Exploring Layer Activation Dynamic of CoT via Knowledge Probe

Chuanxin Zhang[1], **Jiajun Liu**[1], **Yao He**[3], **Wenjun Ke**[1,2*],
Peng Wang[1,2], **Yankun Le**[4], **Sirui Liu**[1], **Zhaoyu Yang**[1]
[1]School of Computer Science and Engineering, Southeast University, China
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
[3]Institute of Collaborative Innovation, University of Macau, Macau, China
[4]School of Computer Science and Engineering, Northeastern University, Shenyang, China
{kewenjun, chuanxinzhang, jiajliu, pwang, 213230729, 220252379}@seu.edu.cn
mc46477@um.edu.mo,2401851@stu.neu.edu.cn

Abstract

Chain-of-thought (CoT) reasoning has emerged as a crucial paradigm for enhancing large language model (LLM) performance on multi-step reasoning tasks. However, the internal mechanisms by which LLMs invoke knowledge and propagate information across different steps of the CoT are poorly understood. To fill this gap, we propose a multi-stage probing framework that enforces structured reasoning with three explicit stages: keyword extraction, theorem generation, and computation execution. The framework integrates attention knockout to trace cross-layer information flow and theorem probing to examine how specific contents are encoded within representations. To enable controlled and stage-aligned analysis, we construct a structured CoT dataset that covers the mathematics and physics domains. Experiments on four instruction-tuned LLMs reveal distinct stage-specific patterns. First, keyword information is progressively aggregated into the final token in later layers. Second, theorem semantics are encoded in the mid-to-late layers and undergo two stages of propagation. Finally, parameter substitution is achieved through joint extraction by the final token and other tokens. The first parameter predominantly relies on the final token, whereas later parameters increasingly depend on information extracted by other tokens. Overall, our findings shed light on the neural implementation of CoT reasoning and provide actionable insights for developing more interpretable and reasoning-capable LLMs. We further evaluate a free-form prompting setting without labeled fields and observe consistent qualitative trends.

1 Introduction

Chain-of-thought (CoT) (Wei et al., 2022; Vig et al., 2020b; Zhang and Nanda, 2023; Kriegeskorte et al., 2008) has emerged as an efficient paradigm to improve large language model (LLM) reasoning ability (Deng et al., 2024; Cheng and Van Durme, 2024; Wang and Zhou, 2024; Xu et al., 2025) by explicitly decomposing the reasoning process into intermediate steps. Although recent studies (Dutta et al., 2024; Yu and Ananiadou, 2024; Geva et al., 2021; Chen et al., 2024; Geva et al., 2023) have uncovered certain aspects of the internal mechanisms underlying LLM reasoning, it remains unclear how multi-stage knowledge invocation and information flow are implemented in CoT reasoning.

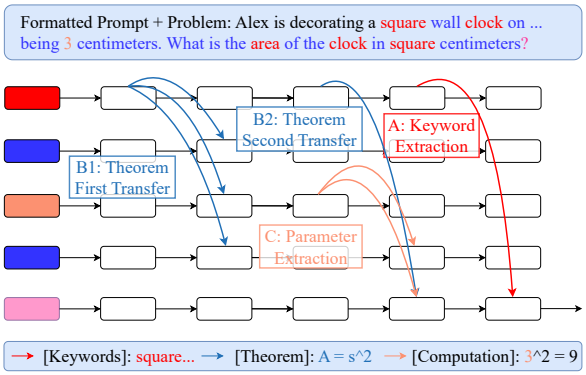

Figure 1: Illustration of our motivations: keyword information converges to the final token in later layers (A); theorem information is relayed from non-final to final tokens and re-extracted in later layers (B1, B2); parameters are jointly inferred by final and non-final tokens in middle-to-late layers (C).

Existing research on the interpretability of reasoning tasks mainly follows two directions. One line of work relies on empirical analysis based on explicit CoT outputs (Deng et al., 2024; Wang et al., 2023; Zheng et al., 2025). By observing the intermediate reasoning chains generated by the model, these studies attempt to infer the knowledge invocation process (Ling et al., 2023). Such methods can verify the necessity of reasoning steps through component ablation or output reformatting, but they only reveal coarse-grained correlations and fail to capture the dynamic mechanisms inside the model (Hu et al., 2025).

*Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11811–11830
July 2-7, 2026

To overcome these limitations, a second line of work turns to token-level activation analysis. These methods examine internal activations of individual tokens within model layers to trace how information is processed and propagated inside (Geva et al., 2023; Qiang et al., 2022; Dai et al., 2022; Meng et al., 2022). They offer a finer-grained view of internal dynamics and have been applied to probing factual knowledge and token interaction.

However, most existing interpretability studies remain focus on factual recall, with an emphasis on single-step retrieval and static token relationships (Feng et al., 2023; Chen et al., 2025). This narrow focus leaves the dynamic, multi-stage knowledge invocation and information flow that support chain-of-thought reasoning unexplored.

To fill this gap, we propose a multi-stage probing framework for structured CoT reasoning tasks that require explicit theorem invocation and intermediate steps, encompassing keyword extraction, theorem generation, and computation execution. The framework integrates two complementary techniques: (i) attention knockout, which blocks critical tokens during generation to trace inter-layer information flow (Dutta et al., 2024), and (ii) knowledge probing, which inspects how theorem semantic encoded across layers (Pimentel et al., 2020; Zhu and Rudzicz, 2020; Dalvi et al., 2020).

To probe reasoning dynamics, we construct a structured CoT dataset in which problem instances are generated from mathematics and physics theorems with controlled output formats (Section 3.1). Our experiments show that LLMs exhibit cross-domain consistent information extraction mechanisms across mathematics and physics, while revealing distinct stage-wise patterns in different stages of CoT reasoning (Figure. 1). These trends remain consistent under free-form prompting without labeled fields (Appendix B):

- **Keyword extraction stage**: keyword-related information is primarily transmitted to the final token in the later layers (Section 3.2).
- **Theorem generation stage**: theorem-related information at theorem-related token positions is first captured by non-final tokens in the early layers and relayed to the final token for prediction. In the later layers, it is re-extracted at these positions and directly transmitted to the final token; notably, semantic features of the theorem are encoded in the hidden states at intermediate layers (Section 3.3).
- **Parameter substitution stage**: parameter values are not matched one-to-one with the formal parameters in the formula. They are jointly extracted in the middle and later layers by the final token and non-final tokens, with the first parameter mainly captured by the final token, while later parameters are primarily extracted by non-final tokens. The model's numerical computation using the generated formula depends on the model and the task type (Section 3.4).

2 Methods

2.1 Preliminary

To understand the information flow mechanism of the model in reasoning tasks, we adopted the attention blocking method (Geva et al., 2023). It blocks specific attention connections in the Multi-Head Self-Attention (MHSA) sublayer, which is the only component that transmits information across positions, to evaluate their effect on the model's predictive ability. Specifically, we set the attention weights between different tokens in designated layers to negative infinity, thereby blocking these connections, and analyzed the resulting changes in prediction accuracy to determine whether critical information propagates across these layers. This methodology enables us to pinpoint the propagation pathways of key information throughout the model's reasoning process.

For a given input and layer ℓ, we block the attention connection between positions r and c in that layer. Formally, at layer $\ell + 1$, let r and c be the indices of the two tokens between which the attention is removed. The knockout operation updates the attention mask as:

$$M_{rc}^{\ell+1,j} = -\infty \quad \forall j \in [1, H] \tag{1}$$

where H is the number of attention heads and $M^{\ell+1,j}$ denotes the additive attention mask for head j at layer $\ell + 1$. This assignment effectively blocks the information flow from position c to position r across all heads.

2.2 Multi-stage Probing Framework

Since CoT outputs exhibit randomness in theorem reasoning, we fix the output format to enable probing at consistent token positions. We additionally construct a free-form prompting variant without labeled fields for comparison. To avoid introducing

additional knowledge, we adopt a prompt-based approach rather than fine-tuning. Specifically, the structured output sequence is illustrated in Figure 1. We enforce a structured output sequence where the model first declares the required formula name, then outputs its formal definition, and finally performs parameter instantiation for computation. Concrete formatted prompts are provided in Appendix A.1.

2.3 Blocking Configurations

To address the complexity of information propagation in CoT reasoning, we explicitly consider its potential pathways. These may involve direct extraction by the final token, relay through intermediate tokens, joint extraction by both final and non-final tokens, or alignment with specific source positions. We define *Other tokens* as all tokens positioned after the target subset and before the final token. The set of *All tokens* additionally includes the final token itself. Accordingly, we design four complementary blocking configurations:

- **Last-Sub blocking:** blocking attention from the final token to the target subset, thereby isolating the direct read-out effect and testing whether predictions rely on direct extraction.
- **Other-Sub blocking:** blocking attention from *Other tokens* to target subset to evaluate relay pathways, where information is transmitted via intermediate tokens to the final token.
- **All-Sub blocking:** blocking attention from both the *final* and *Other tokens* to target subset. This provides the upper bound of dependency on the target subset and serves as an attribution anchor for comparison.
- **Specific blocking:** blocking attention from a designated source set to the target subset, enabling hypothesis testing of whether explicit mappings or one-to-one dependencies exist.

