When Can Large Reasoning Models Save Thinking? Mechanistic Analysis of Behavioral Divergence in Reasoning

Anonymous ACL submission

Abstract

Large reasoning models (LRMs) have significantly advanced performance on complex tasks, yet their tendency to overthink introduces inefficiencies. This study investigates the internal mechanisms of reinforcement learning (RL)-trained LRMs when prompted to save thinking, revealing three distinct thinking modes: no thinking (NT), explicit thinking (ET), and implicit thinking (IT). Through comprehensive analysis of confidence in thinking termination, attention from thinking to generation, and attentional focus on input sections, we uncover key factors influencing the reasoning behaviors. We further find that NT reduces output length at the cost of accuracy, while ET and IT maintain accuracy with reduced response length. Our findings expose fundamental inconsistencies in RL-optimized LRMs, necessitating adaptive improvements for reliable efficiency.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance across a wide range of domains (OpenAI, 2023; Yang et al., 2024; DeepSeek-AI et al., 2024). When integrated with retrieval-augmented generation (Lewis et al., 2020; Zhu et al., 2025), supervised fine-tuning (SFT) (Parthasarathy et al., 2024; Luo et al., 2024), or other techniques (Fang et al., 2025), LLMs achieve improved capabilities in specialized domains beyond their original pretraining scope. However, LLMs may still struggle with complex reasoning tasks (Anand et al., 2024). The breakthrough of large reasoning models (LRMs) like OpenAI-o1 (OpenAI, 2025) highlights the benefits of scaling test time, enabling extended reasoning and the generation of comprehensive thoughts. This approach can significantly improve the accuracy on complex reasoning tasks. As a result, an increasing number of LRMs, such as QwQ (Qwen Team, 2025) and DeepSeek-R1 (DeepSeek-AI et al., 2025), have been developed. Broadly, these LRMs can be categorized based on their training methodology: one category is developed through SFT on chain-of-thought (CoT) data (Muennighoff et al., 2025), while the other is trained directly using reinforcement learning (RL). These models demonstrate superior performance compared to LLMs across various reasoning tasks (Chen et al., 2025).

Most LRMs, such as QwQ-32B, generate tags like `<think>` and `</think>` to encapsulate the thinking process, followed by the final answer. Despite the significant improvement in reasoning capabilities, LRMs usually suffer from the overthinking problem. They tend to generate excessively long reasoning chains, even for simple tasks (Chen et al., 2024), which leads to computational inefficiency and sometimes worse accuracy (Sui et al., 2025). To address this issue, Ma et al. (2025) propose prompting LRMs to bypass thinking by pre-filling the segment between `<think>` and `</think>`. The prompt template is given below.

> **User**: Your final answer should follow immediately after the phrase "The final answer is".
> **Question**: [*Question*].
> **Assistant**:
> `<think>`
> Okay, I think I have finished thinking.
> `</think>`

Intriguingly, our empirical evidence reveals that when this prompt is applied to the native LRMs trained using RL (e.g., QwQ-32B), models sometimes skip thinking as expected while at other times re-engage in thinking. This behavioral inconsistency raises a critical question: *what internal mechanisms cause RL-trained LRMs to exhibit such different responses when guided to "save thinking"*. To investigate this problem, we categorize the behaviors of LRMs under save-thinking instructions into three modes based on our empirical findings:

no thinking (NT), explicit thinking (ET), and implicit thinking (IT), as illustrated in Figure 1:

- **No thinking**: The LRM bypasses further thinking and directly generates the answer.
- **Explicit thinking**: The LRM re-engages in a thinking process before providing the answer. It appends an additional </think> tag upon completing its renewed thinking phase.
- **Implicit thinking**: The LRM also re-engages in additional thinking but does not output the </think> tag to mark the end of thinking.

