (Compute Law)**.** For an optimal reasoning model M_θ and a question x with complexity $\kappa(x)$, there exist $\alpha_\theta > 0$ with,

$$C_\theta(x) = \alpha_\theta \, \kappa(x) + o(\kappa(x)),$$

for some $\alpha_\theta > 0$ that depends only on M_θ and the decoding strategy. $o(\kappa(x))$ denotes a small systematic overhead that is sublinear, *i.e.*, $o(\kappa)/\kappa \to 0$ when $\kappa \to \infty$.

[2] One example of c can be "`Answer the following questions in order: Q1. {Q1}\nQ2. {Q2}`".

Specifically, the $o(\kappa(x))$ term captures the introductory and transition tokens during the reasoning process. These tokens generally constitute a very small portion of the overall reasoning and can therefore be ignored in practice. Further discussion on compute definitions is provided in Appendix F.

Two Tractable Alternative Properties as Proxies. As discussed in Definition 1, the complexity $\kappa(x)$ is difficult to measure in practice. Consequently, empirically validating the linear relationship is nontrivial, as it would require known complexity values for individual questions. To address this, we adopt two tractable properties as empirical proxies for studying the laws: *monotonicity* and *compositionality*. These properties offer two key advantages: (i) they are tractable to verify without access to the exact value of $\kappa(x)$. Monotonicity relies only on relative comparisons between questions, while compositionality tests whether compute is additive over independent question pairs; (ii) they are theoretically sufficient to imply the proposed compute law (Proposition 1).

Property 1 (Compute-Complexity Monotonicity)**.** For $x_1, x_2 \in \mathcal{X}$, the reasoning compute is *monotonically non-decreasing* with complexity:

$$\kappa(x_1) \leq \kappa(x_2) \implies C_\theta(x_1) \leq C_\theta(x_2).$$

For compositionality, when two sub-questions are related, their composition may involve shared reasoning efforts that cannot be reliably quantified. To examine compositionality in a controllable manner, we therefore require the sub-questions to be independent.

Definition 4 (Independence)**.** For $x_1, x_2 \in \mathcal{X}$, x_1 and x_2 are *independent* if the complexity of their composition is additive, *i.e.*, $\kappa(x_1 \oplus x_2) = \kappa(x_1) + \kappa(x_2)$.

Since the exact complexity is difficult to obtain, we define independence operationally. Suppose each $x \in \mathcal{X}$ is associated with a set of mathematical concepts[3] $\mathcal{S}(x) \subseteq \mathcal{S}$ relevant to solving it. We consider two questions x_1 and x_2 independent if their concept sets are disjoint, *i.e.*, $\mathcal{S}(x_1) \cap \mathcal{S}(x_2) = \varnothing$.

Property 2 (Compute-Complexity Compositionality)**.** For $x_1, x_2 \in \mathcal{X}$, if x_1 and x_2 are independent, their composite $x_1 \oplus x_2$ exhibits *additive compute*:

$$C_\theta(x_1 \oplus x_2) = C_\theta(x_1) + C_\theta(x_2) + o(\kappa(x_1) + \kappa(x_2)),$$

where the sublinear terms account for systematic overhead in the reasoning process (as assumed in Hypothesis 1). Therefore, the reasoning compute is *approximately additive*:

$$C_\theta(x_1 \oplus x_2) \approx C_\theta(x_1) + C_\theta(x_2).$$

Discussion. Intuitively, these properties are motivated by two basic principles: (i) more complex questions naturally require more reasoning; (ii) two independent sub-questions involve no overlapping reasoning, so the total compute is the sum of solving each one individually. In the next proposition, we state informally that these properties imply the compute law (Hypothesis 1); a formal proof is provided in Appendix E. These tractable properties offer a practical way to evaluate whether current LRMs follow the compute law.

Proposition 1. Under certain conditions, if a reasoning model M_θ satisfies compute-complexity monotonicity and compositionality, then its reasoning compute $C_\theta(x) \propto \kappa(x)$ for $x \in \mathcal{X}$.

2.3 Beyond Compute: Accuracy Law

Following Definition 1, suppose a question requires solving $\kappa(x)$ unit-cost primitive steps. If each step succeeds independently with a fixed probability and all steps must succeed for the final answer to be correct, then the overall accuracy is expected to decay exponentially with $\kappa(x)$. This intuition motivates the following formulation of the accuracy law:

Hypothesis 2 (Accuracy Law)**.** For an optimal reasoning model M_θ and a question x with complexity $\kappa(x)$, when $0 < A_\theta(x) \leq 1$, there exists the decay rate $\lambda_\theta \geq 0$ with,

$$A_\theta(x) \;=\; \exp\big(-\lambda_\theta\, \kappa(x)\big).$$

Equivalently, $\log A_\theta(x) \propto -\kappa(x)$.

Similar to the compute law, we assume that the reasoning accuracy for the optimal reasoning model M_θ satisfies two fundamental properties: *monotonicity* and *compositionality*.

[3]For example, concepts may come from Calculus (*e.g.*, derivatives), Algebra (*e.g.*, group theory), or Discrete mathematics (*e.g.*, logic).

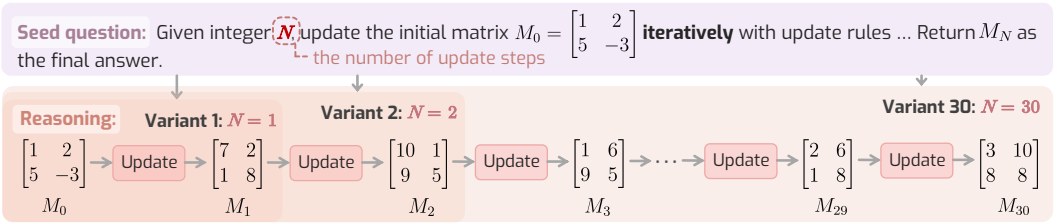


Figure 3: **Question Generation of LoRe-Mono.** For each seed question, we generate 30 variants with increasing complexity. Specifically, variant N applies the update rules N times to derive the answer, so the question complexity increases monotonically with N.

Property 3 (Accuracy-Complexity Monotonicity)**.** For $x_1, x_2 \in \mathcal{X}$, the reasoning accuracy is *monotonically non-increasing* with complexity:

$$\kappa(x_1) \leq \kappa(x_2) \implies A_\theta(x_1) \geq A_\theta(x_2).$$

Property 4 (Accuracy-Complexity Compositionality)**.** For $x_1, x_2 \in \mathcal{X}$, if x_1 and x_2 are independent, their composite $x_1 \oplus x_2$ exhibits *multiplicative accuracy*:

$$A_\theta(x_1 \oplus x_2) = A_\theta(x_1) \cdot A_\theta(x_2).$$

Discussion. These properties are motivated by two basic principles: (i) more complex questions tend to have lower accuracy; (ii) for two independent questions with accuracies p_1 and p_2 (e.g., $p_1 = 0.8$, $p_2 = 0.7$), the probability of correctly answering both should be $p_1 \cdot p_2$ (e.g., 0.56). We state below that these properties imply the accuracy law, with a formal proof provided in Appendix E.

Proposition 2. Under certain conditions, if a reasoning model M_θ satisfies accuracy-complexity monotonicity and compositionality, then its reasoning accuracy $\log A_\theta(x) \propto -\kappa(x)$ for $x \in \mathcal{X}$.

3 Do Current LRMs Follow the Laws?

In this section, we evaluate whether current LRMs follow the *reasoning laws*, since they may not necessarily satisfy these laws in practice. Hence, we propose LoRe-Bench, a two-fold benchmark that leverages two tractable properties, monotonicity and compositionality, to examine LRMs.

3.1 LoRe-Mono

Evaluating the monotonicity property in Property 1 or Property 3 requires comparing the complexity of arbitrary question pairs. However, due to its definition via minimal solution length, complexity is inherently difficult to quantify in practice. As a result, existing benchmarks are not suited for such analysis. To address this challenge, we construct LoRe-Mono, a synthetic benchmark where questions are carefully curated and validated to follow known complexity orderings. LoRe-Mono comprises 1200 evaluation instances derived from seed questions across four reasoning-intensive domains, allowing us to systematically assess the monotonicity of reasoning compute and accuracy.

(1) Seed Question Curation. We select four reasoning-intensive domains—*math*, *science*, *language*, *code*—and curate 10 diverse seed questions for each. A seed question defines a problem template shared across its variants.

(2) From Seed Questions to Variants. As shown in Fig. 3, for each seed question, we create a series of variants (30 in total) that become increasingly complex by requiring more steps to reach the final answer. For example, variant 1 requires one matrix operation, and variant 30 requires thirty, with the identical operation applied repeatedly. By design, a larger number of steps corresponds to higher complexity.

(3) Program-based Generation and Manual Verification. All variants are generated through Python scripts to ensure correctness and scalability. To prevent unintended shortcuts such as periodic patterns, we manually verify each seed question and review sampled variants. We provide detailed seed questions and variants for each domain in Appendix G.

We use the Spearman correlation coefficients $\rho \in [-1, 1]$ to measure how the variant index, which directly determines the constructed question's complexity, relates to two quantities: reasoning compute and log accuracy. A high correlation with reasoning compute indicates that compute grows

monotonically with complexity (Property 1), while a negative correlation with log accuracy indicates that accuracy tends to degrade as complexity increases (Property 3).

3.2 LoRe-Compo

In contrast, assessing compositionality is more straightforward: it only requires taking any two independent questions as sub-questions and constructing a composition. This benchmark serves as a controlled, quantitative testbed for rigorously evaluating compositional behaviors. We build LoRe-Compo from MATH500 (Lightman et al., 2023), where each question is labeled by subject (e.g., Algebra, Geometry). Specifically, we randomly sample a pair of questions (x_1, x_2) from *distinct subjects* to ensure independence, and concatenate them into a composite question x_{12}. Each original question is used only once, yielding 250 triplets, each with two sub-questions and their composition: $\mathcal{D}_{\text{LoRe-Compo}} = \{(x_1^{(i)}, x_2^{(i)}, x_{12}^{(i)})\}_{i=1}^{250}$. Recall that for a function $f_\theta(\cdot)$ (either $C_\theta(\cdot)$ or $\log A_\theta(\cdot)$), compositionality implies that $f_\theta(x_{12}) \approx f_\theta(x_1) + f_\theta(x_2)$. We therefore quantify the degree to which a model follows this property using the mean absolute deviation (MAD):

$$\text{MAD}_f = \mathbb{E}_{(x_1,x_2,x_{12})\sim\mathcal{D}_{\text{LoRe-Compo}}} \left[\left| f_\theta(x_{12}) - \big(f_\theta(x_1) + f_\theta(x_2)\big) \right| \right].$$

A smaller MAD indicates stronger adherence to the compositionality property. However, MAD is scale-dependent. To address this, we adopt the Normalized MAD (nMAD):

$$\text{nMAD}_f = \frac{\text{MAD}_f}{S_f}, \quad S_f = \mathbb{E}_{(x_1,x_2,x_{12})\sim\mathcal{D}_{\text{LoRe-Compo}}} \left[|f_\theta(x_1) + f_\theta(x_2)| \right].$$

3.3 Findings and Analysis

Evaluation Setups. We examine 11 LRMs on LoRe-Bench: 8 standard open-sourced models—DeepSeek-R1-Distill (Qwen-1.5B, Qwen-7B, Llama-8B, Qwen-14B) (Guo et al., 2025), Phi-4-mini-reasoning (Xu et al., 2025a), OpenReasoning-Nemotron-14B (Ahmad et al., 2025), Sky-T1-32B-Preview (Team, 2025a) and Qwen3-Next-80B-A3B-Thinking (Team, 2025b)—and 2 models that apply reasoning length control, Thinkless-1.5B-RL-DeepScaleR (Fang et al., 2025) and AdaptThink-7B-delta0.05 (Zhang et al., 2025a), and 1 proprietary model, Gemini 3.1 Flash-Lite. For each question, we sample 8 outputs per model with a fixed temperature (0.8 for the Phi-4 family and 0.6 for other model families from technical reports) and a maximum length of 20480 tokens. For LoRe-Mono, at each variant index we average reasoning compute[4] and log accuracy over 40 questions then compute the Spearman correlation.