For Last-Sub blocking, we block the attention from the final token to the positions of a target subset in the input. Let $S \subset [1, N)$ denote the set of positions for the target subset within the input sequence. For the i-th subset position S_i, we block attention from the last position N, to S_i:

$$M^{\ell+1,j}_{N,S_i} = -\infty \quad \forall j \in [1, H] \tag{2}$$

For Other-Sub blocking, we block attention from the position of each *Other token* to S_i:

$$M^{\ell+1,j}_{(S_{i+1},S_{i+2},\ldots,N-1),S_i} = -\infty \quad \forall j \in [1, H] \tag{3}$$

For All-Sub blocking, we block attention from the position of each *All token* to S_i:

$$M^{\ell+1,j}_{(S_{i+1},S_{i+2},\ldots,N),S_i} = -\infty \quad \forall j \in [1, H] \tag{4}$$

For Specific blocking, we block the attention from a designated set of source positions to the i-th substitution position. Let $\mathcal{P} \subset [1, N)$ denote the set of specific source positions, we block attention from $p \in \mathcal{P}$ to S_i:

$$M^{\ell+1,j}_{p,S_i} = -\infty \quad \forall j \in [1, H] \tag{5}$$

2.4 Theorem Probe

To investigate whether LLMs activate theorem-related knowledge during formula generation (Hendrycks et al., 2021), we introduce the Theorem Probe technique. This method takes the hidden states from arbitrary layers as inputs and employs a lightweight classifier to predict the corresponding theorem category (Azaria and Mitchell, 2023), thereby evaluating whether theorem-related semantics are encoded in the representations (Conneau et al., 2018).

In order to probe how theorem-related semantics are encoded in hidden representations, we adopt a Multi-Layer Perceptron (MLP) as the probing model. The input is the last-token hidden vector $\mathbf{h}_i$, normalized and nonlinearly transformed to produce a categorical distribution over theorem classes. The distribution p_i is computed as:

$$p_i = \sigma(\mathbf{W}_2\ (\mathrm{GELU}(\mathbf{W}_1\,\mathbf{h}_i + \mathbf{b}_1)) + \mathbf{b}_2) \tag{6}$$

where $\sigma(\cdot)$ denotes the softmax function. The model parameters are $\mathbf{W}_1 \in \mathbb{R}^{1024\times d}$, $\mathbf{W}_2 \in \mathbb{R}^{C\times 1024}$, $\mathbf{b}_1 \in \mathbb{R}^{1024}$, and $\mathbf{b}_2 \in \mathbb{R}^{C}$, where d is the hidden size of the language model and C is the number of theorem categories.

We extract hidden states across different layers as training samples and pair them with corresponding theorem labels. The training objective minimizes the cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\,\log p_{i,c} \tag{7}$$

where $y_{i,c}$ denotes the one-hot ground-truth label and $p_{i,c}$ the predicted probability of class c. This probing framework enables quantitative analysis of how theorem knowledge is encoded across layers and reveals knowledge flow in formula generation.

3 Experiments & Findings

3.1 Experimental Setup

Dataset. We construct a reasoning-formatted CoT dataset to probe theorem-level reasoning behaviors. The dataset covers two domains, mathematics and physics, with 50 distinct theorems per domain. For each theorem, we prompt an LLM to generate ten unique problem instances, resulting in a total of 1,000 problems. Concrete dataset examples are provided in Appendix A.2. In addition to the structured dataset with controlled output formats, we also construct a free-form dataset variant without labeled fields, provided in Appendix B. Supplementary sensitivity analyses of key experimental settings are reported in Appendix D.

Models. To ensure generalizability across different model families, we select four instruction-tuned LLMs as backbones in our experiments, namely Meta-Llama-3-8B-Instruct (LLaMA3-8B) (AI@Meta, 2024), Meta-Llama-3-70B-Instruct (LLaMA3-70B) (AI@Meta, 2024), DeepSeek-R1-Distill-Qwen-7B (R1-Qwen7B) (DeepSeek-AI, 2025), and Qwen2.5-Math-7B-Instruct (Qwen2.5-Math-7B) (Yang et al., 2024), respectively. The results on LLaMA3-70B are reported in Appendix C.

3.2 Keyword Extraction

To investigate how keyword-related information is propagated when language models extract keywords from problem statements, we apply three types of blocking operations over k consecutive layers and examine their effects on keyword extraction at different positions. We set $k = 10$.

Figure 2 illustrates the experimental results. Across all models tested, blocking the attention from the final token to the keyword tokens in the later layers causes a sharp drop in the prediction probability of the corresponding keywords, with reductions of 70% to 80%. In contrast, blocking the attention from *other tokens* to keyword tokens across all layers has almost no effect. These findings indicate that keyword information is directly transmitted to the final token in the later layers. They further suggest that similar mechanisms are shared across different models and domains.

3.3 Theorem Generation

Keyword extraction for theorem generation. For theorem generation, the model is expected to extract theorem-related information to determine which theorem to produce. To examine how this information propagates, we apply the same blocking strategies detailed in section 3.2 to theorem tokens and observe the resulting changes in the model's predictive performance. We set $k = 10$.

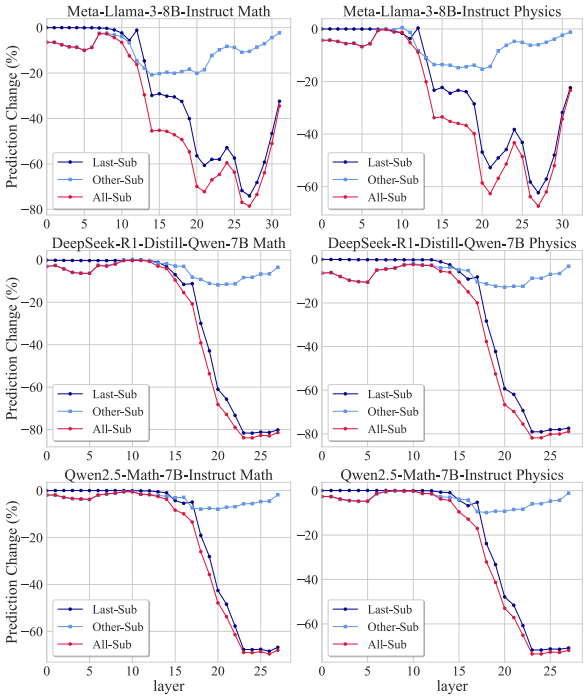

Figure 2: Relative change in prediction probability when blocking attention to keyword token positions during keyword extraction across layers and models.

The results are summarized in Figure 3. We observe that across different domains and models, information from theorem tokens propagated twice: once in the early layers and again in the later layers. The prediction probability trends under All-Sub blocking align closely with those of Other-Sub blocking in early layers (within 10 layers), with reductions of about 10%–20%. In contrast, in the middle-to-late layers (after layer 15), the trends become consistent with Last-Sub blocking. These findings indicate that information localized at theorem-token positions is initially captured by *other tokens* in the early layers and relayed to the final token for prediction, whereas in the later layers, the information at these positions is re-extracted and directly transmitted to the final token.

Encoding theorem information in LLMs. We investigate how LLMs encode theorem information in their hidden states on mathematical and physical reasoning datasets. To this end, we train theorem probes on hidden states of tokens generated during

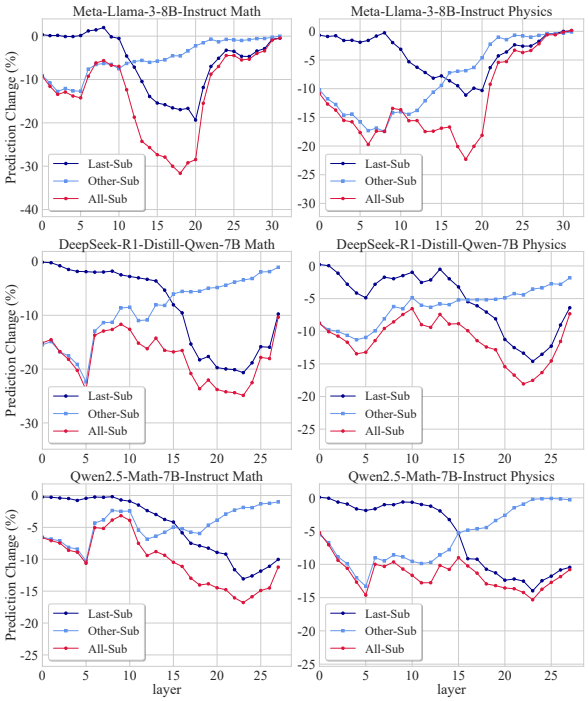

Figure 3: Relative change in prediction probability when blocking attention to theorem-related token positions during theorem generation across layers.

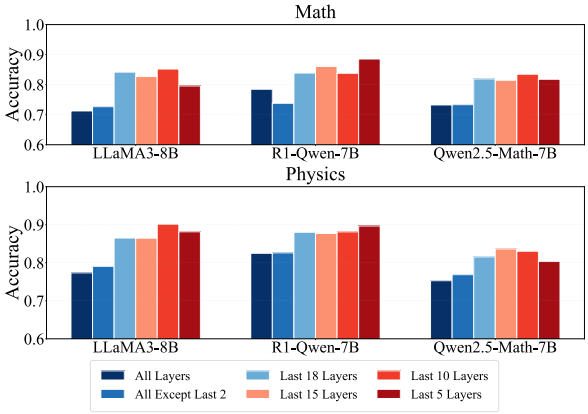

Figure 4: Accuracy of probes trained on hidden states from different layer ranges for detecting theorem tokens.

theorem prediction, where performance directly reflects the extent to which theorem information is encoded. Inspired by prior findings that LLMs capture different features at different layers (Geva et al., 2023), we construct probes for each model with multiple layer configurations, including all layers, all except the final two layers, last 18 layers, last 15 layers, last 10 layers, and last 5 layers.

We train and evaluate these probes on hidden states from selected layer configurations, and the results are presented in Figure 4. The detailed training configuration is provided in Appendix A.3. On the LLaMA3-8B mathematics dataset, using representations from all layers or all except the final two layers yields notably lower accuracies (0.779 and 0.780, respectively). In contrast, probes trained on the last 18, last 15, and last 10 layers achieve the highest accuracies, with the last 10 layers reaching the best performance (0.889). Similar trends are observed on the physics dataset and other models. These findings suggest that theorem-related information is most linearly separable in middle-to-late layers, while incorporating early-layer representations slightly degrades probe performance.