To explore different thinking modes and analyze their internal distinctions, we examine LRMs from three perspectives: confidence in thinking termination, attention from thinking to generation, and attentional focus on input sections. We also analyze how modes affect performance and output length. Our contributions are summarized as follows:

- We identify three behavioral modes—NT, ET, and IT—that emerge when the native LRM is instructed to save thinking. The divergences suggest LRMs struggle with thinking control, emphasizing the need for improved instruction adherence and behavioral consistency.
- We analyze three internal states of LRMs in different thinking modes. In NT mode, the LRM prioritizes the pre-filled thinking content and has high confidence in thinking termination. In ET and IT modes, the LRM retains focus on the task context, enabling their re-engagement in reasoning. Attention activation patterns across the three modes begin to diverge distinctly from the early layers. Our analysis connects observed behavior and internal decision-making, guiding future improvements in reasoning reliability of LRMs.
- We compare the output length and accuracy in different modes against a baseline where the LRM follows standard thinking. NT mode significantly shortens output length but reduces accuracy. ET preserves accuracy while moderately reducing response length. This evaluation connects thinking modes and answer quality, offering valuable insights for research on efficient and effective reasoning.

Our manual annotations of the three thinking modes in the GSM8K and MATH500 datasets will be publicly available to support future research.

(a) No thinking

Question: James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?

LRM output: The final answer is 540.

(b) Explicit thinking

Question: … The choir plans to sing “Deck the Halls” twice and “Jingle Bells” once. If “Deck the Halls” is 150 s long and “Jingle Bells” is 240 s long, how long will they be caroling?

LRM output: *The total time for “Deck the Halls” is 2 times 150 seconds. “Jingle Bells” is sung once, so that’s 240 seconds. Adding them together gives 300 + 240 = 540 seconds.* </think> The final answer is 540 seconds.

(c) Implicit thinking

Question: Ryan plants 2 flowers a day in his garden. After 15 days, how many flowers does he have if 5 did not grow?

LRM output: *... I need to multiply the number of flowers per day by the number of days. So, 2 flowers/day multiplied by 15 days equals 30 flowers. But wait, the problem says that 5 did not grow. ... So I should subtract those 5 from the total planted. ...* The final answer is 25.

Figure 1: Examples of the three modes of QwQ-32B under the save-thinking instructions. The final answers are underlined. The thought is marked in italics. The token </think> marks the end of the thought.

2 Related Work

2.1 Large Reasoning Models

Large language models (LLMs) have demonstrated stronger capabilities with larger scale (Zhao et al., 2023), yet they still struggle with complex reasoning tasks like mathematics and code generation (Anand et al., 2024; Xu et al., 2025a). Recent work has found that instead of continuously scaling model size and training data, scaling the model’s thinking time and having it generate chain-of-thought reasoning—similar to human thinking—can significantly improve the accuracy of complex tasks (Snell et al., 2024). Consequently, LRMs such as OpenAI-o1 (OpenAI, 2025), DeepSeek-R1 (DeepSeek-AI et al., 2025), and QwQ (Qwen Team, 2025) have emerged, specifically designed to produce structured reasoning processes before the final answer. These LRMs improve performance on challenging reasoning benchmarks by generating longer thinking processes that involve considering multiple potential solutions and backtracking from errors (Chen et al., 2025).

2.2 Efficient Reasoning

LRMs often exhibit the phenomenon of “overthinking”. This manifests as the LRMs generating extensive thinking processes even for relatively simple problems, such as “calculating 2+3”, leading to significant computational resource waste (Chen et al., 2024). Furthermore, some studies (Sui et al., 2025; Yang et al., 2025) have indicated that in certain domains, these lengthy thinking processes can introduce noise, which may paradoxically lead to a decrease in the model's overall performance.

To address this issue, several methods have been explored, including approaches like RL with Length Reward Design (Meng et al., 2024; Aggarwal and Welleck, 2025; Qu et al., 2025) and SFT with different Variable-Length CoT Data (Muennighoff et al., 2025; Han et al., 2024). Among these, utilizing prompts to explicitly limit the thinking length is considered a highly efficient method (Xu et al., 2025b; Li et al., 2025). Previous work (Ma et al., 2025) demonstrated that by pre-filling the thinking section of LRMs with a completion instruction, models trained via SFT on CoT data, such as DeepSeek-R1-Distill-Qwen-32B, could effectively skip the thinking process without a significant drop in accuracy. However, when the same prompting strategy is applied to LRMs trained using RL, such as QwQ-32B, models sometimes skip the thinking process while at other times re-engage in thinking, a phenomenon also mentioned in prior work (Liu et al., 2025). Our work aims to analyze the internal states of the model under different thinking modes to investigate reasons behind this varied behavior in RL-trained LRMs.