Table 1: **Monotonicity Results on LoRe-Mono.** We examine whether LRMs satisfy monotonicity. Spearman correlations are reported for reasoning compute and log accuracy. *Lang.* denotes *Language*.

Size	Models	Reasoning Compute ↑					Log Accuracy ↓				
		Math	Science	Lang.	Code	All	Math	Science	Lang.	Code	All
1.5B	DeepSeek-R1-1.5B	0.861	0.910	-0.346	0.151	**0.875**	-0.795	-0.864	-0.210	-0.487	**-0.868**
	Thinkless-1.5B	0.943	0.961	0.648	0.794	**0.976**	-0.951	-0.934	-0.556	-0.539	**-0.960**
3.8B	Phi-4-mini	0.980	0.973	0.936	0.922	**0.988**	-0.965	-0.802	-0.911	-0.822	**-0.954**
7B	DeepSeek-R1-7B	0.956	0.975	0.901	0.970	**0.991**	-0.946	-0.876	-0.899	-0.818	**-0.978**
	AdaptThink-7B	0.984	0.995	0.950	0.984	**0.995**	-0.963	-0.949	-0.904	-0.888	**-0.972**
8B	DeepSeek-R1-8B	0.982	0.962	0.864	0.963	**0.988**	-0.944	-0.796	-0.924	-0.843	**-0.947**
14B	Nemontron-14B	0.964	0.976	0.917	0.970	**0.993**	-0.778	-0.751	-0.793	-0.911	**-0.913**
	DeepSeek-R1-14B	0.978	0.973	0.903	0.980	**0.990**	-0.888	-0.888	-0.803	-0.933	**-0.981**
32B	Sky-T1-32B	0.988	0.982	0.711	0.874	**0.967**	-0.939	-0.876	-0.783	-0.860	**-0.963**
80B	Qwen3-80B-Next	0.977	0.984	0.794	0.993	**0.992**	-0.403	-0.551	-0.774	-0.907	**-0.973**
—	Gemini 3.1 Flash-Lite	0.976	0.916	0.969	0.991	**0.976**	-0.922	-0.690	-0.321	-0.874	**-0.935**

Current LRMs Largely Satisfy Monotonicity. On LoRe-Mono, all LRMs exhibit a strong positive correlation between reasoning compute and the variant index, which directly reflects question

[4]We apply max-min normalization to the reasoning compute of each question to prevent any items from dominating the results.

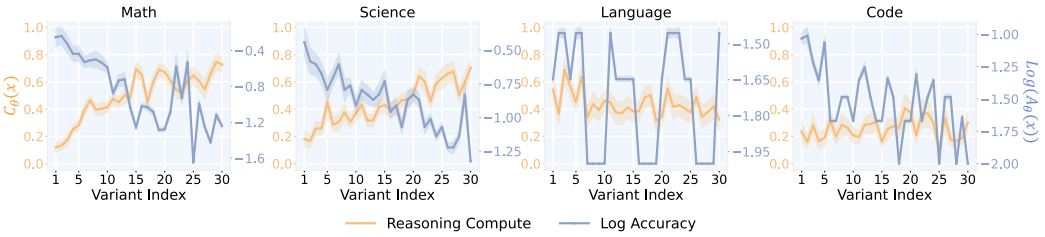


Figure 4: **Visualizations of Monotonicity Results on DeepSeek-R1-1.5B.** For each domain, we plot reasoning compute and log accuracy over variant index. The curves report the mean accuracy across 10 question series, and the shaded regions denote the standard deviation.

Table 2: **Compositionality Results on LoRE-Compo.** We calculate nMAD for reasoning compute (C_θ) and log accuracy ($\log A_\theta$).

Models	$\text{nMAD}_{C_\theta} \downarrow$	$\text{nMAD}_{\log A_\theta} \downarrow$
DeepSeek-R1-1.5B	0.528	2.368
Thinkless-1.5B	0.339	0.694
Phi-4-mini	0.322	0.732
DeepSeek-R1-7B	0.337	1.170
AdaptThink-7B	0.327	0.791
DeepSeek-R1-8B	0.423	0.818
DeepSeek-R1-14B	0.368	1.310
Nemotron-14B	0.354	1.188
Sky-T1-32B	0.392	1.900
Qwen3-80B-Next	0.411	1.487
Gemini 3.1 Flash-Lite	0.343	0.490

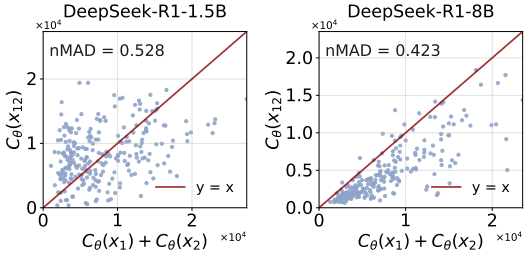


Figure 5: **Visualizations of Compositionality Results on Reasoning Compute.** We plot $C_\theta(x_1 \oplus x_2)$ against $C_\theta(x_1) + C_\theta(x_2)$. Further results are provided in Appendix G.4.

complexity, with most overall Spearman correlations close to 1, as shown in Tab. 1. The only exception is DeepSeek-R1-Distill-Qwen-1.5B, which has the weakest reasoning ability among the six models and yields a lower overall correlation (0.875). As illustrated in Fig. 4, notably, in the language domain its correlation between reasoning compute and complexity is negative (-0.346), while in the code domain, it is near zero (0.151). This indicates that *in some domains, the reasoning compute for this model does not systematically increase with complexity, and may even decrease*. We provide a case study as additional analysis along with visualization results for other models in Appendix G.2. Meanwhile, most LRMs exhibit a negative correlation between log accuracy and the variant index, as expected. For DeepSeek-R1-Distill-Qwen-1.5B, however, this trend appears noticeably weaker.

Current LRMs Fail to Exhibit Compositionality. The nMAD is large for both reasoning compute and log accuracy (Tab. 2), indicating that current LRMs do not satisfy compositionality. This issue persists across different model scales, as larger models do not necessarily better satisfy compositionality than small-scale models. Fig. 5 further plots $C_\theta(x_1 \oplus x_2)$ against $C_\theta(x_1) + C_\theta(x_2)$ for two representative LRMs. If an LRM adhered to the compositionality law, most points would align closely with the $y = x$ line. In practice, however, the majority of points deviate substantially. Notably, even models equipped with reasoning length control mechanisms (Thinkless-1.5B and AdaptThink-7B) exhibit considerable deviations, suggesting that such techniques do not inherently promote compositional behavior. We provide additional analyses of the proprietary model, prompt variants, and commonsense reasoning evaluation in Appendix H.3, H.5, and H.6.

4 Improving Reasoning via Enforcing Compositionality

In Section 3, we showed that while most LRMs generally satisfy *monotonicity*, they often fail to satisfy *compositionality*. Based on Hypothesis 1, this observation motivates a natural question: *can enforcing compositionality lead to stronger reasoning capacity?* In response, we propose a simple yet effective supervised fine-tuning (SFT) method to promote compositional behavior in LRMs. Importantly, we focus on enforcing compositionality specifically with respect to *reasoning compute*, as it provides a more direct and actionable criterion for selecting supervision examples.[5]

[5]Accuracy compositionality is hard to enforce directly, as it does not specify which reasoning path to take for supervision.

Proposed Method: SFT-Compo. Specifically, let M_θ be an LRM and $\mathcal{D}_{\text{train}}$ a training dataset. Following the construction in Section 3.2, we select question pairs $(x_1, x_2) \in \mathcal{D}_{\text{train}}$ from distinct categories and form composite questions $x_{12} = x_1 \oplus x_2$. For each triplet (x_1, x_2, x_{12}), we sample K model outputs $o = (r, y) \in \mathcal{O}$ from an LRM (either the current model M_θ or a stronger teacher model), where $r \in \mathcal{R}$ is a reasoning path and $y \in \mathcal{Y}$ is the final answer:

$$\{o_1^{(k)} = (r_1^{(k)}, y_1^{(k)})\}_{k=1}^K \text{ for } x_1, \quad \{o_2^{(k)} = (r_2^{(k)}, y_2^{(k)})\}_{k=1}^K \text{ for } x_2, \quad \{o_{12}^{(k)} = (r_{12}^{(k)}, y_{12}^{(k)})\}_{k=1}^K \text{ for } x_{12}.$$

Since compositionality is defined over reasoning paths, among the K^3 combinations (o_1, o_2, o_{12}), we consider only those where all three reasoning paths r_1, r_2, r_{12} lead to correct answers, and select the combination that best satisfies the compositionality condition:

$$(r_1^*, r_2^*, r_{12}^*) = \arg\min_{r_1, r_2, r_{12}} |\ell(r_1) + \ell(r_2) - \ell(r_{12})| \text{ s.t. } r_1, r_2, r_{12} \text{ yields a correct final answer.} \quad (1)$$

Each triplet thus yields three supervised examples: (x_1, o_1^*), (x_2, o_2^*), and (x_{12}, o_{12}^*), where $o_i^* = (r_i^*, y_i^*)$ with y_i^* the final answer paired with r_i^* in the sampled outputs. Aggregating across all triplets, we construct the compositional supervision dataset $\mathcal{D}_{\text{comp}} = \{(x_1, o_1^*), (x_2, o_2^*), (x_{12}, o_{12}^*) \mid (x_1, x_2) \in \mathcal{D}_{\text{train}}\}$. We then perform SFT on $\mathcal{D}_{\text{comp}}$ to encourage M_θ to internalize compositional reasoning behavior.