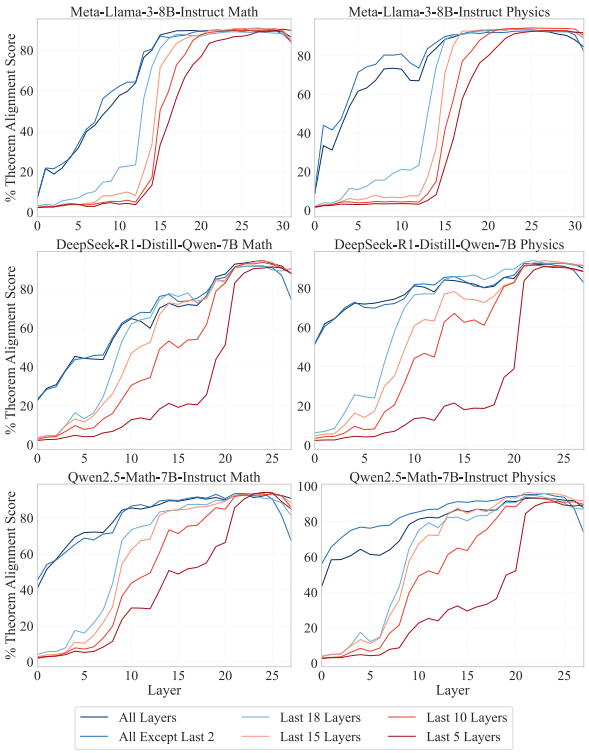

Figure 5: Performance of probes trained on different layer configurations.

Figure 5 presents probe results under different layer configurations during the generation of theorem tokens. Across architectures and domains, we observe a three-stage encoding pattern for theorem-related information, suggesting a fundamental property of how LLMs process information.

(i) **Early phase.** Probes trained on representations including substantial early-layer content, such as all layers and all except the last two layers, already achieve non-trivial accuracy from shallow depths, around 60%. A possible reason is that shallow representations allows the probe to exploit surface-level statistical cues in the input text.

(ii) **Mid-to-late phase.** When probes are trained on representations restricted to deeper middle layers, namely the last 18, last 15, and last 10 lay-

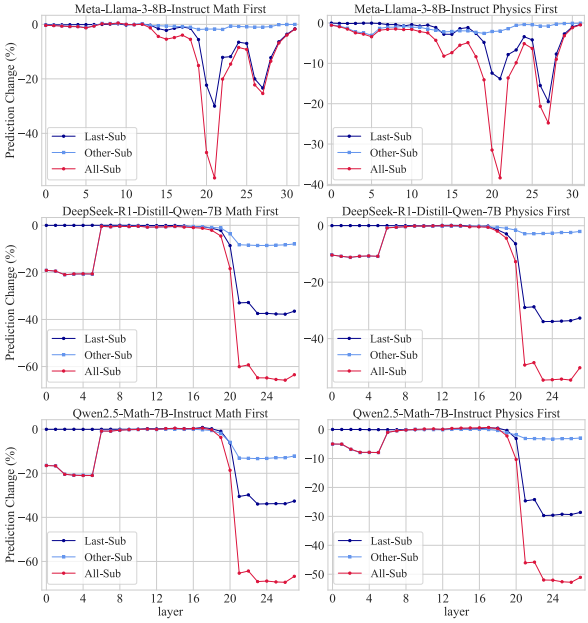

Figure 6: Relative change in prediction probability when blocking attention to parameter-token positions during instantiation of the first parameter across layers.

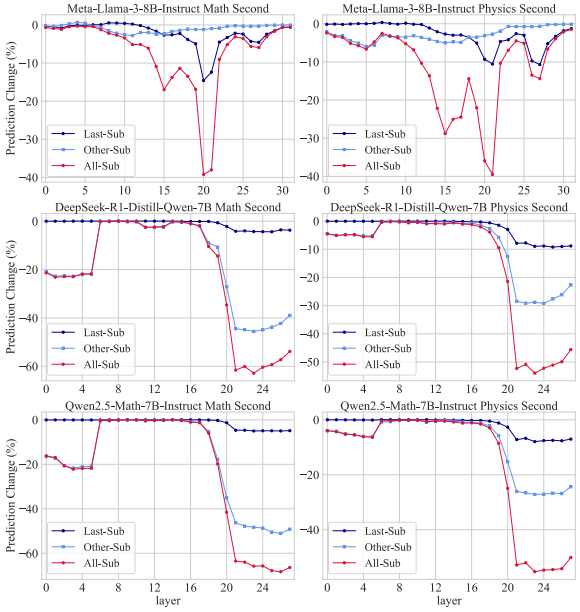

Figure 7: Relative change in prediction probability when blocking attention to parameter-token positions during instantiation of the second parameter across layers.

ers, their accuracies rise steeply and almost synchronously, then converge to a similar plateau. For example, on LLaMA3-8B with the Math dataset, accuracy increases from roughly 10% to between 60% and 80% between layers 10 and 15. This convergence indicates that theorem-specific information becomes robust and linearly decodable within a shared, processing bandwidth in the model's deeper layers, where abstract conceptual knowledge about theorems is integrated and stabilized.

(iii) **Late phase.** Probes confined to the very top layers (last 5) exhibit a notably delayed performance lift. For instance, on the R1-Qwen7B model with the Math dataset, accuracy does not exceed 60% until around layer 20. This delay suggests a functional shift at the topmost layers: rather than further refining abstract theorem representations, these layers are likely repurposed for output formatting, sequence planning, and alignment with instruction-following constraints.

Overall, our findings indicate that the mid-to-late layers are the key locus for effectively encoding abstract theorem information.

3.4 Parameter Extraction and Instantiation

This section aims to investigate the mechanism of parameter substitution, with a particular focus on how LLMs extract parameter-specific information and integrate it with the invoked theorem.

Parameter Extraction. We employ the attention knockout technique ($k = 10$) to trace the propagation paths of information originating from parameter tokens. Since most problems involve multiple parameters, we conduct analyses for the first, second, and third parameters. This design allows us to examine whether the model adopts a consistent mechanism when handling parameter substitutions.

Figure 6 reports the results for the first parameter. On LLaMA3-8B, the positions of probability drops under Last-Sub and All-Sub blocking largely coincide. Last-Sub blocking produces a moderate reduction of around 20% in the middle layers, whereas All-Sub blocking leads to a much stronger decline exceeding 50%. In contrast, Other-Sub blocking induces almost no observable change. A similar trend is consistently observed in R1-Qwen7B and Qwen2.5-Math-7B.

Across all models and datasets, the decline under All-Sub blocking is substantially larger than under Last-Sub alone. These results indicate that parameter substitution primarily occurs in the middle-to-late layers, where parameter information is extracted jointly by the final token and *other tokens*, though with a dominant reliance on the final token.

Interestingly, the extraction mechanism diverges for subsequent parameters. Figures 7 and 8 present the results for the second and third parameters. On LLaMA3-8B, the magnitudes and positions of probability drops remain broadly similar. However, for

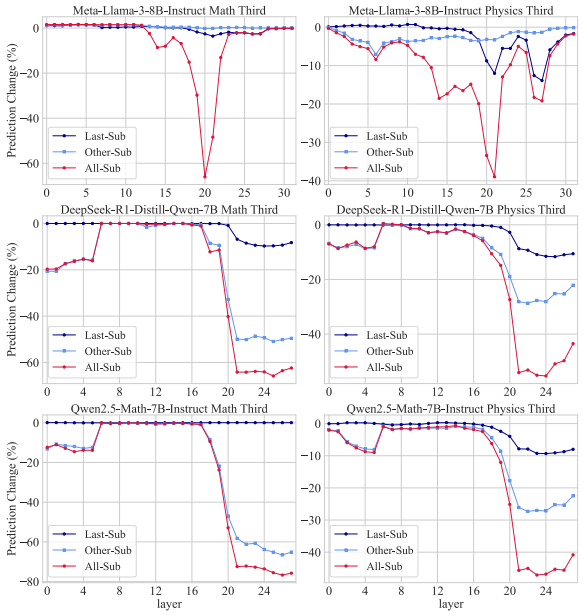

Figure 8: Relative change in prediction probability when blocking attention to parameter-token positions during instantiation of the third parameter across layers.

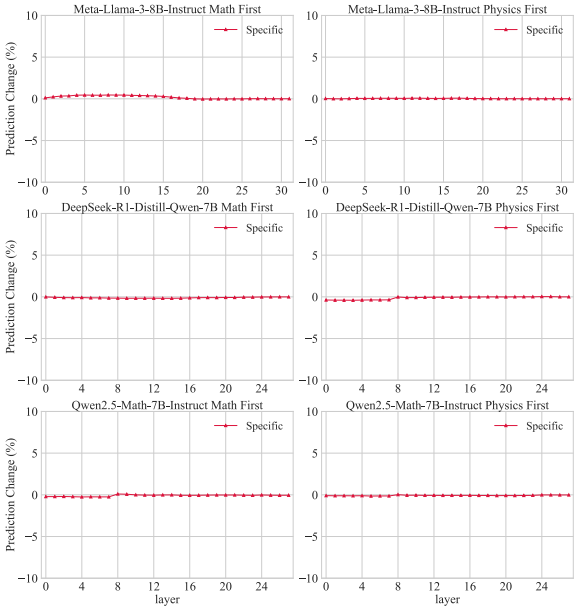

Figure 9: Relative change in prediction probability when blocking attention between parameter tokens and theorem tokens during instantiation of the first parameter across layers.

R1-Qwen7B and Qwen2.5-Math-7B, the effect of Last-Sub blocking becomes negligible, while the impact of Other-Sub blocking grows substantially. These findings suggest that, for later parameters, the models increasingly rely on information captured by *Other tokens* rather than the final token.

Parameter–Theorem Matching. To investigate whether the model explicitly matches problem parameters with the invoked theorem when generating parameter tokens, we design Specific blocking experiment. In this setting, we block the attention connections between parameter tokens in the problem statement and theorem tokens over $k = 10$ consecutive layers, thereby eliminating direct information exchange between the two.