3 Mechanistic Analysis of Thinking Modes

3.1 Experiment Setup and Data Annotations

LRM selection. Our primary experiments use the open-source, RL-trained native LRM QwQ-32B (Qwen Team, 2025). We select this model due to its strong performance and widespread adoption. More importantly, the native LRM usually exhibits behavioral divergence when prompted to economize thought (see the prompt template in Section 1). QwQ-32B frequently engages in reasoning even when explicitly instructed to skip directly to the final answer. This behavior contrasts with LLMs such as ChatGPT and SFT-based LRMs such as DeepSeek-R1-Distill-Qwen-32B, which consistently follow save-thinking instructions, as shown in prior work (Liu et al., 2025; Ma et al., 2025). QwQ-32B thus serves as a valuable case study for investigating the internal dynamics behind such non-compliant reasoning in native LRMs.

Datasets and Annotations. Our experiments and analysis are conducted on two mathematical reasoning datasets: the test set of GSM8K (Cobbe et al., 2021), comprising grade school math word problems, and MATH500 (Hendrycks et al., 2021), a more challenging set from high school competitions. Both datasets require multi-step reasoning from LRMs. Specifically, given a reasoning question, we employ the prompt detailed in Section 1 to direct QwQ-32B to provide an immediate response. We manually check the outputs to identify the corresponding thinking modes. The statistics of questions manually labeled with thinking modes (NT, ET, and IT) are presented in Table1.

Datasets	NT	ET	IT	Total
GSM8K	948	296	75	1319
MATH500	118	379	3	500

Table 1: Statistics of questions in each thinking mode.

We observe that for GSM8K, QwQ-32B directly answer about 71.9% of questions without engaging in thinking. However, for the remaining questions, QwQ-32B still undertakes additional thinking, despite the prompt already pre-filling the reasoning segment. For MATH500, only 23.6% of questions are resolved without additional reasoning. The remaining questions require further thought, with explicit thinking being particularly prevalent.

Please note that, for all the questions, we use the same prompt to instruct QwQ-32B to save thinking. However, QwQ-32B still exhibits different thinking modes on different questions. This behavioral divergence motivates our analysis of its internal states to explore the underlying reasons. More details are given in Appendix A.

3.2 Confidence in Thinking Termination

Motivation. Our prompt includes a pre-filled thinking segment: "<think> Okay, I think I have finished thinking. </think>". The LRM is expected to generate the answer directly. Since the </think> token indicates the end of the thinking process, we posit that its prediction confidence reflects how certain the LRM is in resolving the question without further reasoning. Therefore, we examine this internal prediction confidence to gain insights into the distinctions among different thinking modes. Specifically, we conduct a forward pass on the pre-filled thinking segment to obtain the softmax probabilities when the LRM is predicting the </think> tag, immediately following the pre-filled "<think> Okay, I think I have finished thinking." phrase.

Modes	GSM8K			MATH500			Average		
	Top1	Entropy	DF	Top1	Entropy	DF	Top1	Entropy	DF
NT	78.64	1.05	72.61	73.21	1.15	65.53	78.04	1.06	71.83
IT	70.72	1.29	62.47	74.64	0.94	66.33	70.87	1.28	62.62
ET	73.46	1.22	66.32	67.71	1.28	57.72	70.12	1.25	61.33

Table 2: Prediction confidence metrics when predicting the `</think>` token. Higher Top1 and DF values indicate greater model confidence that the next token to be generated is `</think>`. Lower Entropy signifies that the model's probability distribution for the next token is more sharply peaked, indicating less uncertainty.

This timing captures the internal states of the LRM after processing the entire pre-filled segment, indicating its readiness to finalize thinking by the `</think>` tag and its confidence level in deciding whether to prolong reasoning. Based on our experimental observations, the top-1 prediction token at this point is consistently the `</think>` tag. Therefore, instead of focusing on prediction correctness, we analyze how confidently the model makes this decision. Specifically, we evaluate three key metrics derived from the softmax distribution: the highest probability value (Top1), the entropy of the distribution, and the difference between the highest and second-highest probability values (DF).