5 Experiments

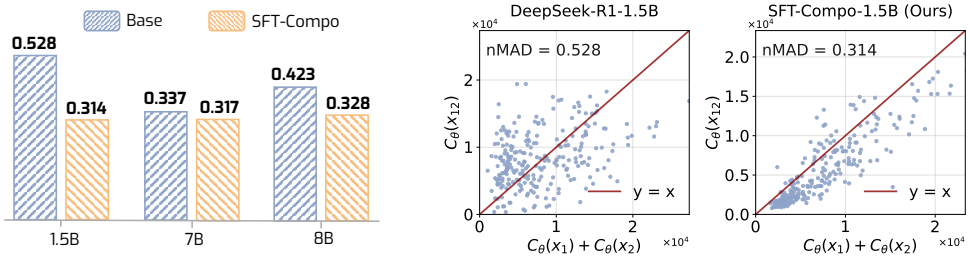
We now empirically evaluate SFT-Compo, addressing two research questions: (1) whether it effectively enforces compositionality, and (2) whether it further improves the reasoning capacity of LRMs. We also provide additional insightful findings in our analysis.

5.1 Experimental Setup

Model, Dataset and SFT Recipe. We evaluate four LRMs: DeepSeek-R1-Distill (Qwen-1.5B, Qwen-7B, Llama-8B) (Guo et al., 2025) and Phi-4-mini-reasoning (Xu et al., 2025a). We construct a dataset of sub-question and composite-question triplets using a subset of DeepScaler (Luo et al., 2025b). For each question, we use DeepSeek-R1-Distill-Qwen-14B as a stronger teacher model to sample $K = 8$ model outputs. We then construct the compositionality-enforced dataset $\mathcal{D}_{\text{comp}}$ as described in Eqn. 1, which contains 3.9K question-output pairs. We fine-tune each LRM on $\mathcal{D}_{\text{comp}}$ for 5 epochs with a batch size of 16. Details are provided in Appendix H.

Evaluation. To evaluate compositionality, we use LoRE-Compo. For general reasoning capacity, we consider six benchmarks: GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2023), AIME 2024, AIME 2025 (Mathematical Association of America, 2025), AMC 2023 (AI-MO, 2024), and OlympiadBench (He et al., 2024). We set the maximum generation length to 10240 tokens.

5.2 Main Results



(a) nMAD_{C_θ} on LoRE-Compo. (b) Visualizations of Reasoning Compute Compositionality.

Figure 6: **Comparison of Reasoning Compute Compositionality** on LoRE-Compo.

Does SFT-Compo Effectively Enforce Compositionality Compared to the Base Model? We compare LRMs before and after SFT using the nMAD of reasoning compute on LoRE-Compo. As shown in Fig. 6a, SFT-Compo consistently reduces nMAD compared to the base model. On the 1.5B model, SFT-Compo achieves a reduction from 0.528 to 0.314 (a 40.5% reduction), and on the 8B model, from 0.423 to 0.328 (a 22.5% reduction). We further visualize the results on the 1.5B

Table 3: **General Reasoning Evaluation Results**. We evaluate `Base` (pre-SFT), `SFT`, and `SFT-Compo` (Ours) on mathematical and science reasoning benchmarks. All numbers report Pass@1 accuracy (%) computed over 8 sampled outputs. $\overline{\text{Pass@1}}$ denotes the average across the six benchmarks. Numbers in **orange** indicate improvements relative to the base model.

Base Model	Method	MATH					SCIENCE	$\overline{\text{Pass@1}}$
		AIME24	AIME25	AMC23	MATH500	GSM8K	Olympiad	
DeepSeek-R1-1.5B	Base	18.8	20.4	59.7	71.6	81.2	33.8	47.6
	SFT	20.4$_{+1.6}$	21.5$_{+1.1}$	59.6$_{-0.1}$	76.4$_{+4.8}$	81.7$_{+0.5}$	36.1$_{+2.3}$	49.3$_{+1.7}$
	SFT-Compo (Ours)	**26.2**$_{+7.4}$	**21.7**$_{+1.3}$	**65.0**$_{+5.3}$	**77.6**$_{+6.0}$	**85.1**$_{+3.9}$	**38.7**$_{+4.9}$	**52.4**$_{+4.8}$
DeepSeek-R1-7B	Base	36.3	27.5	79.0	86.8	91.0	48.1	61.5
	SFT	40.0$_{+3.7}$	32.5$_{+5.0}$	80.4$_{+1.4}$	88.0$_{+1.2}$	**91.6**$_{+0.6}$	48.4$_{+0.3}$	63.5$_{+2.0}$
	SFT-Compo (Ours)	**43.3**$_{+7.0}$	**33.2**$_{+5.7}$	**80.6**$_{+1.6}$	**88.8**$_{+2.0}$	**91.6**$_{+0.6}$	**50.5**$_{+2.4}$	**64.7**$_{+3.2}$
DeepSeek-R1-8B	Base	28.3	22.9	71.9	76.4	86.5	40.9	54.5
	SFT	30.4$_{+2.1}$	24.2$_{+1.3}$	75.2$_{+3.3}$	82.6$_{+6.2}$	88.0$_{+1.5}$	44.7$_{+3.8}$	57.5$_{+3.0}$
	SFT-Compo (Ours)	**31.3**$_{+3.0}$	**29.2**$_{+6.3}$	**76.9**$_{+5.0}$	**83.0**$_{+6.6}$	**89.5**$_{+3.0}$	**46.8**$_{+5.9}$	**59.5**$_{+5.0}$
Phi-4-mini	Base	32.5	24.6	71.9	86.4	91.4	50.0	59.5
	SFT	40.0$_{+7.5}$	26.2$_{+1.6}$	73.4$_{+1.5}$	**87.0**$_{+0.6}$	91.6$_{+0.2}$	49.5$_{-0.5}$	61.3$_{+1.8}$
	SFT-Compo (Ours)	**43.7**$_{+11.2}$	**31.7**$_{+7.1}$	**76.9**$_{+5.0}$	**87.0**$_{+0.6}$	**92.8**$_{+1.4}$	**51.2**$_{+1.2}$	**63.9**$_{+4.4}$

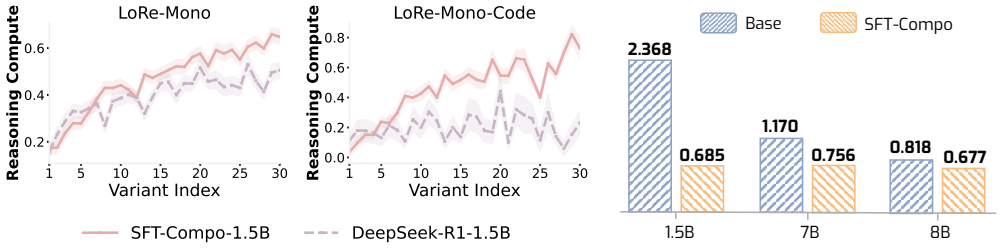
model in Fig. 6b, where `SFT-Compo` aligns much more closely with the $y{=}x$ line. Therefore, *the compositionality of reasoning compute can be effectively enforced in a simple manner via* `SFT-Compo`.

Does Enforcing Compositionality Lead to Stronger Reasoning Capabilities? As shown in Tab. 3, `SFT-Compo` consistently improves performance across all six benchmarks and all four model sizes. For instance, on the 8B model, it yields a notable gain of $+5.0$ in average Pass@1. To rule out the possibility that performance gains stem solely from leveraging outputs generated by a stronger teacher model, we introduce a control baseline, `SFT`, which constructs the training dataset by uniformly sampling one correct reasoning path for each question in the triplet:

$$(r_1^*, r_2^*, r_{12}^*) \sim \text{Unif}\left(\{(r_1, r_2, r_{12}) \mid r_1, r_2, r_{12} \text{ each yield a correct final answer}\}\right).$$

Notably, `SFT-Compo` outperforms `SFT` in all cases, showing that the gains are not just from distilling a stronger model but from better compliance with compositionality. This supports our Hypothesis 1—that *stronger models better follow reasoning laws*—and demonstrate that *encouraging compositionality further enhances the reasoning capabilities of LRMs.*

5.3 Synergistic Effect Analysis



(a) Visualizations of monoticity results on LoRe-Mono. (b) $\text{nMAD}_{\log A_\theta}$ on LoRe-Compo.

Figure 7: **Synergistic Effects** Among Different Reasoning Properties and Laws.

Enforcing Compositionality in Reasoning Compute Improves Its Monotonicity. Recall from Section 3.3 that DeepSeek-R1-1.5B exhibits weak monotonicity in reasoning compute. Fig. 7a shows `SFT-Compo` significantly improves this property, increasing the overall Spearman correlation from 0.875 to 0.977. Specifically, in the code domain, it rises from 0.151 to 0.914. This indicates that enforcing compositionality can implicitly enhance monotonicity.

Enforcing Compositionality in Reasoning Compute Improves Compositionality in Accuracy. Interestingly, though `SFT-Compo` is designed to enhance compositionality in reasoning compute, it improves the compositionality of log accuracy. Fig. 7b shows that the nMAD of log accuracy drops from 2.368 to 0.685 on the 1.5B model (a 71.1% reduction), and from 1.170 to 0.756 on the 7B model (a 35.4% reduction). This suggests a possible interplay among different reasoning laws.

6 Related Work

LRMs have emerged as a family of foundation models (Wiggins & Tejani, 2022). Since the advent of OpenAI o1 (Jaech et al., 2024), the "thinking-then-answering" paradigm has been widely adopted, with notable follow-ups such as DeepSeek-R1 and Phi-4-Reasoning (Abdin et al., 2024; Guo et al., 2025; Team, 2024). Our framework builds upon the contemporary paradigm of adaptive reasoning, wherein the model's reasoning budget is dynamically controlled either through post-training interventions (Luo et al., 2025a; Zhou et al., 2025b) or at test time (Muennighoff et al., 2025; Xu et al., 2025b; Zhang et al., 2025b). Specifically, one line of work explores post-training techniques that modulate when and how long a model should reason (Chen et al., 2024a; Yong et al., 2025), while another frontier focuses on dynamically adjusting reasoning behavior during inference (Qiao et al., 2025; Liu & Wang, 2025). Refer to Appendix C for additional related work.

7 Conclusions

As a comprehensive study from theoretical hypotheses to empirical validation, we advance a theoretical perspective grounded in human reasoning for improving reasoning in LRMs. We hope LoRE can inspire potential strategies that guide models toward their optimal paradigms of thinking.

Acknowledgements and Disclosures

The authors thank Heng Dong, Runpei Dong and Chanakya Ekbote for their insightful feedback and valuable suggestions on this project. Huan Zhang and Junyu Zhang are supported in part by a Schmidt Sciences Trustworthy AI award. The authors declare no competing interests.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning, 2025. *URL https://arxiv. org/abs/2503.04697*, 2025.

Wasi Uddin Ahmad, Somshubra Majumdar, Aleksander Ficek, Sean Narenthiran, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Vahid Noroozi, and Boris Ginsburg. OpenCodeReasoning-II: A Simple Test Time Scaling Approach via Self-Critique, 2025. URL https://arxiv.org/abs/2507.09075.