Figure 9 shows that blocking attention from theorem tokens to parameter tokens has little effect on prediction probabilities in both early and late layers. This suggests that parameter extraction does not rely on direct theorem-to-parameter information flow, but is instead inferred implicitly from the problem context in the middle-to-late layers through the joint contribution of the final token and other contextual tokens.

Evolving Reliance on Explicit Formula Content. To test whether the model uses generated formulas during parameter instantiation, we adopt a 2×2 design. One factor is parameter complexity, with levels "many" and "few". The other factor is the

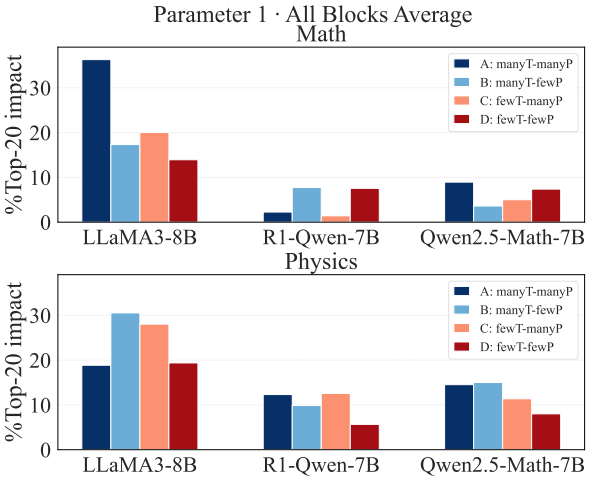

Figure 10: Sensitivity of first-parameter extraction to theorem-token blocking under varying theorem and parameter complexities.

number of theorem tokens, also with levels "many" and "few". This yields four groups: manyT–manyP, manyT–fewP, fewT–manyP, and fewT–fewP. We then apply All-Sub blocking to the theorem content and measure the resulting changes in prediction probabilities across layers. To quantify effects, we select the top 20 layers with the largest probability shifts and report their mean as the sensitivity metric for each group.

The result of the first parameter is shown in Figure 10, and the second and third parameter results are presented in Figure 11 and Figure 12.

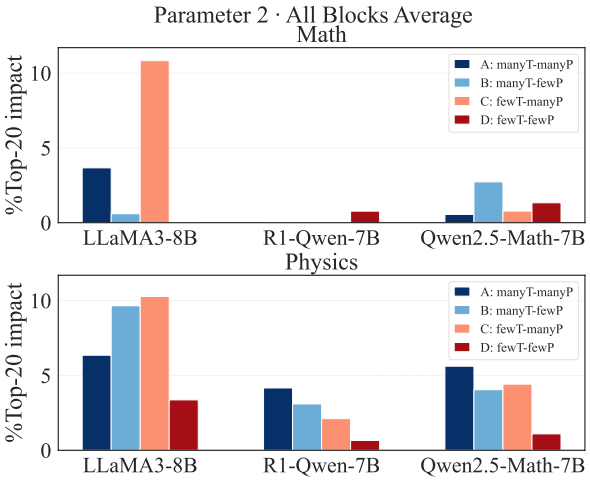

Figure 11: Supplementary sensitivity analysis of second-parameter extraction to theorem-token blocking across complexity groups.

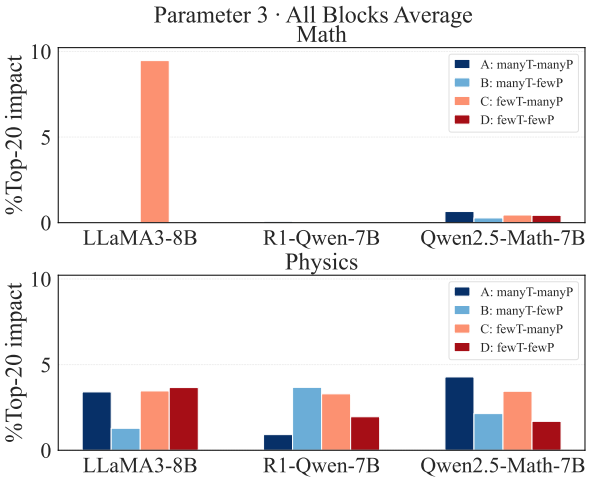

Figure 12: Supplementary sensitivity analysis of third-parameter extraction to theorem-token blocking across complexity groups.

LLaMA3-8B shows higher sensitivity to All-Sub blocking on the first parameter than the other two models. In mathematics tasks, the manyT–manyP group drops by more than 30 percent. R1-Qwen7B and Qwen2.5-Math-7B each drop by less than 10 percent under the same condition. Physics tasks exhibit the same pattern. These results indicate that LLaMA3-8B relies heavily on explicit formula content when instantiating the first parameter, especially when theorems are complex and parameters are numerous, whereas the other two models show minimal sensitivity to theorem blocking. For the second and third parameters, all groups and models exhibit drops below 10%, suggesting that the subsequent parameter instantiations depend far less on direct information from theorem tokens.

4 Related Work

Recent research is increasingly focused on the interpretability of reasoning in LLMs. One line of work examines CoT outputs to assess the faithfulness of reasoning chains (Jie et al., 2024; Paul et al., 2024; Lyu et al., 2023; Lanham et al., 2023). Empirical analyses show models often exhibit an early response phenomenon, where the final answer is determined before intermediate steps are generated (Wang et al., 2025; Liu and Wang, 2025). Perturbing or removing parts of the chain frequently has little effect on the outcome, raising doubts about the authenticity of CoT reasoning (Lanham et al., 2023; Turpin et al., 2023; Barez et al., 2025). To address this, methods such as Thought Anchors (Bogdan et al., 2025) have been introduced to identify critical reasoning steps disproportionately influence subsequent predictions.

Another line of research investigates attention-based information flow. Because transformer attention routes dependencies between tokens, interventions on attention provide a natural way to trace reasoning dynamics (Rohekar et al., 2023; Nam et al., 2025). For instance, Geva et al. (2023) identified a refine–extract mechanism in factual recall, where subject tokens are enriched in early MLP layers and later queried by the final token through specific attention heads. Extending this idea, Bogdan et al. (2025) applied attention masking to quantify the causal contribution of reasoning sentences, while Ortu et al. (2024) showed that particular attention heads can suppress factual memory in favor of misleading context. This effect can be reversed by modifying only a few heads. These studies demonstrate that attention interventions serve as powerful tools for path tracing within reasoning.

A complementary direction probes representations. Geva et al. (2021) interpreted feed-forward layers as key–value memories, Dai et al. (2022) identified knowledge neurons via gradient attribution, and Meng et al. (2022) combined activation patching with causal tracing to propose ROME for editing factual knowledge. Causal mediation analysis provides a lens for attributing behavioral effects to internal pathways (Vig et al., 2020a). Representational similarity analysis compares activation geometries to characterize how semantic structure evolves across layers (Kriegeskorte et al., 2008).

5 Conclusion

This work advances understanding of how large language models implement chain-of-thought reasoning. By introducing a multi-stage probing framework, we reveal that reasoning involves distinct mechanisms at various stages, rather than being a single unified process. Our study highlights structured information flow dynamics, offering new insights into the mechanisms that underlie multi-step reasoning. These findings contribute to both the interpretability of language models and the development of more reliable reasoning systems.

Limitations

Our analysis is limited to theorem-grounded math/physics with stage-aligned control. Although we confirm the qualitative patterns under free-form prompting and on LLaMA3-70B, free-form results rely on a small annotated set and may not generalize to less structured, real-world reasoning. Future work could extend evaluation to a wider range of tasks and models, and develop automated alignment for interventions and probing.

Acknowledgement

We would like to thank the anonymous reviewers for their insightful comments. This work was supported by National Science Foundation of China (Grant Nos.62376057), the Start-up Research Fund of Southeast University (RF1028623234) and the Big Data Computing Center of Southeast University.

References

Moonshot AI. 2025. moonshot-v1-8k model description. https://platform.moonshot.ai/. Accessed October 2025.

AI@Meta. 2024. Llama 3 model card.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976.

Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, and 1 others. 2025. Chain-of-thought is not explainability. *Preprint, alphaXiv*, page v1.

Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2025. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Qiguang Chen, Libo Qin, Jiaqi Wang, Jinxuan Zhou, and Wanxiang Che. 2024. 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.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. 2024. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *arXiv preprint arXiv:2402.18312*.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36:70757–70798.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Wen-Chao Hu, Wang-Zhou Dai, Yuan Jiang, and Zhi-Hua Zhou. 2025. Efficient rectification of neuro-symbolic reasoning inconsistencies by abductive reflection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17333–17341.

Yeo Wei Jie, Ranjan Satapathy, Rick Goh, and Erik Cambria. 2024. How interpretable are reasoning explanations from prompting large language models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2148–2164.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. 2008. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36:36407–36433.

Xin Liu and Lu Wang. 2025. Answer convergence as a signal for early stopping in reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17907–17918.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Andrew Nam, Henry Conklin, Yukang Yang, Thomas Griffiths, Jonathan Cohen, and Sarah-Jane Leslie. 2025. Causal head gating: A framework for interpreting roles of attention heads in transformers. *arXiv preprint arXiv:2505.13737*.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. 2024. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8420–8436.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15012–15032.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622.

Yao Qiang, Deng Pan, Chengyin Li, Xin Li, Rhongho Jang, and Dongxiao Zhu. 2022. Attcat: Explaining transformers via attentive class activation tokens. *Advances in neural information processing systems*, 35:5052–5064.

Raanan Y Rohekar, Yaniv Gurwicz, and Shami Nisimov. 2023. Causal interpretation of self-attention in pre-trained transformers. *Advances in Neural Information Processing Systems*, 36:31450–31465.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020a. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020b. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems*, 37:66383–66409.

Zezhong Wang, Xingshan Zeng, Weiwen Liu, Yufei Wang, Liangyou Li, Yasheng Wang, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2025. Chain-of-probe: Examining the necessity and accuracy of cot step-by-step. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2586–2606.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025. Softcot: Soft chain-of-thought for efficient reasoning with llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23336–23351.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Zeping Yu and Sophia Ananiadou. 2024. Interpreting arithmetic mechanism in large language models through comparative neuron analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3293–3306.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*.