Results and Analysis. The results are presented in Table 2. The NT questions exhibit significantly higher prediction confidence metrics, observed when the model is predicting the `</think>` tag that terminates the pre-filled thinking placeholder content. The average Top1 probability is highest for NT (78.04). The average Entropy is lowest (1.06), and the average DF is highest (71.83). These results indicate that, at this critical moment before the thinking-end signal, the LRM's internal state is already highly determined, suggesting it is "ready" to generate the answer directly.

In contrast, the IT and ET modes, both of which ultimately involve further thinking, show relatively lower and very similar levels of confidence. The average Top1 (70.87) and average DF (62.62) for IT are slightly higher than those for ET (average Top1 70.12, average DF 61.33), but the difference between them is small. Their average Entropy values (IT 1.28, ET 1.25) are also close to each other and significantly higher than the NT mode, by approximately 19%. The lower prediction confidence for the `</think>` token in ET and IT modes indicates a greater internal uncertainty within the LRM. Despite the prompt stating that thinking is finished, the LRM's internal state for these questions suggests a lower conviction to terminate the reasoning process. This hesitation may stem from the LRM's assessment that the question requires more reasoning steps beyond what the pre-filled prompt provides, making it less "convinced" by the save-thinking instruction and and favoring continued reasoning over immediate termination.

Findings. The LRM exhibits higher prediction confidence and lower uncertainty for the `</think>` tag in NT mode, indicating a deterministic internal state favoring immediate answer generation. In contrast, ET and IT modes show lower confidence, reflecting hesitation to terminate reasoning despite save-thinking instructions. This confidence gap offers a key insight: higher confidence in thinking termination may be necessary for skipping reasoning, helping explain the behavioral divergence.

3.3 Attention from Thinking to Generation

Motivation. Upon processing the full prompt including the full pre-filled thinking segment and the `</think>` token, the LRM is prepared to generate the answer. We next investigate whether the LRM displays distinct attention states when shifting from input processing to answer generation, particularly in relation to the observed thinking modes. To do this, we analyze layer-wise attention activation vectors when the LRM predicts its first output token. The goal is to identify distinct high-dimensional patterns or clusters corresponding to NT, ET, and IT modes, thereby revealing how the internal attentional state reflects or precipitates the model's behavioral divergence.

Results and Analysis. As shown in Figure 2, the PCA visualization of the last layer's attention activation reveals distinct clustering patterns corresponding to the different thinking modes. The samples belonging to the NT mode tend to form a cluster that is largely separate from the ET mode. This separation suggests that the internal attention state before generating the first token is notably different when the model proceeds directly to an answer

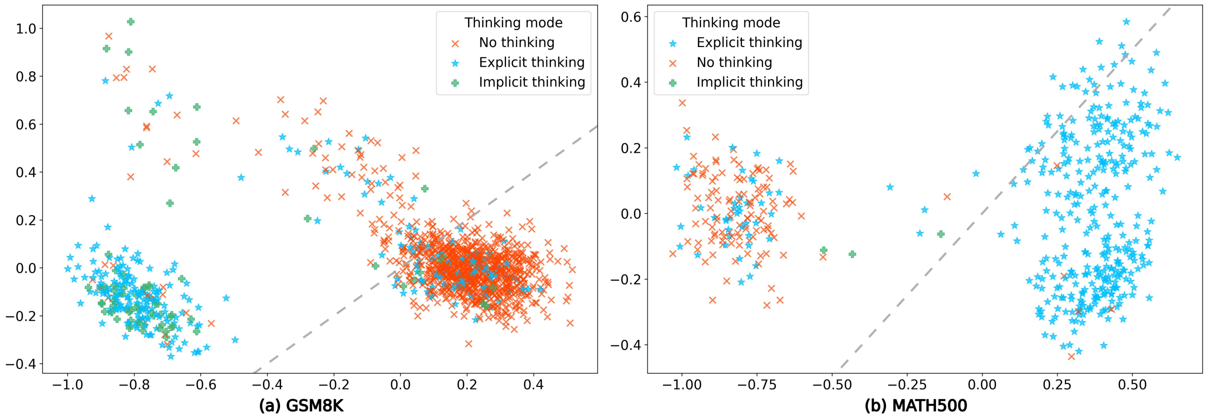

Figure 2: PCA visualization of attention activation from the last layer for all samples in GSM8K and MATH500.

compared to when it engages in further thinking. Due to the smaller sample size of IT samples, they do not exhibit a clear clustering characteristic on their own. However, observing the distribution of IT samples, particularly in the GSM8K dataset, many IT samples are distributed within the ET cluster, suggesting that the internal attention activation patterns of the IT mode are more similar to those of the ET mode than the NT mode.