AI-MO. AIMO Validation Dataset - AMC. https://huggingface.co/datasets/AI-MO/aimo-validation-amc, 2024. URL https://huggingface.co/datasets/AI-MO/aimo-validation-amc. Accessed: 2025-05-19.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

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. *Advances in Neural Information Processing Systems*, 37:54872–54904, 2024a.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024b.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Chongyu Fan, Yihua Zhang, Jinghan Jia, Alfred Hero, and Sijia Liu. Cyclicreflex: Improving large reasoning models via cyclical reflection token scheduling. *arXiv preprint arXiv:2506.11077*, 2025.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *arXiv preprint arXiv:2505.13379*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Junyan Li, Wenshuo Zhao, Yang Zhang, and Chuang Gan. Steering llm thinking with budget guidance. *arXiv preprint arXiv:2506.13752*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*, 2025.

Xin Liu and Lu Wang. Answer convergence as a signal for early stopping in reasoning. *arXiv preprint arXiv:2506.02536*, 2025.

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*, 2025a.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025b. Notion Blog.

Mathematical Association of America. American Invitational Mathematics Examination – AIME. *American Invitational Mathematics Examination – AIME 2025*, February 2025. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime. Accessed: 2025-09-21.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

Allen Newell, Herbert Alexander Simon, et al. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ, 1972.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. Concise: Confidence-guided compression in step-by-step efficient reasoning. *arXiv preprint arXiv:2505.04881*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*, 2025.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. Chain of thoughtlessness? an analysis of cot in planning, 2024. *URL https://arxiv. org/abs/2405.04776*, 2024.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. *arXiv preprint arXiv:2505.00127*, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *Advances in Neural Information Processing Systems*, 38: 173449–173476, 2026a.

Yifan Sun, Naicheng Yu, Jiawen Gong, Jingyan Shen, and Huan Zhang. Your benchmark is an empirical measure over difficulty. In *Advances in Neural Information Processing Systems*, 2026b. URL https://openreview.net/forum?id=fCQ7KtG95S.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, C Chen, C Li, C Xiao, C Du, C Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms, 2025. *URL https://arxiv. org/abs/2501.12599*, 2025.

NovaSky Team. Sky-t1: Fully open-source reasoning model with o1-preview performance in 450 budget, 2025a.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Qwen Team. Qwen3 technical report, 2025b. URL https://arxiv.org/abs/2505.09388.

Sang Truong, Yuheng Tu, Percy Liang, Bo Li, and Sanmi Koyejo. Reliable and efficient amortized model-based evaluation. *arXiv preprint arXiv:2503.13335*, 2025.

Alan M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proc. London Math. Soc.*, s2-42(1):230–265, 1937. doi: 10.1112/PLMS/S2-42.1.230. URL https://doi.org/10.1112/plms/s2-42.1.230.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 318–327, 2025.

Walter F Wiggins and Ali S Tejani. On the opportunities and risks of foundation models for natural language processing in radiology. *Radiology: Artificial Intelligence*, 4(4):e220119, 2022.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.

Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, et al. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math. *arXiv preprint arXiv:2504.21233*, 2025a.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Xixian Yong, Xiao Zhou, Yingying Zhang, Jinlin Li, Yefeng Zheng, and Xian Wu. Think or not? exploring thinking efficiency in large reasoning models via an information-theoretic lens. *arXiv preprint arXiv:2505.18237*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*, 2025a.

Junyu Zhang, Runpei Dong, Han Wang, Xuying Ning, Haoran Geng, Peihao Li, Xialin He, Yutong Bai, Jitendra Malik, Saurabh Gupta, and Huan Zhang. Alphaone: Reasoning models thinking slow and fast at test time. *arXiv preprint arXiv:2505.24863*, 2025b.

Hongli Zhou, Hui Huang, Ziqing Zhao, Lvyuan Han, Huicheng Wang, Kehai Chen, Muyun Yang, Wei Bao, Jian Dong, Bing Xu, et al. Lost in benchmarks? rethinking large language model benchmarking with item response theory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35085–35093, 2026.

Kaiwen Zhou, Chengzhi Liu, Xuandong Zhao, Shreedhar Jangam, Jayanth Srinivasa, Gaowen Liu, Dawn Song, and Xin Eric Wang. The hidden risks of large reasoning models: A safety assessment of r1. *arXiv preprint arXiv:2502.12659*, 2025a.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*, 2025b.

A Reproducibility Statement

Our code repository is available at https://github.com/ASTRAL-Group/LoRe. In Section 3.3 and Section 5.1, we provide a detailed description of the experimental setup, including dataset, models, training and evaluation procedures. Additional implementation details can be found in Appendix H.2.

B Broader Impact

This work targets complex reasoning problems with LRMs, which we believe will lead to no ethical concerns. However, since LRMs are modern variants of LLMs, any ethical concerns raised by LLMs can potentially exist.

C Additional Related Work

Large Reasoning Models. Large Reasoning Models (LRMs) have emerged as a family of foundation models (Wiggins & Tejani, 2022). Since the advent of OpenAI o1 (Jaech et al., 2024), this "thinking-then-answering" paradigm has been widely adopted. Notably, o1-like Reasoning Models can solve increasingly complex reasoning problems through elaborate reasoning chains (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024). Numerous efforts replicating o1's success include DeepSeek-R1 and Phi-4-Reasoning (Abdin et al., 2024; Guo et al., 2025; Team, 2024). Despite impressive progress, the internal mechanisms and behavioral patterns of reasoning in LRMs remain underexplored. Shojaee et al. (2025) take a step in this direction by examining reasoning through the lens of problem complexity, though their analysis is limited to a constrained puzzle-solving setting.

Reasoning Length Control. Our framework builds upon the contemporary paradigm of adaptive reasoning, in which the reasoning budget of the model is controlled either during post-training (Luo et al., 2025a; Zhou et al., 2025b) or at test time (Muennighoff et al., 2025; Xu et al., 2025b; Zhang et al., 2025b). One line of work develops post-training techniques that modulate when and how long a model should reason (Chen et al., 2024a; Yong et al., 2025). This is achieved through two primary strategies: one involves supervised fine-tuning on variable-length CoT with concise yet sufficient reasoning (Aggarwal & Welleck, 2025; Team et al., 2025); the other utilizes RL through length penalty (Zhang et al., 2025a; Fang et al., 2025; Liu et al., 2025). Beyond these, another frontier involves implementing dynamic control over reasoning during inference. For example, some approaches allocate inference budget via confidence (Qiao et al., 2025; Liu & Wang, 2025), while others employ a secondary controller to modulate (Li et al., 2025).

D Limitations

LoRe primarily focuses on two fundamental aspects of reasoning, reasoning compute and accuracy. While compute law and accuracy law capture core reasoning dynamics, there may exist other intrinsic laws. For instance, laws related to reasoning diversity, robustness, and coherence may offer complementary perspectives and could naturally extend the LoRe framework. Although we provide experiments on proprietary models, evaluating our reasoning laws ideally requires access to the full reasoning process, which is typically not available through current commercial APIs. As a result, our evaluation on proprietary models may be less reliable, and stronger conclusions would require internal measurements from service providers. It is technically plausible for the commercial service providers to evaluate their own models based on our framework, and the insights from our paper can be utilized to build better proprietary models.

E Proofs and Corollaries

We first restate Proposition 1 and Proposition 2 formally and provide a complete proof, along with corresponding corollaries.

Proposition 1 (Formal Version)**.** Fix a question space $\mathcal{X}$, a *complexity* map $\kappa : \mathcal{X} \to \mathbb{N} \cup \{\infty\}$, and a *reasoning* compute map $C_\theta : \mathcal{X} \to \mathbb{R}_{\geq 0}$. Let $\oplus$ be a binary composition operator. For $m \geq 3$ and

jointly independent $x_1, \ldots, x_m$, define $x_1 \oplus \cdots \oplus x_m$ by a fixed bracketing (e.g. right-associated). Define

$$\mathcal{X}_{\text{fin}} := \{x \in \mathcal{X} : \kappa(x) < \infty\}, \qquad K := \kappa(\mathcal{X}_{\text{fin}}) \subseteq \mathbb{N}.$$

All assumptions below are imposed on $\mathcal{X}_{\text{fin}}$.

(A1) Monotonicity. If $\kappa(x) \le \kappa(y)$ then $C_\theta(x) \le C_\theta(y)$.

(A2) Additivity under composition of independent questions. If x, y are independent, then

$$\kappa(x \oplus y) = \kappa(x) + \kappa(y), \qquad C_\theta(x \oplus y) = C_\theta(x) + C_\theta(y).$$

(A3) For every $u \in K$ and every $m \in \mathbb{N}$, there exist $x_1, \ldots, x_m \in \mathcal{X}_{\text{fin}}$ such that

$$\kappa(x_i) = u \ \text{ for all } i, \qquad \{x_1, \ldots, x_m\} \text{ is jointly independent.}$$

Consequently, $x_1 \oplus \cdots \oplus x_m$ is valid and

$$\kappa(x_1 \oplus \cdots \oplus x_m) = mu \in K, \qquad C_\theta(x_1 \oplus \cdots \oplus x_m) = \sum_{i=1}^{m} C_\theta(x_i).$$

Then there exists a constant $\alpha_\theta \ge 0$ such that

$$C_\theta(x) = \alpha_\theta \, \kappa(x) \qquad \text{for all } x \in \mathcal{X}_{\text{fin}}.$$

Proof. Define an equivalence relation $x \sim y \iff \kappa(x) = \kappa(y)$. By (A1), $\kappa(x) \le \kappa(y)$ and $\kappa(y) \le \kappa(x)$ imply $C_\theta(x) \le C_\theta(y)$ and $C_\theta(y) \le C_\theta(x)$, hence $C_\theta(x) = C_\theta(y)$ whenever $x \sim y$. Thus there is a well-defined $f : K \to \mathbb{R}_{\ge 0}$ with $f(n) = C_\theta(x)$ for any x such that $\kappa(x) = n$.