Tianshi Zheng, Yixiang Chen, Chengxi Li, Chunyang Li, Qing Zong, Haochen Shi, Baixuan Xu, Yangqiu Song, Ginny Y Wong, and Simon See. 2025. The curse of cot: On the limitations of chain-of-thought in in-context learning. *arXiv preprint arXiv:2504.05081*.

Zining Zhu and Frank Rudzicz. 2020. An information theoretic view on selecting linguistic probes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9251–9262.

A Additional Details

A.1 Prompt Templates

To ensure stage-aligned probing without introducing additional knowledge, we enforce structured outputs through carefully designed prompt templates. Tables 4 and 5 present the full templates used for LLaMA3-8B, R1-Qwen7B, and Qwen2.5-Math-7B, respectively.

A.2 Dataset Examples

Tables 6 and 7 provide illustrative dataset examples across four models and two domains. Each example follows the structured output format defined in Appendix A.1. Problems were generated by the moonshot-v1-8k model (AI, 2025), and responses by the evaluated models.

A.3 Training Configurations

We trained 36 independent MLP probes across different layer configurations, models, and subject domains. Each probe is a single-hidden-layer classifier trained to predict the theorem identity from the hidden representations extracted at a specific layer segment. The layer configurations include: (1) All Layers (2) All Except Last 2 (3) Last 18 Layers (4) Last 15 Layers (5) Last 10 Layers (6) Last 5 Layers. For each of the three instruction-tuned models and two subject domains, all six probes were trained, yielding $3 \times 2 \times 6 = 36$ total training runs.

Each probe receives as input the hidden representation of the final token corresponding to the theorem-prediction step. For a given layer configuration, all selected layer representations are flattened into training samples, and labels are replicated accordingly. To control class balance across theorem categories, we fix the number of prefix samples for each theorem to three. Only three prefixes are randomly selected from all available prefix candidates of each theorem to prevent long or complex formulas from dominating the dataset. This sampling strategy ensures a balanced class distribution, stabilizes training, and improves evaluation comparability.

The dataset is split by theorem category with a fixed ratio of 6:1:3 for training, validation, and test sets, ensuring that no duplicate problems occur across splits. All probes share the same optimization setup. We employ the AdamW optimizer with a learning rate of 1×10^{-3} and weight decay of 1×10^{-4}. Each model is trained for up to 20 epochs with early stopping determined by validation loss, using a patience of 3. A cosine annealing scheduler is applied across 20 epochs, and the batch size is 64. Each MLP probe consists of one hidden layer of 1024 dimensions with a GELU activation, Layer Normalization at the input, and a Dropout rate of 0.2. The key training configurations are summarized in Table 2.

Unless otherwise specified, we fix the random

Model	Mathematics						Physics					
	All	All (-2)	Last 18	Last 15	Last 10	Last 5	All	All (-2)	Last 18	Last 15	Last 10	Last 5
LLaMA3 -8B	0.779 ±0.003	0.780 ±0.009	0.879 ±0.005	0.889 ±0.005	0.882 ±0.007	0.877 ±0.020	0.847 ±0.005	0.849 ±0.012	0.904 ±0.007	0.912 ±0.013	0.911 ±0.010	0.924 ±0.008
R1-Qwen7B	0.786 ±0.014	0.795 ±0.016	0.855 ±0.006	0.867 ±0.002	0.874 ±0.012	0.879 ±0.012	0.869 ±0.007	0.862 ±0.018	0.895 ±0.004	0.905 ±0.003	0.913 ±0.005	0.917 ±0.010
Qwen2.5-Math-7B	0.751 ±0.015	0.743 ±0.016	0.823 ±0.017	0.822 ±0.012	0.829 ±0.014	0.811 ±0.019	0.766 ±0.015	0.755 ±0.007	0.828 ±0.012	0.830 ±0.014	0.840 ±0.002	0.842 ±0.006

Table 1: Test accuracy of theorem-classification probes across different layer configurations and domains. Values are reported as mean and standard deviation over 5 random seeds.

seed to 42 for reproducibility. All probes are trained using cross-entropy loss, and we report classification accuracy as the evaluation metric. Table 1 reports mean ± standard deviation of test accuracy over five random seeds $\{42, 123, 456, 789, 1024\}$.

Hyperparameter	Value
Input Dim.	4096 (LLaMA3) 3584 (R1/Qwen2.5)
Hidden Dim.	1024
Output Dim.	50
Optimizer	AdamW
Learning Rate	1×10^{-3}
Epochs / Patience	20 / 3

Table 2: Training configurations for theorem-classification probes.

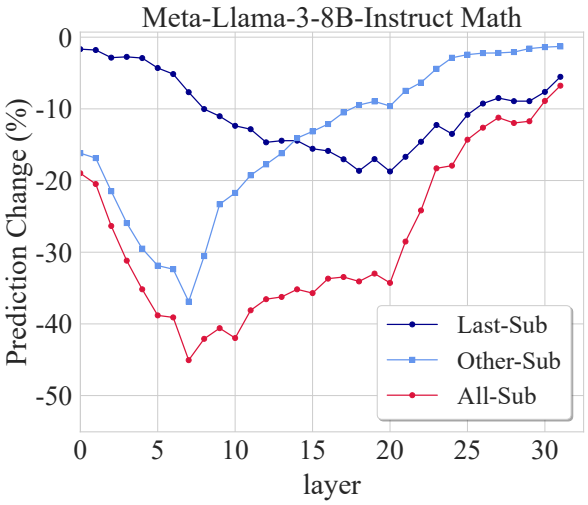

Figure 13: Relative change in prediction probability when blocking attention to theorem-related token positions during theorem generation across layers.

B Free-form Prompting Experiments

Motivation. Our main experiments adopt a structured output to ensure deterministic stage boundaries and stable token alignment for attention interventions and probing. To verify that the observed multi-stage dynamics are not artifacts of this formatting constraint, we additionally evaluate a free-form prompting setting without labeled fields.

Prompt and free-form data. Given that free-form traces lack labeled fields, we manually annotated 100 data points. Table 3 presents the prompt used for free-form cot. Table 8 presents the dataset examples under free-form cot.

Free-form CoT Prompt Template
You are a mathematical reasoning assistant. Please solve the following problem step by step and finally output [end] to conclude. Now solve the following problem using the above format: Problem: {question}

Table 3: Prompt template used for free-form CoT.

B.1 Theorem Generation under Free-form CoT

We analyze information flow when the model generates the first token of a theorem content. Consistent with the structured setting, we observe a two-stage propagation pattern: in the early layers, blocking attention from *other-tokens* leads to a larger performance drop of around 30%; in the mid-to-late layers, the *final-token* becomes dominant for theorem usage, and blocking its attention causes a prediction probability decrease of about 20%. Figure 13 presents the results.

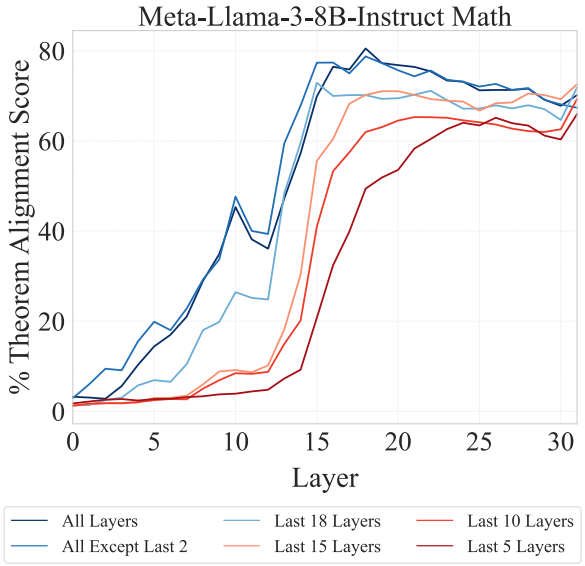

Figure 14: Performance of probes trained on different layer configurations.

B.2 Theorem Probing under Free-form CoT

For theorem probing, we extract the hidden states when the model generates the first token of a theorem content and apply the theorem-classification probes trained under the fixed-format setting. Figure 14 presents the results. The layer-wise trend remains highly consistent with the fixed-format case: probe accuracy rises sharply in the middle layers, from about 0.1 to 0.8, indicating that even under unconstrained traces, theorem semantics remain most linearly decodable in the mid-to-late layers.

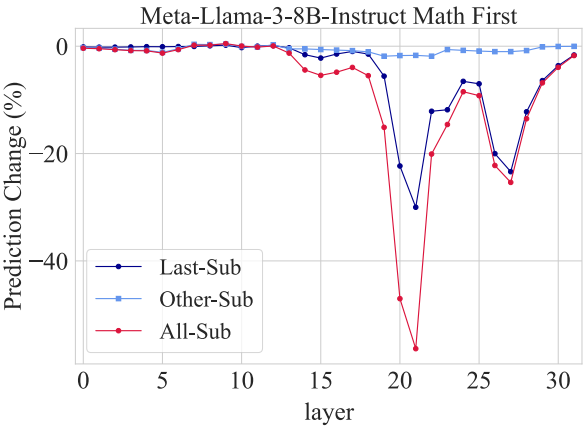

Figure 15: Relative change in prediction probability when blocking attention to parameter-token positions during instantiation of the first parameter across layers.

B.3 Parameter Substitution under Free-form CoT

We analyze parameter instantiation on the first generated numeric parameter token. When we block attention to parameter tokens, the resulting pattern of changes in prediction probability is consistent with the structured setting: in the mid-to-late layers, Last-Sub blocking causes a substantial probability drop of roughly 20%, and All-Sub leads to an even larger degradation of around 50%, while Other-Sub blocking has almost no effect. Figure 15 presents the results.