Further Analysis. To quantitatively assess the separation between the clusters based on layer-wise attention activation, we compute the Davies-Bouldin Index (DB Index) for each layer. The DB Index is a metric for evaluating clustering algorithms. A lower DB Index indicates better separation between clusters. The index is defined as:

$$DB = \frac{S_1 + S_2}{D_{1,2}}, \quad (1)$$

where S_1 and S_2 are the average dispersion within the two clusters, respectively, and $D_{1,2}$ is the distance between the centroids of the two clusters. Due to the relatively small number of IT samples, we focus our analysis mainly on ET and NT.

As shown in Figure 3, we observe a notable trend in the DB Index values across the 64 layers in QwQ-32B. For both datasets, the DB Index starts relatively high in the initial layers, indicating that the attention activation patterns of NT and ET samples are less distinct. However, there is a sharp decrease in the DB Index value starting around Layer 5 for both datasets. This indicates that the clusters corresponding to NT and ET thinking modes become significantly more separated in terms of their attention activation patterns in early layers. The DB Index then remains at a consistently lower level throughout the subsequent layers,

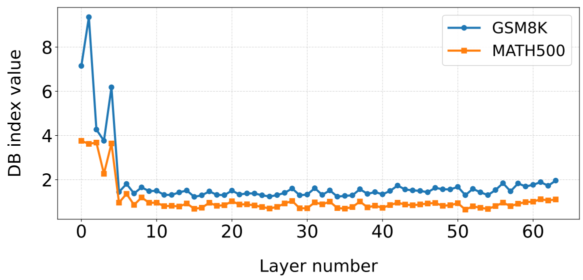

Figure 3: Davies-Bouldin Index (less is better) calculated for the NT and ET clusters based on layer-wise attention activation across all 64 layers.

suggesting that this learned difference in attention focus is maintained as the input signal propagates through the deeper layers of the LRM. This early and sustained divergence in layer-wise attention activation provides strong evidence that the LRM develops distinct internal representations for processing prompts that lead to either direct answers (NT) or require further thinking (ET).

Furthermore, we observe that the trend of the DB Index is remarkably similar for both the GSM8K and MATH500 datasets. This consistency across different datasets suggests that the mechanism by which the LRM differentiates between ET and NT is an inherent property of the model, rather than being strongly influenced by the specific dataset.

Findings. Attention activation patterns diverge significantly between NT and ET modes, with distinct clustering emerging in early layers. The consistent DB Index trends across datasets highlight that behavioral divergence stems from inherent model architecture rather than task-specific factors. This layered separation underscores the role of early-stage attention dynamics in shaping reasoning strategies. In summary, the tendency to save thinking is partly

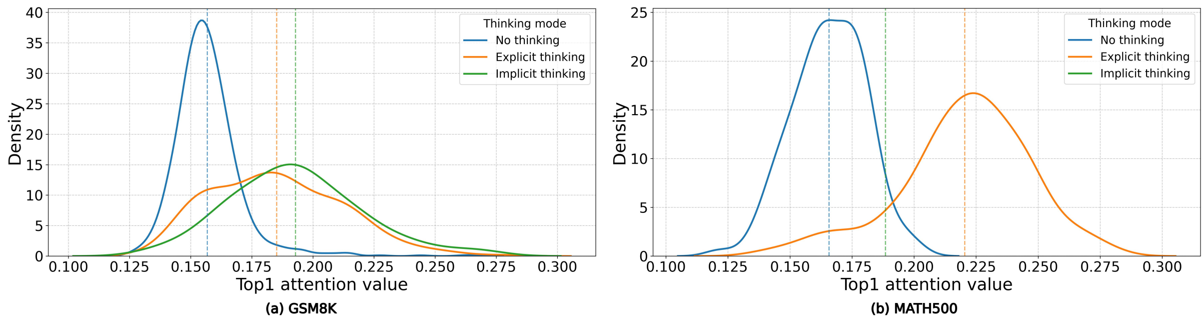

Figure 4: The density distribution of attention scores under different thinking modes. Implicit thinking samples in MATH500 are too few to plot a density curve. The green dashed line indicates their average value.