Fix $u \in K$ and $m \in \mathbb{N}$. By (A3) choose jointly independent $x_1, \ldots, x_m$ with $\kappa(x_i) = u$. By (A2) and the fixed bracketing,

$$\kappa(x_1 \oplus \cdots \oplus x_m) = mu, \qquad C_\theta(x_1 \oplus \cdots \oplus x_m) = \sum_{i=1}^{m} C_\theta(x_i) = m\, f(u),$$

so

$$f(mu) = m\, f(u) \qquad (\forall\, u \in K,\ \forall\, m \in \mathbb{N}).$$

If $K = \{0\}$ then $C_\theta \equiv 0$ and the claim holds with $\alpha_\theta = 0$. Otherwise take $u, v \in K$ with $u, v > 0$ and let $\ell = \operatorname{lcm}(u, v)$. Then

$$f(\ell) = f\Big(u \cdot \frac{\ell}{u}\Big) = \frac{\ell}{u}\, f(u) \quad \text{and} \quad f(\ell) = f\Big(v \cdot \frac{\ell}{v}\Big) = \frac{\ell}{v}\, f(v),$$

hence $f(u)/u = f(v)/v$, independent of u, v. Write this common ratio as $\alpha_\theta \ge 0$. Therefore $f(n) = \alpha_\theta n$ for all $n \in K$, and $C_\theta(x) = f(\kappa(x)) = \alpha_\theta \, \kappa(x)$ for all $x \in \mathcal{X}_{\text{fin}}$. □

Corollary E.1 (Asymptotic version with sublinear overhead)**.** *If the compositional compute holds up to a sublinear overhead, i.e., for independent x, y,*

$$C_\theta(x \oplus y) = C_\theta(x) + C_\theta(y) + o(\kappa(x) + \kappa(y))\,,$$

and the same (A3) assumption holds, then the above proof yields

$$C_\theta(x) = \alpha_\theta \, \kappa(x) + o\big(\kappa(x)\big) \qquad (\kappa(x) \to \infty).$$

Proposition 2 (Formal Version)**.** Let $\mathcal{X}_{\text{fin}} = \{x \in \mathcal{X} : \kappa(x) < \infty\}$. Assume the setting and independence notion of Property 3, Property 4, and Assumption (A3). Then there exists $\lambda_\theta \ge 0$ such that for all $x \in \mathcal{X}_{\text{fin}}$ with $0 < A_\theta(x) \le 1$,

$$A_\theta(x) = \exp\big(-\lambda_\theta \, \kappa(x)\big).$$

Proof. Define an equivalence relation $x \sim y$ iff $\kappa(x) = \kappa(y)$. By (A1), if $x \sim y$ then both $\kappa(x) \le \kappa(y)$ and $\kappa(y) \le \kappa(x)$ hold, hence $A_\theta(x) \ge A_\theta(y)$ and $A_\theta(y) \ge A_\theta(x)$, so $A_\theta(x) = A_\theta(y)$. Therefore there exists a well-defined map

$$f : K \to (0, 1], \qquad f(n) := A_\theta(x) \text{ for any } x \in \mathcal{X}_{\text{fin}} \text{ with } \kappa(x) = n.$$

Let $g : K \to \mathbb{R}_{\ge 0}$ be $g(n) := -\log f(n)$.

Fix $u \in K$ and $m \in \mathbb{N}$. By (A3), choose jointly independent $x_1, \dots, x_m$ with $\kappa(x_i) = u$. By (A2) and the fixed bracketing,

$$\kappa(x_1 \oplus \cdots \oplus x_m) = mu, \qquad A_\theta(x_1 \oplus \cdots \oplus x_m) = \prod_{i=1}^{m} A_\theta(x_i) = \big(f(u)\big)^m.$$

Hence

$$g(mu) = -\log A_\theta(x_1 \oplus \cdots \oplus x_m) = m\, g(u) \qquad (\forall u \in K,\ \forall m \in \mathbb{N}). \tag{2}$$

If $K = \{0\}$ then $A_\theta \equiv 1$ and the claim holds with $\lambda_\theta = 0$. Otherwise, let $u, v \in K$ with $u, v > 0$ and set $\ell = \operatorname{lcm}(u, v)$. Applying Eqn. 2 twice gives

$$g(\ell) = g\Big(u \cdot \frac{\ell}{u}\Big) = \frac{\ell}{u}\, g(u) \quad \text{and} \quad g(\ell) = g\Big(v \cdot \frac{\ell}{v}\Big) = \frac{\ell}{v}\, g(v),$$

so $g(u)/u = g(v)/v$. This ratio is independent of $u, v > 0$ in K; denote it by $\lambda_\theta \ge 0$.

For any $n \in K$, if $n = 0$ then $g(n) = 0 = \lambda_\theta n$; if $n > 0$ pick any $u \in K \setminus \{0\}$ and write $n = \frac{n}{u}\, u$ to get from Eqn. 2 that $g(n) = \frac{n}{u} g(u) = \lambda_\theta n$. Therefore $g(n) = \lambda_\theta n$ for all $n \in K$, i.e. $f(n) = \exp(-\lambda_\theta n)$, and for any $x \in \mathcal{X}_{\text{fin}}$,

$$A_\theta(x) = f\big(\kappa(x)\big) = \exp\big(-\lambda_\theta\, \kappa(x)\big).$$

□

Corollary E.2 (Asymptotic version with sublinear coupling)**.** *If for independent x, y the multiplicativity holds up to a sublinear deviation in the exponent,*

$$\log A_\theta(x \oplus y) = \log A_\theta(x) + \log A_\theta(y) + o\big(\kappa(x) + \kappa(y)\big),$$

and (A3) *holds, then*

$$\log A_\theta(x) = -\lambda_\theta\, \kappa(x) + o\big(\kappa(x)\big) \quad (\kappa(x) \to \infty),$$

equivalently $A_\theta(x) = \exp\big(-\lambda_\theta \kappa(x) + o(\kappa(x))\big)$.

F Additional Discussions on Compute Definitions

In Definition 2, we adopt the expected number of reasoning tokens as a practical and effective proxy for measuring reasoning compute. Importantly, our conclusions will remain consistent under multiple reasonable alternatives, including model FLOPs and effective compute components that remove verbosity, repetition, and stylistic artifacts from the reasoning traces.

F.1 Model FLOPs

Given a fixed model, the *model FLOPs* incurred during the thinking phase are directly proportional to the number of generated reasoning tokens. Concretely, the thinking-phase FLOPs can be written as a model-dependent scaling factor multiplied by the reasoning token length, where the scaling factor is commonly approximated as $2N$, with N denoting the number of model parameters (Kaplan et al., 2020; Hoffmann et al., 2022; Wu et al., 2024).

In our LoRE-Bench evaluations, replacing reasoning token length with model FLOPs is therefore equivalent to *uniformly rescaling the y-axis in the plots by a constant factor*. This transformation would not change the relative distribution of data points and does not affect any of our experimental conclusions.

F.2 Effective Compute

Beyond total reasoning tokens, we also consider *effective compute*, which aims to capture the meaningful reasoning effort by removing systematic overhead from the reasoning process.

Intuitively, verbosity tends to scale with complexity, as more complex problems naturally require longer chains of reasoning and associated artifacts. To examine the relationship between effective compute and total compute, we conduct an additional experiment on LoRe-Compo, where effective compute is derived by applying heuristic string-matching rules to filter out verbosity artifacts. Specifically, we define a set of rules to identify isolated hedging phrases (*e.g.*, "`Wait, hold on`", "`Hmm`"), contentless fillers, and duplicate sentences, and manually verify the results on a subset of samples.

Our analysis reveals *a strong linear relationship* between effective compute and total compute. Notably, Pearson correlation r exceeds 0.8 across all settings. Here we define the effective-to-total compute ratio as Eff. Ratio $= \frac{\mathbb{E}[C_\theta^{\text{eff}}]}{\mathbb{E}[C_\theta]}$, where C_θ^{eff} and C_θ denote effective compute and total compute.

This indicates that, for a fixed model, replacing total compute with effective compute is also equivalent to rescaling the y-axis in the plots by an constant factor. Empirically, the effective-to-total compute ratio remains stable across both pre- and post-SFT settings. As a result, substituting effective compute for total compute preserves the relative distribution of data points and also does not alter our conclusions.

Table 4: **Analysis of Effective and Total Compute on LoRe-Compo.** We report the Pearson correlation r between effective compute and total compute, and the effective-to-total compute ratio.

Base Model	Method	Pearson r	Effective Ratio
DeepSeek-R1-1.5B	`Base`	0.869	85.9%
	`SFT-Compo`	0.880	86.5%
DeepSeek-R1-7B	`Base`	0.920	93.0%
	`SFT-Compo`	0.934	90.8%

G Additional Details and Results of LoRe-Bench

G.1 Additional Details of LoRe-Mono

G.1.1 Example seed questions of LoRe-Mono

Here we provide one representative seed question example for each domain.

Code - Example seed question

You are given runnable Python 3.10 code. Execute it exactly as-is in a clean environment (no extra imports). This is a Code Execution task: run the program, do not rewrite it. The loop counter i is 0-based. Return only the value of ANSWER (no other text, no formatting).
Code:

```
N = {N}; s = {init_state!r}

def f(s, i):
    if len(s) == 0:
        return s
    L = len(s); r = (i % L) + 1; s1 = s[r:] + s[:r]
    trans = str.maketrans({'a':'e','e':'i','i':'o','o':'u','u':'a'})
    return s1.translate(trans)

for i in range(N):
    s = f(s, i)

ANSWER = s
```

Math - Example seed question

Given an integer $n = \{N\}$, consider the order-2 recurrence over integers modulo M with an alternating update rule and a mild nonlinear term. You are given the initial values

$$x_0 = x0, \quad x_1 = x1.$$

We update the sequence one step at a time. Let $t = 1, 2, 3, \ldots, n$ denote the update index, where $t = 1$ is the update that produces x_2 from (x_1, x_0). At each update t, compute x_{k+1} from (x_k, x_{k-1}) using the parity of t:
- Define the nonlinear map $\varphi(z) = (z + 1)^2$. (You may reduce intermediate values modulo M at any time.)
- **Odd step (t odd):**

$$x_{k+1} \equiv Ax_k + Bx_{k-1} + C\,\varphi(x_k) \pmod{M}.$$

- **Even step (t even):**

$$x_{k+1} \equiv Ax_k - Bx_{k-1} + C\,\varphi(x_{k-1}) \pmod{M}.$$

For clarity, the first two updates are:

$$t = 1 : x_2 \equiv Ax_1 + Bx_0 + C\,\varphi(x_1) \pmod{M},$$
$$t = 2 : x_3 \equiv Ax_2 - Bx_1 + C\,\varphi(x_1) \pmod{M}.$$

Apply exactly $n - 1$ updates starting from x_0, x_1 to reach $x_n (n = n)$, and **return x_n** as a single non-negative integer in $[0, M - 1]$.
Conventions: - All modular reductions are taken modulo M and return a non-negative remainder. - The alternating rule depends on the **update index** t. - Output only the integer value of x_n (no extra text).

Science - Example seed question

You are modeling a **batch bioreactor** where an enzyme E converts substrate A to product B, but each catalytic event requires a recyclable **cofactor token** C (e.g., NAD*/NADH). Let A_t, B_t, C_t be the nonnegative integer counts of A, B, and C **after** completing tick t. You are given fixed initial counts and a regeneration period: - $A_0 = args.A0$, $B_0 = args.B0$, $C_0 = args.C0$ - Regeneration period $k = k$

For **each discrete tick** $t = 1, 2, \ldots, n$ (with $n = \{N\}$), apply the following **biochemical rule order**:
1) **Reaction (consumes cofactor)** — if both substrate and cofactor are available: - If $A_{t-1} > 0$ **and** $C_{t-1} > 0$, then one catalytic turnover occurs:

$$A_t = A_{t-1} - 1, \quad B_t = B_{t-1} + 1, \quad C_t = C_{t-1} - 1.$$

- Otherwise, no reaction this tick:

$$A_t = A_{t-1},\ B_t = B_{t-1},\ C_t = C_{t-1}.$$

2) **Cofactor regeneration (post-reaction)** — models an external respiratory/oxidative cycle returning the cofactor to its usable form at fixed intervals: - If $t \bmod k = 0$, then **after** the reaction stage:

$$C_t \leftarrow C_t + 1.$$

All updates are integer and at most ± 1 per tick ("min/+=1" granularity). **Output** the product count B_n after completing exactly $n = \{N\}$ ticks (i.e., after applying the regeneration rule at tick n).