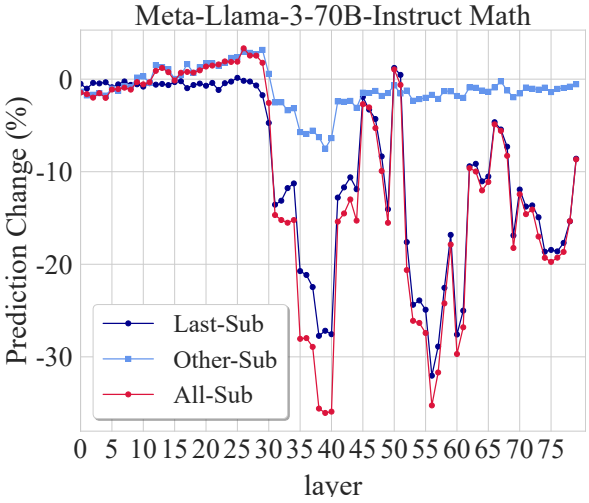

Figure 16: Relative change in prediction probability.

C Large-Model Experiments on LLaMA3-70B

To test whether our findings scale to larger backbones, we extend our analysis to Meta-Llama-3-70B-Instruct (80 layers). Since attention blocking is applied over a contiguous layer window, the effective coverage of an intervention depends on model depth. We therefore scale the window to $k = 30$ for the 80-layer model to keep the intervention span comparable to the main experiments in relative depth, while preserving the same blocking definitions and evaluation metric.

C.1 Keyword Extraction

As shown in Figure 16, the Last-Sub intervention causes a pronounced probability drop of up to 30% in the mid-to-late layers, whereas the Other-Sub intervention leads to almost no observable change.

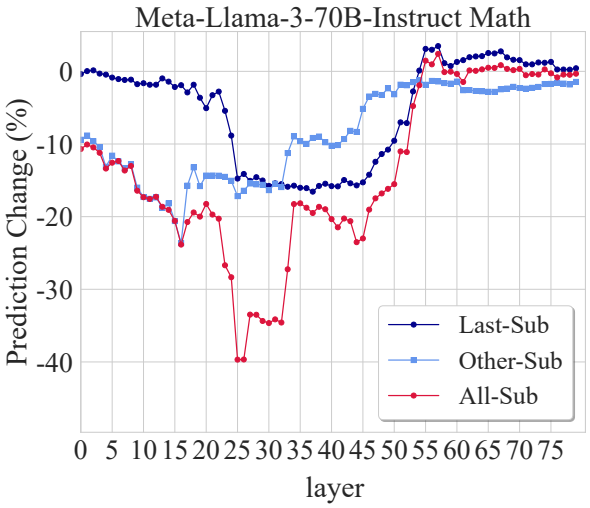

Figure 17: Relative change in prediction probability.

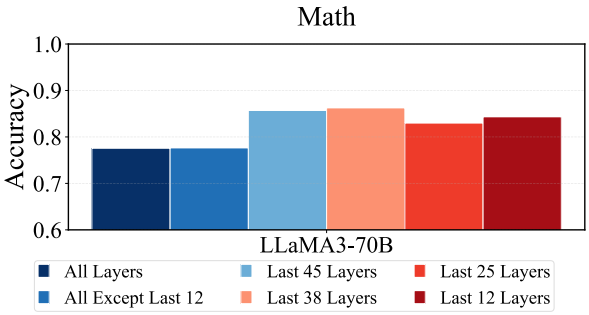

Figure 18: Accuracy of probes trained on hidden states from different layer ranges for detecting theorem tokens.

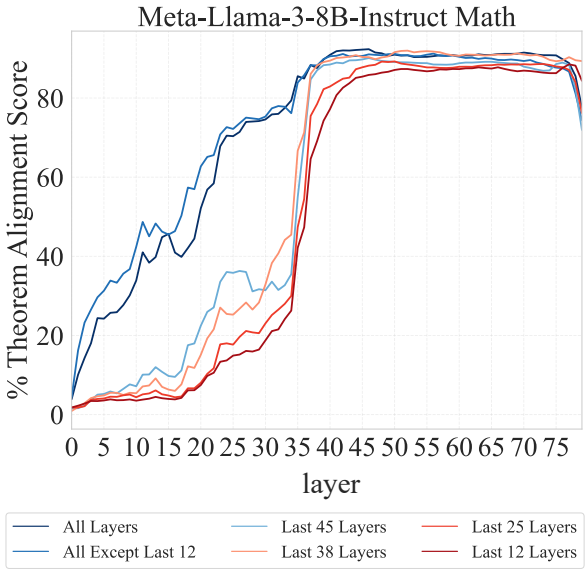

Figure 19: Probe performance under different layer configurations on LLaMA3-70B.

C.2 Theorem Generation: Keyword Extraction for Theorem Generation

As shown in Figure 17, theorem generation exhibits the same two-stage information-flow pattern as in smaller models. In the early layers, the effect of blocking the Other-Sub positions dominates, leading to an approximate 20% decrease in prediction probability. As the depth increases, the impact of the Last-Sub blocking becomes increasingly pronounced. This change in prediction probability is consistent with that observed in smaller models.

C.3 Encoding Theorem Information in LLMs

We train the same single-hidden-layer MLP probe on LLaMA3-70B hidden states and evaluate different layer groups. As shown in Figure 18, probes trained on mid-to-late layer ranges achieve the best accuracy for detecting theorem tokens, and the probe training results are consistent with those observed on smaller models. Figure 19 reports the corresponding probe performance under different layer configurations, and the overall trends closely match those of smaller models.

C.4 Parameter Extraction and Instantiation

C.4.1 Parameter Extraction

Figure 20 presents the results for the first parameter. Last-Sub blocking causes an approximately 40% decrease in prediction probability in the middle layers, whereas All-Sub blocking leads to a drop exceeding 60%. Meanwhile, Other-Sub blocking produces almost no observable impact.

Figures 21 and 22 show the results for the subsequent parameters. For these later parameters, the effect of Last-Sub blocking gradually diminishes, while the influence of Other-Sub blocking becomes significantly stronger. This behavior is consistent with the parameter-extraction mechanism observed in smaller models.

C.4.2 Parameter–Theorem Matching

As shown in Figure 23, blocking attention connections between parameter tokens and theorem tokens at different layers leads to almost no change in prediction probability. This indicates that, for LLaMA3-70B, parameter extraction is not primarily driven by direct information flow from theorem tokens, but is instead achieved through broader contextual representations, aligning with the observations in smaller models.

D Sensitivity Analyses

Setup. We analyze the sensitivity of our conclusions to two design choices in our probing and

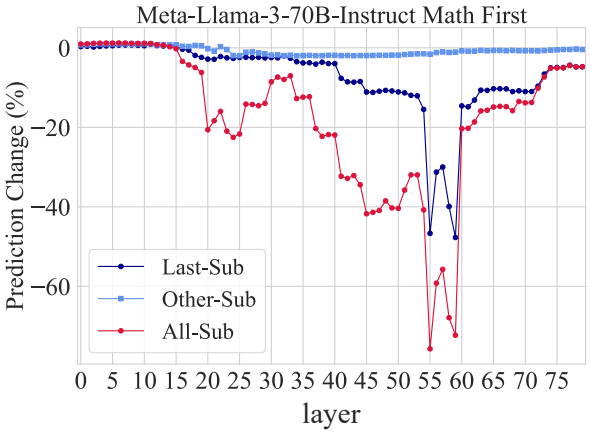

Figure 20: Relative change in prediction probability of the first parameter across layers.

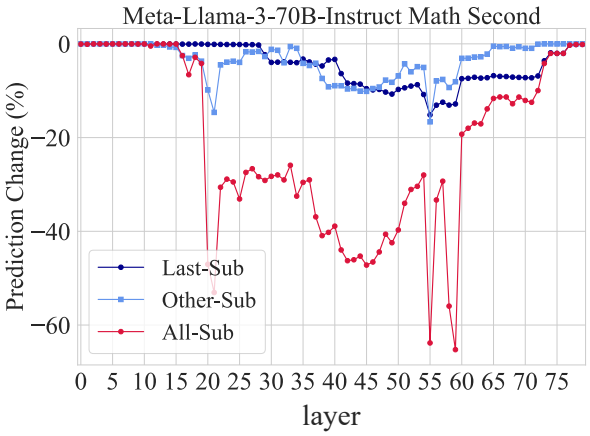

Figure 21: Relative change in prediction probability of the second parameter across layers.

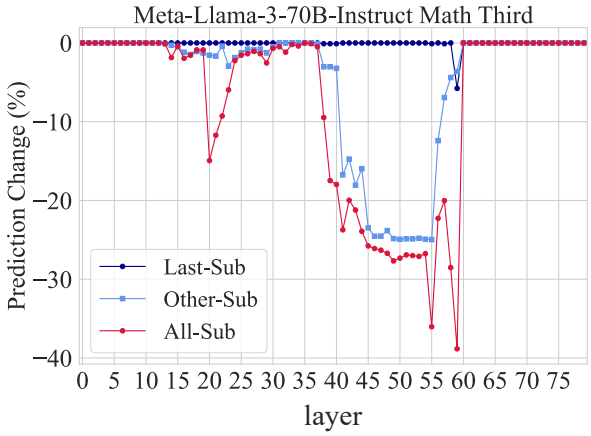

Figure 22: Relative change in prediction probability of the third parameter across layers.

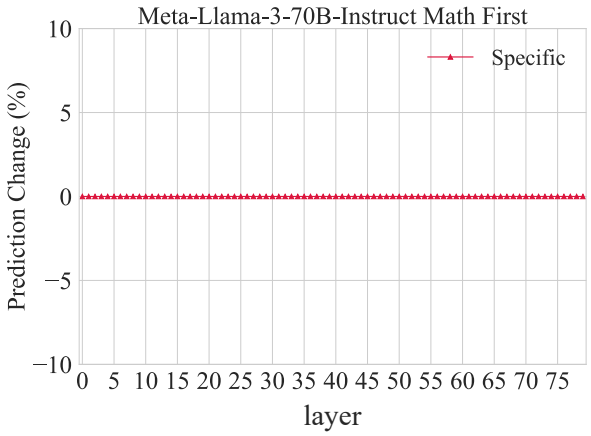

Figure 23: Relative change in prediction probability.

intervention pipeline. (i) We vary the attention knockout blocking window size k, using $k{=}10$ as the default setting and additionally evaluating $k{=}5$ and $k{=}20$. (ii) We vary the MLP probe depth. The main paper uses a one-hidden-layer probe, and we further evaluate probes with two and three hidden layers. Unless otherwise specified, all other settings follow the main experiments.