Datasets	NT		IT		ET	
	Top1	DF	Top1	DF	Top1	DF
GSM8K	15.68	3.67	19.29	11.39	18.52	9.89
MATH500	16.57	4.33	18.85	10.58	22.04	14.30
Average	15.78	3.74	19.27	11.36	20.56	12.45

Table 3: Top1 and DF attention scores across different thinking modes and datasets.

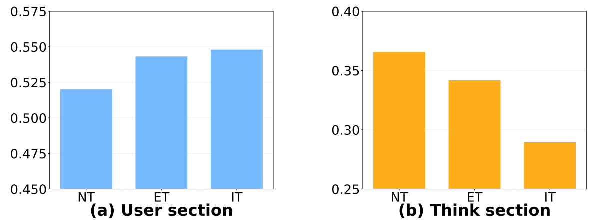

Figure 5: Attention scores across prompt sections for different thinking modes on the MATH500 dataset. The results on GSM8K are given in Appendix B and similar findings can be obtained.

shaped by early internal states, which guide its subsequent processing path before the model begins to actually generate content.

3.4 Attentional Focus on Input Sections

Motivation. The preceding analysis indicates significant differences in the internal states of the LRM across different modes. We now explore the underlying reasons for this difference. Specifically, we aim to identify which input tokens and prompt sections the LRM prioritizes at the onset of generation. To achieve this, we analyze the attention output from the last layer (Layer 63), averaging across all heads to identify the top-k most attended tokens in the input. This analysis highlights the key parts of the input that may influence the LRM's varying behaviors during generation.

The attention score for the first token t_1 attending to an input token x_i in the last layer is:

$$\mathrm{AvgAttn}(t_1, x_i) = \frac{1}{H}\sum_{h=1}^{H} \mathrm{Attn}_h(t_1, x_i) \quad (2)$$

where H is the number of attention heads in the last layer, and $\mathrm{Attn}_h(t_1, x_i)$ denotes the attention weight from the first generated token t_1 to the input token x_i for the attention head h.

Based on our observation that the token with the highest attention score consistently belongs to the initial "user" role token, we quantify the attention distribution towards this specific "user" token using metrics Top1 attention score, and DF. In this analysis, Top1 attention score is simply the average attention score directed towards the "user" token, and DF is the difference between the attention to the "user" token and the next most attended token.

Results and Analysis. Table 3 presents the average attention scores and differences for the first generated token's attention to the "user" token across different thinking modes. The samples in the NT mode exhibit significantly lower average Top1 attention scores (15.78) and average DF values (3.74) compared to the samples in the IT mode (Top1 19.27, DF 11.36) and ET mode (Top1 20.56, DF 12.45). This significant numerical difference is visually confirmed in the attention score density distribution shown in Figure 4. As depicted, the peak of the attention score distribution for the NT mode is notably shifted towards the left compared to the other two modes, indicating that the first token generated in this mode generally has lower attention scores directed towards the "user" token.

This result directly reveals a difference in the model's internal processing focus across thinking modes. When the LRM exhibits NT behavior, its reduced attention to the initial "user" role token during the generation of the first token suggests

LRM setups	Question groups	GSM8K		MATH500	
		Accuracy	Length	Accuracy	Length
QwQ-32B (w/ pre-filled thinking)	NT	37.76	35	52.54	29
	IT	92.00	4037	100.00	710
	ET	96.35	3505	97.63	9296
	Average	53.28	999	87.00	7059
QwQ-32B (w/o pre-filled thinking)	NT	94.09	5049	99.15	5714
	IT	94.67	7583	100.00	3965
	ET	95.61	5197	95.78	10788
	Average	94.32	5227	96.60	9549

Table 4: Performance across different LRM setups and question groups.

it is less anchored to the beginning of the user's request. This implies that relatively more attention is allocated to other parts of the input, including the pre-filled thinking content and formatting tokens. This shift in attention focus, from the original request's starting point towards the pre-filled completion signal, suggests that the LRM perceives the pre-filled thinking segment as having provided sufficient context and a signal to bypass further reasoning on the original question. Consequently, this internal state encourages the model to proceed directly to final answer generation, characteristic of NT behavior. Conversely, the higher attention to the user token in thinking modes (IT/ET) indicates a stronger focus on the original task, aligning with the model's tendency to re-engage in reasoning.