Language - Example seed question

You are given a letter maze and a number of moves n={N}. The maze is a rectangular grid of letters G with h={H} rows and w={W} columns:

{grid_block}

Start at the cell (r0, c0) = ({R0}, {C0}). Build a string S as you move:
1) First, write down the starting letter G[r0][c0] into S. (This is done before any moves.)
2) Then repeat the following exactly n={step} times (t = 1..n):
• Let (r, c) be your current cell BEFORE moving, and let ch = G[r][c].
• Move one step based on ch (case-insensitive):
– If ch ∈ {a, e, i, o, u} (a vowel): move RIGHT → c ← (c + 1) mod {W}
– Otherwise (a consonant): move DOWN → r ← (r + 1) mod {H}
• After moving to the destination cell (r, c), append its letter G[r][c] to S.
• Now mutate the grid based on ch (the letter you moved FROM):
– If ch is a vowel: cyclically rotate COLUMN c upward by 1.
(Formally, for all i: G[i][c] ← old G[(i + 1) mod H][c].)
– Otherwise (ch is a consonant): cyclically rotate ROW r left by 1.
(Formally, for all j: G[r][j] ← old G[r][(j + 1) mod W].)

Important: The mutation happens AFTER appending G[r][c] to S, and it affects the grid used for the NEXT iteration. Indices are 0-based and the maze wraps around like a torus.
Thus, after n moves, S has length n + 1 (because the starting letter was included).

Let k = {K}. Your task is to return the word W made by the LAST k letters of S (in order).
Output W as a plain string.

G.1.2 Potential Shortcut Failures

A basic requirement of LoRe-Mono is that the complexity of question variants increases monotonically with the variant index. However, certain seed questions may violate this requirement. For instance, if answers follow a periodic pattern (*e.g.*, when all even-indexed variants have the answer 1 and all odd-indexed variants have the answer 0), a model could exploit prior patterns to guess the correct answer without performing the intended computation. To ensure benchmark reliability, we manually reviewed all variants and excluded those exhibiting periodic answer patterns.

G.2 Additional Results of LoRe-Mono

G.2.1 Additional Visualization Results

Visualizations of monotonicity results for the remaining 9 LRMs are provided in Fig. 8.

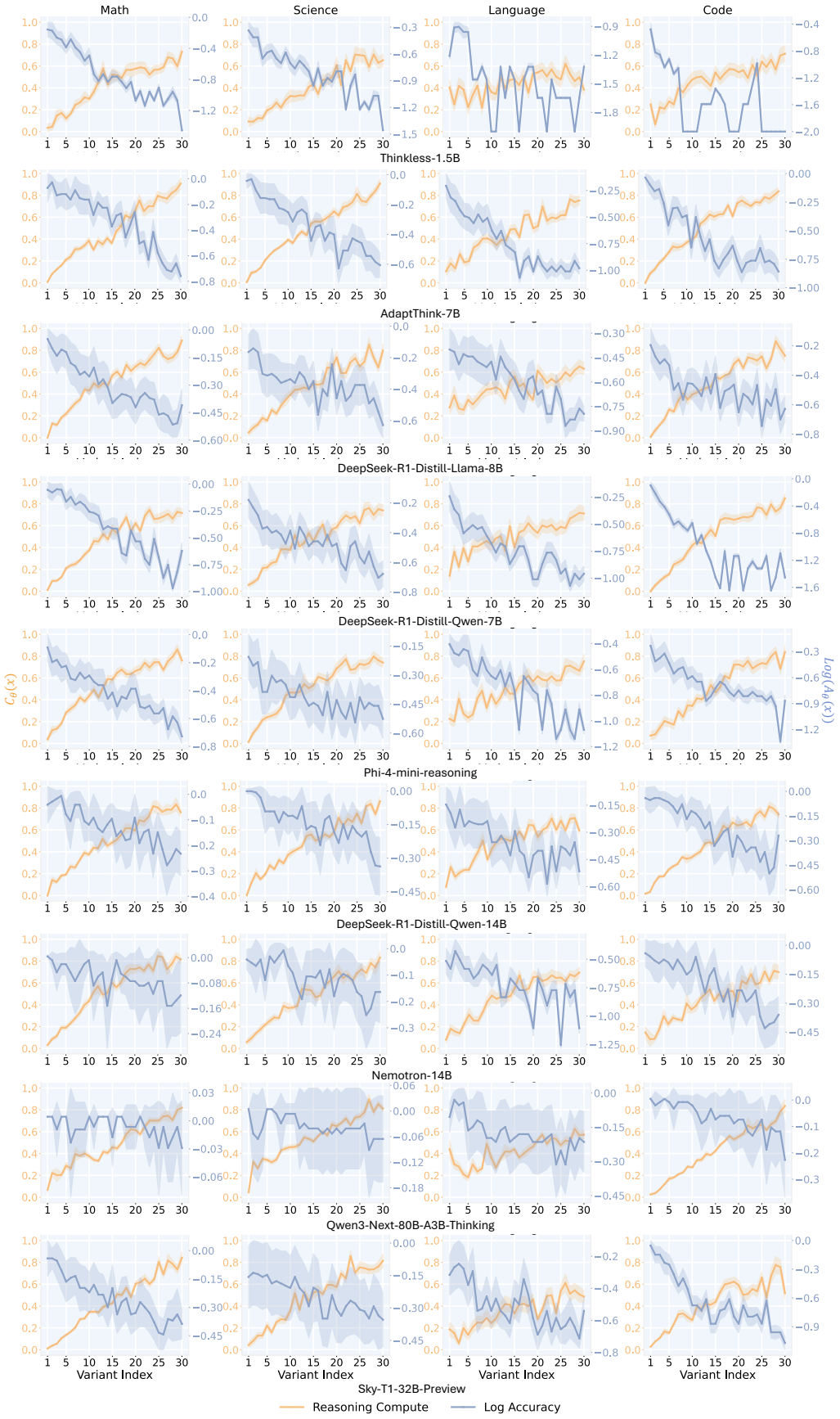


Figure 8: **Visualizations of Monotonicity Results on Remaining LRMs.**

G.2.2 "Fail to Think" Case Study

We observe that the weakest model, DeepSeek-R1-Distill-Qwen-1.5B, occasionally produces no reasoning tokens on certain code questions. For example, on variant 1 of a seed code question, the generated output is simply `<think>\n</think>`, corresponding to a reasoning length of 0.

"Fail to think" - Example model output

<think>\n</think>\n\n To solve this problem, we need to simulate the execution of a given Python code and determine the final value assigned to the variable 'ANSWER'. The code involves string manipulation and state transitions, which we will simulate step by step.\n\n### Approach\n1. **Initialization**:\n - We start with 'N = 1' iteration, 's = \"01938\"', and a function 'f' that processes the string 's'.\n\n2. **Function Definition**:\n - The function 'f' takes a string 's' and processes each character based on its position in the string. The position determines whether the count of '0', '1', '2', or '3' is incremented modulo 5.\n\n3. **Processing Each Character**:\n - For each character in 's', we count the occurrences of '0', '1', '2', and '3'..... (*omitted ...*)

G.3 Additional Details of LoRe-Compo

For all composite questions, a fixed connector links the sub-questions, ensuring they are answered in sequential order. An illustrative example is provided below.

Example composite question

Answer the following questions in order:
Q1. Jim and Martha are standing together at the corner of a rectangular field. Jim walks diagonally across the field. Martha gets to the same location by walking along its length and width. The field is 300 feet wide and 400 feet long. How many feet less than Martha does Jim walk?
Q2. Find all values of x that satisfy the equation $x = \sqrt{11 - 2x} + 4$.

G.4 Additional Results of LoRe-Compo

Visualizations of compositionality results for the remaining LRMs are provided in Fig. 9.

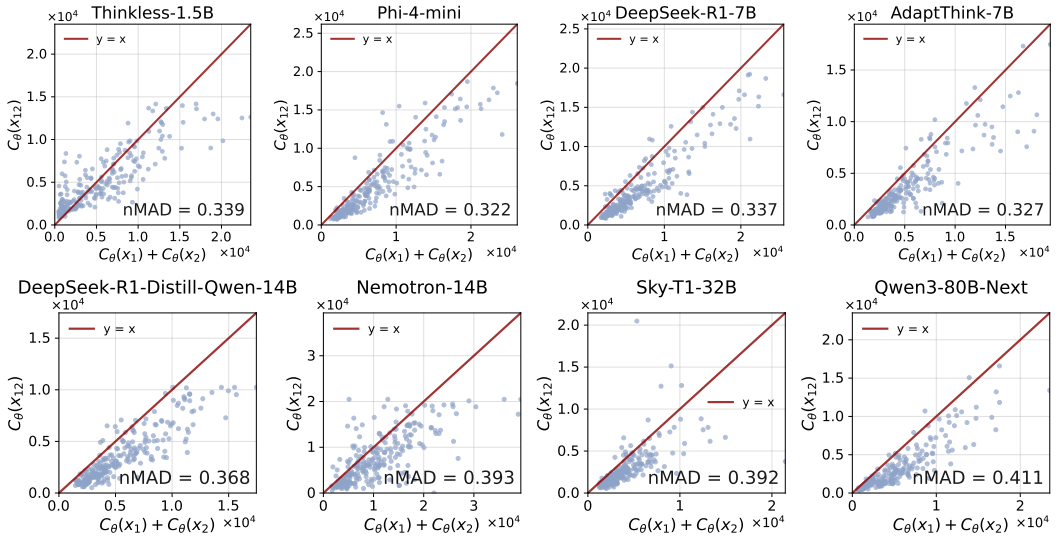


Figure 9: **Visualizations of Reasoning Compute Compositionality on More LRMs.**

H Experimental Details and Additional Results

H.1 Computational Budget

We used 8 NVIDIA L40S 48G GPUs and 8 NVIDIA RTX PRO 6000 Blackwell 96G GPUs for the experiments. For SFT-Compo training, the runtime varies with model size and is approximately 2–6 hours on RTX PRO 6000 GPUs. For inference, the runtime also depends on model size: LoRe-Bench evaluation takes approximately 30 hours for all models, while general reasoning benchmark evaluation takes around 10 hours for all fine-tuned models, both conducted on L40S GPUs.

H.2 Implementation Details

Since DeepScaler does not come with predefined categories, we first annotate each question using GPT-4.1-mini to assign it to one of the following categories: Algebra (Prealgebra), Counting & Probability, Geometry, Number Theory, or Calculus (Precalculus). Based on these annotations, we construct sub-question and composite-question triplets by pairing questions from different categories.