D.1 Sensitivity to the blocking window size k

Keyword extraction. We rerun Attention Knockout with $k{=}5$ and $k{=}20$. The qualitative pattern remains stable: mid-to-late layers primarily rely on the *final token* to extract keyword information. As shown in Figures 24 and 25, Last-Sub continues to induce a substantial probability drop with $k{=}5$ (about -48%), and the magnitude further increases with $k{=}20$ (up to about -80%), while the overall layer-wise trend is preserved.

Theorem generation. We rerun Attention Knockout with $k{=}5$ and $k{=}20$ and summarize the results in Figures 26 and 27. Across both settings, we observe the same two-stage propagation pattern. In the early layers, the trend under All-Sub closely follows Other-Sub, indicating an early relay effect where theorem-token information is primarily captured by non-final tokens. In the mid-to-late layers, the trend shifts to align with Last-Sub, suggesting that theorem-token information is re-extracted and then read out more directly by the final token.

Parameter extraction. We further validate robustness on parameter extraction. Across k, the mid-to-late-layer joint extraction mechanism consistently appears. As illustrated in Figures 28 and 29, with $k{=}5$, Last-Sub yields a smaller probability drop for the first parameter (about -8%), while with $k{=}20$, Last-Sub declines become much larger (about -78% to -87%). Despite the magnitude change, the sensitive layer region and the

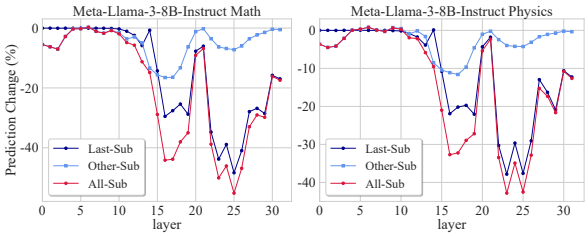

Figure 24: Keyword extraction with k=5 showing relative probability change across layers.

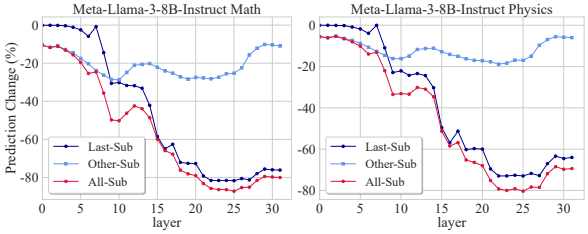

Figure 25: Keyword extraction with k=20 showing relative probability change across layers.

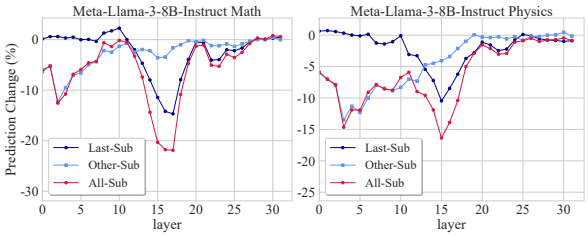

Figure 26: Theorem generation with k=5 showing relative change in prediction probability across layers.

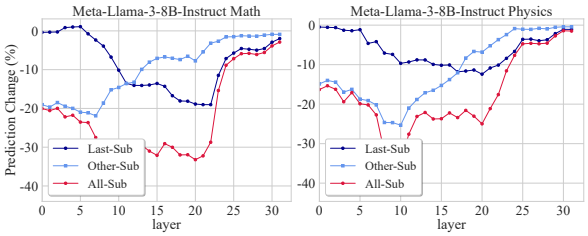

Figure 27: Theorem generation with k=20 showing relative change in prediction probability across layers.

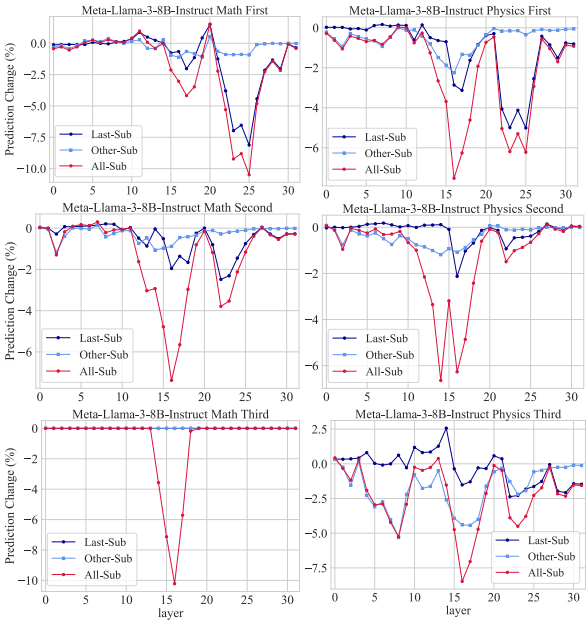

Figure 28: Parameter extraction with k=5 showing relative change in prediction probability across layers.

concentration of critical information remain consistent.

D.2 Sensitivity to the MLP probe depth

Probe depth. In the main paper, we use an MLP probe with **one** hidden layer. Here we additionally compare deeper probes with **two** and **three** hidden layers. On hidden states from the last 15 layers of the mathematics dataset, the one-hidden-layer probe already achieves 82.7% accuracy, and deeper probes bring only marginal improvements (about 83.0% and 83.2%). Figures 30–31 and Figures 32–33 further show that increasing probe depth does not alter the qualitative encoding patterns across layers: theorem-related information remains linearly separable, and the probe identifies highly consistent layer ranges as most informative.

D.3 Overall robustness.

Taken together, these ablations suggest that our conclusions about the information-flow mechanisms during CoT reasoning do not hinge on a particular choice of blocking window size k or MLP probe depth. Varying k primarily changes the overall size of the probability shifts, while preserving the same layer-wise roles across tasks, and deeper probes yield only negligible gains while maintaining consistent theorem-related encoding trends.

Model	Prompt Template
LLaMA3-8B	You are a reasoning assistant. Follow **exactly** the five-step template below to solve the problem using a known physical or mathematical formula. [Keywords]: <comma-separated list of key concepts mentioned in the problem> [Theorem]: <formula name> = <symbolic formula using variables only — absolutely no numbers from the problem> [Computation]: <numeric expression showing substituted values only> = <result> [Answer]: <numeric result only — do NOT include any units> Strict rules: - The formula name and its variable-based form must appear **only** in [Theorem]. - The [Keywords] must exactly match the **original words** from the problem. Keep only the minimal set of words sufficient to determine which formula to use. - [Theorem] must include a symbolic expression using variables only. Do NOT use any numbers in this step. - [Computation] must contain **only numeric values**, and no variable names or formula names. - Do NOT skip any step or add extra explanation. Do NOT include any units. - **End your answer with [end].** Problem: {question}
R1-Qwen7B	You are a mathematical reasoning assistant. You must strictly follow the 5-step template below. Do **not** add any explanations, LaTeX formatting, or extra sentences. Only output the following 5 fields, exactly in this format: [Keywords]: Extract only key concepts from the problem, exactly as they appear. Separate by commas. [Theorem]: Write the formula name and its symbolic form using variables only. Do NOT include any numbers from the problem. Do not explain the formula. The formula must contain an equals sign '='. [Computation]: Replace variables with numbers and compute. Only include numeric expressions and results, no variables or formula names. [Answer]: Final numeric result only. Do NOT include any units. [end]: Indicate the end of the answer. Strict rules: - Theorem must be pure symbolic form. Example: GOOD: F = a $\times$ m BAD: resultant force is given by F = a $\times$ m, where a is acceleration and m is mass. - Output only these 5 fields, and nothing else. - Use one line per field. Do NOT include LaTeX, explanations, or additional sentences. - The field labels must be exactly as shown above. - No extra formatting, no markdown, no lists, no multi-turn conversation. Now solve the following problem using the above format: Problem: {question}

Table 4: Prompt templates for LLaMA3-8B and R1-Qwen7B.

Model	Prompt Template
Qwen2.5-Math-7B	You are a mathematical reasoning assistant. You must strictly follow the 5-step template below. Do **not** add any explanations, LaTeX formatting, or extra sentences. Only output the following 5 fields, exactly in this format: [Keywords]: Extract only key concepts from the problem, exactly as they appear. Separate by commas. [Theorem]: Write only the symbolic form of the formula. Do NOT include any numbers from the problem. Do not explain the formula or output the formula name. The formula must contain an equals sign '='. [Computation]: Replace variables with numbers and compute. Only include numeric expressions and results, no variables or formula names. [Answer]: Final numeric result only. Do NOT include any units. [end]: Indicate the end of the answer. Strict rules: - Theorem must be pure symbolic form. Example: GOOD: F = a * m BAD: resultant force is given by F = a * m, where a is acceleration and m is mass. - Output only these 5 fields, and nothing else. - Use one line per field. Do NOT include explanations or extra sentences. - Write all formulas and equations as plain text. Do not use LaTeX format, and you **must not** use LaTeX delimiters like or $. - The field labels must be exactly as shown above. - No extra formatting, no markdown, no lists, no multi-turn conversation. Now solve the following problem using the above format: Problem: {question}

Table 5: Prompt template for Qwen2.5-Math-7B.