Further Analysis. To further investigate how the model's attention is distributed across different parts of the input prompt in each thinking mode, we segment the input based on the "assistant" role token. The prompt is divided into three main sections: the "user" section, the "thinking" section (the pre-filled content between the <think> and </think> tags), and the "other" section (including the "assistant" token itself and formatting tokens). For each thinking mode, we compute the sum of attention scores for tokens within each section.

As illustrated in Figure 5, a distinct pattern emerges regarding the distribution of attention across prompt sections in different thinking modes. Consistent with our earlier findings on attention to the initial "user" token, the NT mode exhibits a significantly lower sum of attention directed towards the "user" section and a markedly higher sum of attention towards the "thinking" section compared to the ET and IT modes. This shift in attention focus from the original task context to the pre-filled thought suggests that in the NT mode, the LRM perceives the pre-filled content as sufficient. This internal state occasionally leads the LRM to bypass further reasoning and proceed directly to answer generation. Conversely, the higher sum of attention allocated to the "user" section in ET and IT modes aligns with the LRM's tendency to re-engage with the original task for additional reasoning.

Findings. The NT mode shifts attention away from user instructions toward pre-filled thinking content. In contrast, the ET and IT modes retain focus on the task context, likely facilitating their re-engagement in additional reasoning. This divergence reflects a tension between external prompts and internal reasoning demands. Future work could explore whether explicitly guiding attention patterns improves instruction adherence in RL-trained LRMs.

4 Analysis of Reasoning Performance

4.1 Thinking Modes and Performance

In this section, we analyze how different thinking modes affect the final output quality. We report key performance metrics—specifically accuracy and output length—for each thinking mode. Furthermore, we compare the results to a baseline where the LRM processes the same set of questions using a standard prompt without the pre-filled thinking section. This comparison helps assess the influence of the pre-filled save-thinking instruction on the LRM's ability to generate correct answers.

We divide questions in GSM8K and MATH500 into three groups based on the LRM's behavior (NT, IT, or ET) under the save-thinking instruction (i.e., with pre-filled thinking segment). Then, we evaluate the performance of each group with a baseline that uses a standard prompt and can engage in reasoning (i.e., without pre-filled thinking segment). Table 4 presents the accuracy and output length results for each group. In the NT group, when

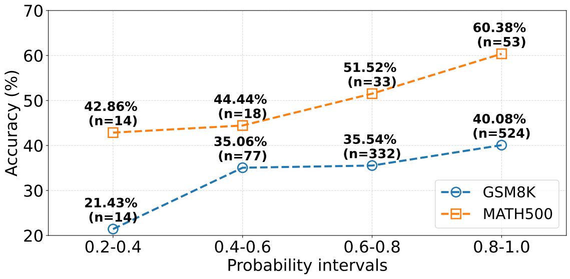

Figure 6: Top1 softmax probability vs. acc. for NT.

thinking is bypassed, the LRM exhibits a dramatic reduction in generated token count compared to the baseline, saving over 99% of tokens. However, this significant decrease in output length comes at the cost of accuracy, with GSM8K accuracy falling from 94.09% to 37.76% and MATH500 accuracy dropping from 99.15% to 52.54%.

In contrast, in the ET group, the pre-filled thinking segment significantly shortens the output length while still achieving high accuracy. Notably, its accuracy even surpasses the baseline condition without pre-filling. For instance, the output length for the ET group decreased by approximately 32.6% on GSM8K and 13.8% on MATH500 compared to its baseline, yet its accuracies outperform the baseline ET mode. Besides, although the sample number of the IT group is limited, preliminary observations indicate it also exhibits substantial length reductions while maintaining robust accuracy. Additionally, baseline results (w/o pre-filled thinking segment) also show ET-categorized questions elicit longer thought chains than NT-categorized questions. This implies that these questions in the ET group are inherently more complex for the LRM. These findings collectively suggest that prompting LRMs to skip thinking can enhance efficiency without a necessary trade-off in performance, and may even yield accuracy improvements.