For SFT, we perform a grid search over learning rates in {1e-6, 5e-6, 5e-5}, using a batch size of 8, gradient accumulation of 2, and a warmup ratio of 0.

H.3 Additional Proprietary Model Analysis

To further demonstrate the generalization of our conclusions beyond open-source LRMs, we conduct an evaluation on a representative proprietary model, *Gemini 3.1 Flash-Lite*. Our experiments reveal two important findings: (1) this proprietary model still fails to exhibit compositionality, and (2) it does not better satisfy monotonicity or compositionality than small-scale models.

Due to API constraints, proprietary systems typically return only *summarized reasoning*, rather than full reasoning traces, which may not accurately reflect the real-world inference cost. A more reliable evaluation and conclusion would require internal measurements from the commercial service providers.

Table 5: **Monotonicity Results on Proprietary Models.** We report Spearman correlations for reasoning compute and log accuracy on LoRe-Mono. *Lang.* denotes *Language*.

Models	Reasoning Compute ↑					Log Accuracy ↓				
	Math	**Science**	**Lang.**	**Code**	**All**	**Math**	**Science**	**Lang.**	**Code**	**All**
Gemini 3.1 Flash-Lite	0.976	0.916	0.969	0.991	**0.976**	-0.922	-0.690	-0.321	-0.874	**-0.935**

Table 6: **Compositionality Results on LoRe-Compo.** We calculate nMAD for reasoning compute (C_θ) and log accuracy ($\log A_\theta$).

Models	nMAD_{C_θ} ↓	$\mathrm{nMAD}_{\log A_\theta}$ ↓
Gemini 3.1 Flash-Lite	0.343	0.490

H.4 Additional Experimental Results

In Fig. 10, we further compare the reasoning compute compositionality of DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B before and after `SFT-Compo`. With `SFT-Compo`, the nMAD decreases, and the results align more closely with the $y = x$ line compared to their base counterparts.

H.5 Analysis of Prompt Variations

In this section, we analyze the sensitivity of our conclusions to prompt variations and perform controlled ablations to rule out potential confounding effects from different prompt formulations.

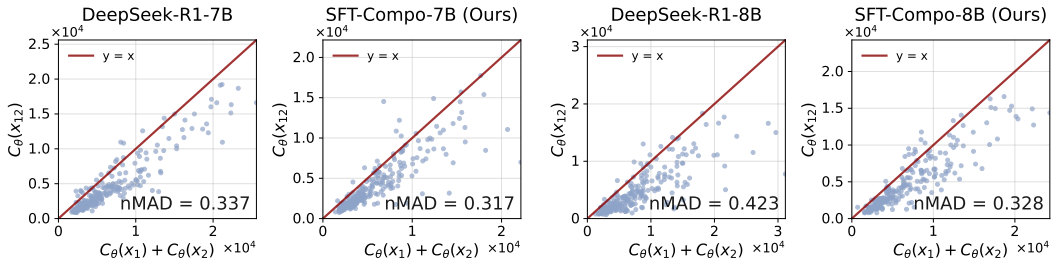


Figure 10: **Visualizations of Reasoning Compute Compositionality on DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B.**

H.5.1 Prompt Length Effects

In addition to the standard prompt from the official technical report that is used in this work, we have conducted additional experiments comparing it with two variants: one adding a brief role description, and another adding detailed reasoning instructions, listed as follows,

- **Standard:** `Please reason step by step, and put your final answer within \\boxed{}.`
- **Variant A:** `You are a language model trained to help users. Your role is to explore questions through thorough and careful reasoning before providing the final solutions. This includes analysis, summary, reflection and iteration for well-considered thinking. Please reason step by step, and put your final answer within \\boxed{}.`
- **Variant B:** `You are a language model trained to help users. Your role as an assistant involves thoroughly exploring questions through a systematic thinking process before providing the final precise and accurate solutions. This requires engaging in a comprehensive cycle of analysis, summarizing, exploration, reassessment, reflection, backtracing, and iteration to develop well-considered thinking process. Please reason step by step, and put your final answer within \\boxed{}.`

The results are shown in Tab. 7. In the LoRE-Compo evaluation, variations in prompt length lead to only minor variations, and nMAD_{C_θ} metric used to evaluate compositionality remains consistently large. This indicates that the models still fail to exhibit compositionality, regardless of prompt length change.

Table 7: **Prompt Ablation Results.** We report nMAD_{C_θ} under different prompt variants.

Models	Standard	Variant A	Variant B	Formatting
DeepSeek-R1-1.5B	0.528	0.535	0.481	0.478
Phi-4-mini	0.322	0.315	0.318	0.320
DeepSeek-R1-7B	0.337	0.338	0.339	0.344

H.5.2 Formatting Artifacts

To eliminate the effect of prompt formatting, we presented the sub-questions in the following format to make them consistent with the composite question.

- **Formatting Variant:** `Answer the following question. Q1: {sub-question}`

The results in Tab. 7 show that this modification leads to only minor variations in performance. This suggests that formatting differences are unlikely to be the primary cause of the observed compositionality failures.

H.5.3 Additional Reasoning Overhead

Concatenation may force the model to generate extra meta-reasoning to manage multiple tasks, which could potentially lead to $C_\theta(x_1 \oplus x_2) > C_\theta(x_1) + C_\theta(x_2)$. According to the empirical visualizations in Fig. 5 and Fig. 9, across all model sizes, we observed that most data points lie in the region where $C_\theta(x_1 \oplus x_2) < C_\theta(x_1) + C_\theta(x_2)$. Therefore, if meta-reasoning were induced, the discrepancy between $C_\theta(x_1 \oplus x_2)$ and $C_\theta(x_1) + C_\theta(x_2)$ would be even larger and the nMAD_{C_θ} metric would be worse than currently measured. This implies that the models' failure to exhibit compositionality may be even more severe than what is reflected in our current results. This further highlights the importance of studying reasoning laws.

H.6 Additional Evaluations Beyond Mathematics

To examine whether popular LRMs follow the compositionality beyond the mathematical domain, we evaluate three representative base models on a *widely used commonsense reasoning* benchmark, CommonsenseQA (Talmor et al., 2019).

Specifically, we sampled 500 CommonsenseQA problems and constructed a LoRe-Compo-Commonsense set using the same procedure as in LoRe-Compo. We then measured compositionality using the same metric nMAD_{C_θ}.

Table 8: **Compositionality Results on Commonsense Reasoning.** We report nMAD_{C_θ} on LoRe-Compo-Commonsense and compare with LoRe-Compo.

Models	LoRe-Compo-Commonsense	LoRe-Compo
DeepSeek-R1-1.5B	0.475	0.528
Phi-4-mini	0.479	0.322
DeepSeek-R1-7B	0.481	0.337

Our experiments reveal two important findings: (1) all these models still failed to exhibit compositionality on commonsense reasoning tasks, and (2) several models even perform much worse on non-mathematical reasoning tasks, such as Phi-4-mini and DeepSeek-R1-7B. These results suggest that compositionality failures are not limited to mathematical reasoning, but generalize to broader reasoning domains.

I Qualitative Examples of LoRe-Compo

We present an example generated by DeepSeek-Distill-1.5B and after SFT-Compo. For DeepSeek-Distill-1.5B, we report metrics for the question pair and the composite question with accuracies $A_\theta(x_1) = 1.0$, $A_\theta(x_2) = 1.0$, $A_\theta(x_{12}) = 0.25$. Reasoning compute is $C_\theta(x_1) = 2177.25$, $C_\theta(x_2) = 3546.50$, and $C_\theta(x_{12}) = 732.13$, showing that the model severely underthinks on the composite question and compromises the accuracy. After SFT-Compo, the composite accuracy improves to $A_{\tilde{\theta}}(x_{12}) = 0.875$ with reasoning compute as $C_{\tilde{\theta}}(x_{12}) = 4300.00$. We provide detailed reasoning processes of the two models below.

Example composite question

Q: Answer the following questions in order:
Q1. Pat is to select six cookies from a tray containing only chocolate chip, oatmeal, and peanut butter cookies. There are at least six of each of these three kinds of cookies on the tray. How many different assortments of six cookies can be selected? (Note that cookies of the same type are not distinguishable.)
Q2. Let $f(x) = x^{10} + 5x^9 - 8x^8 + 7x^7 - x^6 - 12x^5 + 4x^4 - 8x^3 + 12x^2 - 5x - 5$. Without using long division (which would be horribly nasty!), find the remainder when $f(x)$ is divided by $x^2 - 1$.

Example reasoning process (DeepSeek-R1-1.5B)

Q1. To determine the number of different assortments of six cookies, I need to consider the number of ways to select six cookies from three types: chocolate chip, oatmeal, and peanut butter. Since the cookies of the same type are indistinguishable, the order of selection doesn't matter.

I can model this problem using combinations with repetition. The formula for the number of ways to choose k items from n types is given by:

$$\binom{n+k-1}{k}$$

In this case, $n = 3$ (types of cookies) and $k = 6$ (cookies to select). Plugging in the values:

$$\binom{3+6-1}{6} = \binom{8}{6}$$

Calculating $\binom{8}{6}$ gives:

$$\binom{8}{6} = \frac{8!}{6!(8-6)!} = 28 \quad \text{[Correct Answer]}$$

Therefore, there are 28 different assortments of six cookies that can be selected.

Q2. To find the remainder when $f(x)$ is divided by $x^2 - 1$ without using long division, I can utilize the Remainder Theorem. The Remainder Theorem states that the remainder of a polynomial $f(x)$ divided by $x - a$ is $f(a)$. However, since the divisor here is $x^2 - 1$, which factors into $(x-1)(x+1)$, I need to evaluate $f(x)$ at the roots of x^2-1, which are $x = 1$ and $x = -1$.

The remainder will be a linear polynomial of the form $R(x) = ax + b$. To find a and b, I can set up two equations by plugging in $x = 1$ and $x = -1$ into $f(x)$: 1. $f(1) = a(1) + b$ 2. $f(-1) = a(-1) + b$

Calculating $f(1)$:

$$f(1) = 1^{10} + 5(1)^9 - 8(1)^8 + 7(1)^7 - 1(1)^6 - 12(1)^5 + 4(1)^4 - 8(1)^3 + 12(1)^2 - 5(1) - 5$$

$$f(1) = 1 + 5 - 8 + 7 - 1 - 12 + 4 - 8 + 12 - 5 - 5$$

$$f(1) = -1$$

Calculating $f(-1)$:

$$f(-1) = (-1)^{10} + 5(-1)^9 - 8(-1)^8 + 7(-1)^7 - 1(-1)^6 - 12(-1)^5 + 4(-1)^4 - 8(-1)^3 + 12(-1)^2 - 5(-1) - 5$$

$$f(-1) = 1 - 5 - 8 - 7 - 1 + 12 + 4 + 8 + 12 + 5 - 5$$

$$f(-1) = 13$$

Now, setting up the equations: 1. $a + b = -1$ 2. $-a + b = 13$
Solving these equations: Adding both equations: $2b = 12$ => $b = 6$ Substituting $b = 6$ into the first equation: $a + 6 = -1$ => $a = -7$

Thus, the remainder $R(x)$ is:

$$R(x) = -7x + 6 \quad \text{[Incorrect Answer]}$$

</think>

Example reasoning process (DeepSeek-R1-1.5B SFT-Compo)

Okay, so I have these two questions to solve. Let me start with the first one.