Model	Mathematics Dataset	Physics Dataset
LLaMA3-8B	**Example 1**: Problem: Xiao Li is decorating his room and wants to hang a square decorative painting on the wall. If each side of the decorative painting is 20 centimeters long, what would be the area of this decorative painting? [Keywords]: square, decorative painting, area [Theorem]: A = s ^ 2 [Computation]: 20 ^ 2 = 400 [Answer]: 400	**Example 1**: Problem: Xiao Gang walked from home to school, covering a total distance of 300 meters in 6 minutes. What is Xiao Gang's average speed? [Keywords]: distance, speed, time [Theorem]: v = d / t [Computation]: 300 / 6 = 50 [Answer]: 50
	Example 2: Problem: Xiao Gang is making a square picture frame and has chosen a wooden stick that is 80 centimeters long as the material. He wants to know if this stick can form the frame and is asking what the side length of the square picture frame would be. [Keywords]: square picture frame, side length [Theorem]: P = 4 ÃŮ s [Computation]: 4 ÃŮ s = 80 [Answer]: 20	**Example 2**: Problem: Xiao Gang started from his home to go to the park, walking at a speed of 5 kilometers per hour and took 1 hour to reach the park. What is the distance from Xiao Gang's home to the park? [Keywords]: distance, speed, time [Theorem]: d = s ÃŮ t [Computation]: 5 ÃŮ 1 = 5 [Answer]: 5
R1-Qwen7B	**Example 1**: Problem: Xiao Li is decorating his room and wants to hang a square decorative painting on the wall. If each side of the decorative painting is 20 centimeters long, what would be the area of this decorative painting? [Keywords]: square, decorative painting, area, side length, centimeters [Theorem]: A = s ^ 2 [Computation]: 20 ^ 2 = 400 [Answer]: 400	**Example 1**: Problem: Xiao Gang walked from home to school, covering a total distance of 300 meters in 6 minutes. What is Xiao Gang's average speed? [Keywords]: Xiao Gang, home, school, total distance, 300 meters, 6 minutes, average speed [Theorem]: Average speed = total distance / total time [Computation]: 300 / 6 = 50 [Answer]: 50
	Example 2: Problem: Xiao Gang is making a square picture frame and has chosen a wooden stick that is 80 centimeters long as the material. He wants to know if this stick can form the frame and is asking what the side length of the square picture frame would be. [Keywords]: Xiao Gang, square picture frame, wooden stick, length, side length [Theorem]: P = 4 ÃŮ s [Computation]: 80 = 4 ÃŮ s [Answer]: 20	**Example 2**: Problem: Xiao Gang started from his home to go to the park, walking at a speed of 5 kilometers per hour and took 1 hour to reach the park. What is the distance from Xiao Gang's home to the park? [Keywords]: Xiao Gang, home, park, speed, time [Theorem]: d = s ÃŮ t [Computation]: 5 ÃŮ 1 = 5 [Answer]: 5

Table 6: Illustrative dataset examples for LLaMA3-8B and R1-Qwen7B across Mathematics and Physics.

Model	Mathematics Dataset	Physics Dataset
Qwen2.5-Math-7B	**Example 1**: Problem: Xiao Li is decorating his room and wants to hang a square decorative painting on the wall. If each side of the decorative painting is 20 centimeters long, what would be the area of this decorative painting? [Keywords]: square, side length, area [Theorem]: A = s ^ 2 [Computation]: 20 ^ 2 = 400 [Answer]: 400	**Example 1**: Problem: Xiao Gang walked from home to school, covering a total distance of 300 meters in 6 minutes. What is Xiao Gang's average speed? [Keywords]: Xiao Gang, distance, time, average speed [Theorem]: v = d / t [Computation]: 300 / 6 = 50 [Answer]: 50
	Example 2: Problem: Xiao Gang is making a square picture frame and has chosen a wooden stick that is 80 centimeters long as the material. He wants to know if this stick can form the frame and is asking what the side length of the square picture frame would be. [Keywords]: square picture frame, wooden stick, side length [Theorem]: P = 4 ÃŮ s [Computation]: 4 ÃŮ s = 80 [Answer]: 20	**Example 2**: Problem: Xiao Gang started from his home to go to the park, walking at a speed of 5 kilometers per hour and took 1 hour to reach the park. What is the distance from Xiao Gang's home to the park? [Keywords]: Xiao Gang, speed, time, distance [Theorem]: d = s ÃŮ t [Computation]: 5 ÃŮ 1 = 5 [Answer]: 5

Table 7: Illustrative dataset examples for Qwen2.5-Math-7B across Mathematics and Physics.

Problem	Free-form CoT Output
A square clock has a side length of 15 centimeters. What is the area of the square?	Let's solve the problem step by step. Step 1: Identify the given information The clock is a square with a side length of 15 centimeters. Step 2: Find the area of the square The area of a square is calculated by squaring the side length. Area = side^2 = 15^2 = 15 $\times$ 15 = 225 Step 3: Express the answer The area of the square clock is 225 square centimeters.
Xiao Hua is making a rectangular picture frame, with the length being 20 centimeters and the width being 15 centimeters. What is the area of this picture frame?	To solve this problem, I'll follow the steps: Step 1: Identify the given information Length (L) = 20 centimeters Width (W) = 15 centimeters Step 2: Calculate the area of the picture frame Area = L $\times$ W = 20 $\times$ 15 = 300 square centimeters Step 3: Output the solution The area of the picture frame is 300 square centimeters.
Xiao Hua is making a globe model, and he chose a sphere with a radius of 5 centimeters. What is the surface area of this globe model?	To solve this problem, we can use the formula for the surface area of a sphere, which is: Surface Area = 4 * π * r^2 where r is the radius of the sphere. Given that the radius of the sphere is 5 centimeters, we can plug this value into the formula: Surface Area = 4 * π * (5)^2 = 4 * π * 25 = 4 * 3.14 * 25 = 314 So, the surface area of the globe model is 314 square centimeters.

Table 8: Dataset examples under free-form CoT.

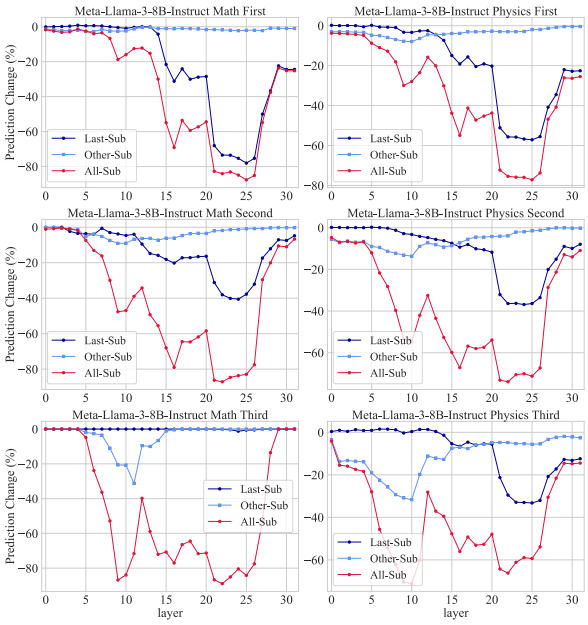

Figure 29: Parameter extraction with k=20 showing relative change in prediction probability across layers.

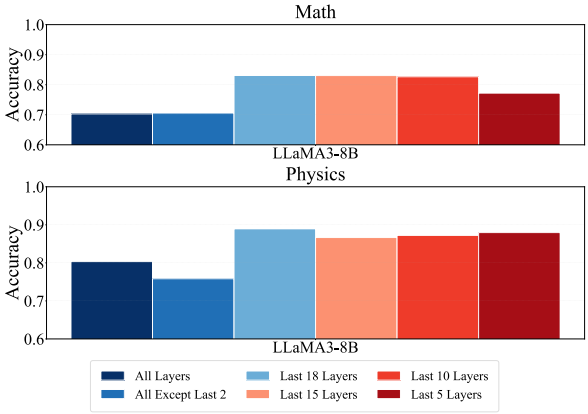

Figure 30: Probing accuracy with a two-hidden-layer MLP probe.

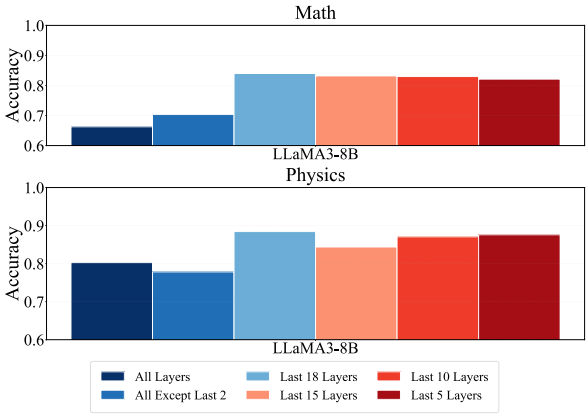

Figure 31: Probing accuracy with a three-hidden-layer MLP probe.

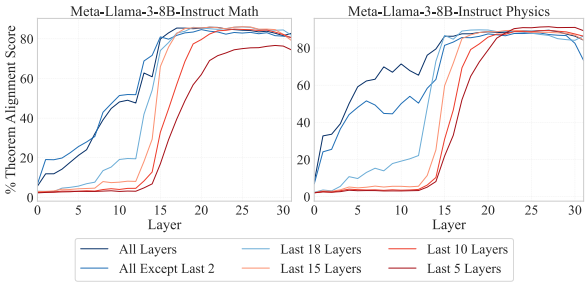

Figure 32: Performance of probes trained on different layer configurations.

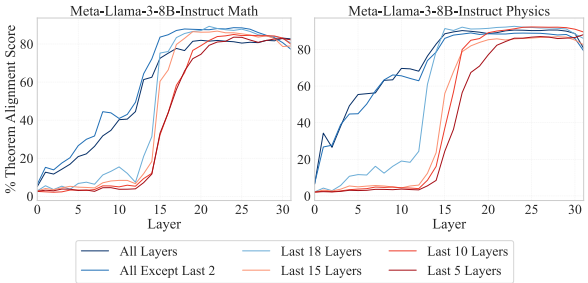

Figure 33: Performance of probes trained on different layer configurations.