Q1. Pat is to select six cookies from a tray containing only chocolate chip, oatmeal, and peanut butter cookies. There are at least six of each of these three kinds of cookies on the tray. How many different assortments of six cookies can be selected? (Note that cookies of the same type are not distinguishable.)

Hmm, okay. So Pat is selecting six cookies, and the tray has three types: chocolate chip, oatmeal, and peanut butter. Each type has at least six cookies, so there's no shortage. Since the cookies of the same type are indistinct, we're dealing with combinations with repetition, I think.

I remember that the formula for combinations with repetition is C(n + k - 1, k), where n is the number of types and k is the number of items to choose. In this case, n is 3 (since there are three types of cookies) and k is 6 (since Pat is selecting six cookies). So plugging into the formula, it should be C(3 + 6 - 1, 6) = C(8, 6).

Wait, let me double-check. So, combinations with repetition: if order doesn't matter and items are indistinct except for type, yes, that's the case here. So, the formula is indeed C(n + k - 1, k). So, n=3, k=6, so 3 + 6 -1 = 8, so C(8,6). Alternatively, C(8,2) since combinations are symmetric, which is 28. So is that the answer?

But wait, let me think again. Since each type has at least six cookies, and we're selecting six, does that affect anything? Because if each type has at least six, then when we select six, the maximum number we can take from any one type is six, but since we're only selecting six, the maximum from any type is six, but since we're selecting six in total, the maximum from any type is six, but the minimum is zero.

But in this case, since we have at least six of each, so when we choose six, the number of ways is the same as the number of non-negative integer solutions to the equation:

x1 + x2 + x3 = 6, where x1, x2, x3 are the number of chocolate chip, oatmeal, and peanut butter cookies respectively.

Which is exactly the stars and bars problem, so C(6 + 3 -1, 3 -1) = C(8,2) = 28.

So yeah, the answer is 28. That seems right.

Q2. Let $f(x) = x^{10} + 5x^9 - 8x^8 + 7x^7 - x^6 - 12x^5 + 4x^4 - 8x^3 + 12x^2 - 5x - 5$.

Without using long division (which would be horribly nasty!), find the remainder when $f(x)$ is divided by $x^2 - 1$. Okay, so polynomial division. But $f(x)$ is a degree 10 polynomial, and we're dividing by $x^2 - 1$, which is also degree 2. So the remainder should be of degree less than 2, so it should be a linear polynomial, right? So it can be written as ax + b, where a and b are constants.

To find a and b, we can use the Remainder Theorem. Since $x^2 - 1$ factors as $(x - 1)(x + 1)$, the remainder when f(x) is divided by $x^2 - 1$ is the same as the remainder when f(1) and f(-1) are computed, and then we can set up equations to solve for a and b.

Wait, let me recall: if we have f(x) divided by $(x - c)(x - d)$, the remainder is R(x) = e x + f. So, to find R(x), we can evaluate f(c) and f(d), which gives us two equations: $f(c) = ec + f \quad f(d) = ed + f$

Then, solving these two equations gives us e and f. In our case, the divisor is $x^2 - 1$, which is $(x - 1)(x + 1)$. So, c =1 and d = -1.

[To be continued]

Therefore, we can compute f(1) and f(-1), set up the equations:
$f(1) = a(1) + b = a + b$
$f(-1) = a(-1) + b = -a + b$
Then, solving these two equations will give us a and b.

So, let me compute f(1) and f(-1). First, f(1):
$f(1) = (1)^10 + 5 * (1)^9 - 8 * (1)^8 + 7 * (1)^7 - (1)^6 - 12 * (1)^5 + 4 * (1)^4 - 8 * (1)^3 + 12 * (1)^2 - 5 * (1) - 5$

Simplify each term:
$1 + 5 - 8 + 7 - 1 - 12 + 4 - 8 + 12 - 5 - 5$

Let me compute step by step: Start with 1.
$1 + 5 = 6 \quad 6 - 8 = -2 \quad -2 + 7 = 5 \quad 5 - 1 = 4 \quad 4 - 12 = -8 \quad -8 + 4 = -4$
$-4 - 8 = -12 \quad -12 + 12 = 0 \quad 0 - 5 = -5 \quad -5 - 5 = -10$
So $f(1) = -10$. Similarly, compute $f(-1)$:
$f(-1) = (-1)^10 + 5 * (-1)^9 - 8 * (-1)^8 + 7 * (-1)^7 - (-1)^6 - 12 * (-1)^5 + 4 * (-1)^4 - 8 * (-1)^3 + 12 * (-1)^2 - 5 * (-1) - 5$

Simplify each term:
$(-1)^10 = 1 \quad 5 * (-1)^9 = 5 * (-1) = -5 \quad -8 * (-1)^8 = -8 * (1) = -8$
$7 * (-1)^7 = 7 * (-1) = -7 \quad -(-1)^6 = -1 \quad -12 * (-1)^5 = -12 * (-1) = 12$
$4 * (-1)^4 = 4 * (1) = 4 \quad -8 * (-1)^3 = -8 * (-1) = 8 \quad 12 * (-1)^2 = 12 * (1) = 12 \quad -5 * (-1) = 5$
-5 remains as is.

So let's write all terms: $1 - 5 - 8 - 7 - 1 + 12 + 4 + 8 + 12 + 5 - 5$

Let me compute step by step:

Start with 1.

$1 - 5 = -4 \quad -4 - 8 = -12 \quad -12 - 7 = -19 = \quad -19 - 1 = -20$
$-20 + 12 = -8 \quad -8 + 4 = -4 \quad -4 + 8 = 4 \quad 4 + 12 = 16 \quad 16 + 5 = 21 \quad 21 - 5 = 16$
So $f(-1) = 16$. Therefore, we have: $f(1) = a + b = -10 \quad f(-1) = -a + b = 16$

So now, we have the system of equations: 1) $a + b = -10$ 2) $-a + b = 16$. Let me solve this.

Add equations 1 and 2: $(a + b) + (-a + b) = (-10) + 16$
Simplify: $0a + 2b = 6$ So $2b = 6 => b = 3$. Then, plug b=3 into equation 1: $a + 3 = -10 => a = -13$. So the remainder is $ax + b = -13x + 3$

Therefore, the remainder when f(x) is divided by $x^2 - 1$ is $-13x + 3$. Let me double-check my computations for f(1) and f(-1), just to make sure I didn't make any arithmetic mistakes.

Recomputing f(1):
$1 + 5 = 6 \quad 6 - 8 = -2 \quad -2 + 7 = 5 \quad 5 - 1 = 4 \quad 4 - 12 = -8 \quad -8 + 4 = -4$
$-4 - 8 = -12 \quad -12 + 12 = 0 \quad 0 - 5 = -5 \quad -5 - 5 = -10$

Yes, that's correct. **Recomputing $f(-1)$:** $1 - 5 = -4 \quad -4 - 8 = -12 \quad -12 - 7 = -19 \quad -19 - 1 = -20 \quad -20 + 12 = -8 \quad -8 + 4 = -4 \quad -4 + 8 = 4 \quad 4 + 12 = 16 \quad 16 + 5 = 21 \quad 21 - 5 = 16$ Yes, that's correct.

So the equations are: $a + b = -10 \quad -a + b = 16$

[To be continued]

Adding them gives $2b = 6 => b = 3$, then a= -13. So the remainder is -13x +3. I think that's correct.

Final Answer

Q1: $\boxed{28}$ [Correct Answer]

Q2: $\boxed{-13x + 3}$ [Correct Answer]

</think>

J Licenses for Existing Assets

J.1 Model Assets

We utilize a set of publicly available models in our work. The DeepSeek-R1-Distill series models (Qwen-1.5B, Qwen-7B, Qwen-14B, and Llama-8B) are released under the MIT License and are distilled from the Qwen-2.5 series (Apache License 2.0) or Llama-3.1-8B-Base (Llama 3.1 Community License). Additionally, Phi-4-mini-reasoning is released under the MIT License. OpenReasoning-Nemotron-14B is governed by the Creative Commons Attribution 4.0 International License (CC-BY-4.0), with additional Apache License 2.0 information. Sky-T1-32B-Preview, Qwen3-Next-80B-A3B-Thinking, and Thinkless-1.5B-RL-DeepScaleR are released under the Apache License 2.0. AdaptThink-7B-delta0.05 is released under the MIT License. We comply with all respective license terms in our use of these models.

J.2 Data Assets

We employ publicly available datasets in our experiments. GSM8K, OlympiadBench, DeepScaler, and CommonsenseQA are released under the MIT License, which permits unrestricted use, modification, and redistribution. MATH500 is derived from the MATH dataset, which is also released under the MIT License. AIME24 and AIME25 are copyrighted by the Mathematical Association of America. The AMC23 dataset does not have an explicitly specified license, so we treat it as having an unspecified license and exercise caution in its usage. We ensure full compliance with the respective license terms of all datasets used.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize the main contributions: the theoretical framework, LoRe-Bench, and SFT-Compo. These claims are supported by the theoretical formulation in Section 2 and empirical validation in Sections 3, 4, 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses its limitations in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical assumptions and formulations are presented in Section 2, and the complete proof is provided in Appendix E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The implementation details are provided in Section 3 for LoRe-Bench evaluation, and in Section 4 for SFT-Compo. These sections cover the benchmark construction, evaluation setup, model configurations, and training procedure necessary to reproduce the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized code repository is referenced in Appendix A. The paper also provides detailed implementation information in Section 3.3, Section 5.1, and Appendix H.2 to support reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the experimental settings in Section 3.3, Section 5.1, and Appendix H.2, including the evaluated models, sampling hyper-parameters, evaluation metrics, and fine-tuning details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports variability using shaded regions that denote the standard deviation across samples, as shown in Fig. 4 and Fig. 8.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the compute resources and execution time in Appendix H.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This paper follows the NeurIPS Code of Ethics. It studies reasoning models on mathematical and scientific reasoning benchmarks and does not involve human subjects, private data, or high-risk deployment.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix B.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release models or datasets that pose a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The paper provides a detailed description of the licenses for all models and datasets in Section J, and ensures compliance with their respective terms of use.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces new assets, including the LoRe-Bench benchmark, the SFT-Compo data construction pipeline, and four fine-tuned models obtained via SFT-Compo. Their construction, training, and usage are documented in Section 3, Section 4, and Appendix H.2; an anonymized repository is referenced in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper studies LRMs as an important component and describes their usage in Section 3, 4, 5, and Appendix H.2.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.