4.2 Confidence-Accuracy Correlation

Furthermore, we observe a clear relationship between the LRM's internal prediction confidence and the answer accuracy in NT mode. As illustrated in Figure 6, higher Top1 softmax probabilities for the token following the pre-filled segment correspond to increased accuracy in NT group. This suggests that a more confident LRM, after processing the pre-filled completion signal, is more likely to generate a correct direct answer without engaging in further explicit or implicit thinking.

Questions	Golden answers	Answers
I have 10 liters of orange drink that are two-thirds water and I wish to add it to 15 liters of pineapple drink that is three-fifths water. But as I pour it, I spill one liter of the orange drink. How much water is in the remaining 24 liters?	There are 15 x 3/5 = 9 liters of water from the 15 liters pineapple drink. After 1 liter of orange drink was spilled, there were 10 - 1 = 9 liters of orange drink left. Out of the 9 liters, 9 x 2/3 = 6 liters are water. Thus, there are a total of 9 + 6 = 15 liters of water out of the 24 liters. The final answer is $\boxed{15}$.	The final answer is $\boxed{16}$.
A curve is parameterized by $(x, y) = (t^3 + 7, -3t^2 - 6t - 5)$. Find the point the curve passes through at $t = 2$.	At $t = 2$, $(x, y) = (2^3+7, -3\cdot2^2-6\cdot2-5) = \boxed{(15, -29)}$.	The final answer is $\boxed{(15, -23)}$.

Table 5: Examples of incorrect answers in the NT mode.

4.3 Answer Error Analysis in NT Questions

When manually examining the incorrect answers generated in the NT mode, we discover an interesting phenomenon: despite bypassing explicit reasoning steps, the LRM frequently produces answers that closely resemble the correct solution, often differing by just a single digit or a minor numerical variation. Table 5 presents two real examples illustrating this phenomenon: one from the GSM8K dataset and one from the MATH500 dataset. This suggests that even when the LRM skips explicit reasoning, it still engages in partial numerical processing, leading to near-correct outputs. The absence of a complete thinking in the NT mode likely leads to the calculation errors.

5 Conclusion and Future Work

In this work, we investigate how RL-trained LRMs respond to save-thinking instructions, uncovering three distinct behavioral modes: NT, ET, and IT. NT exhibits higher termination confidence, with early-layer attention patterns diverging fundamentally from ET and IT. NT shifts focus to pre-filled thinking content, while ET and IT maintain task-specific attention. NT reduces output length but decreases accuracy, whereas ET preserves accuracy with shorter outputs. These findings expose critical inconsistencies in attention allocation and reasoning reliability, demanding improved training strategies for improving LRMs.

6 Limitations

While our study provides valuable insights into the internal mechanisms of RL-trained LRMs under save-thinking instructions, several limitations should be acknowledged. First, our study relies on two mathematical reasoning datasets, GSM8K and MATH500, which, while representative of structured reasoning tasks, may not fully capture the diversity of reasoning challenges encountered in real-world applications. Second, our study does not explore potential mitigation strategies for improving instruction adherence in RL-trained LRMs. Investigating alternative RL objectives, fine-tuning approaches, or adaptive prompting techniques could help enhance model reliability and efficiency.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: controlling how long A reasoning model thinks with reinforcement learning. *CoRR*, abs/2503.04697.

Avinash Anand, Mohit Gupta, Kritarth Prasad, Navya Singla, Sanjana Sanjeev, Jatin Kumar, Adarsh Raj Shivam, and Rajiv Ratn Shah. 2024. Mathify: Evaluating large language models on mathematical problem solving tasks. *CoRR*, abs/2404.13099.

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. *CoRR*, abs/2503.09567.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2024. Do NOT think that much for 2+3=? on the overthinking of o1-like llms. *CoRR*, abs/2412.21187.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. In *ICLR*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware LLM reasoning. *CoRR*, abs/2412.18547.

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. In *NeurIPS Datasets and Benchmarks*.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020,*.

Chenghao Li, Chaoning Zhang, Yi Lu, Jiaquan Zhang, Qigan Sun, Xudong Wang, Jiwei Wei, Guoqing Wang, Yang Yang, and Heng Tao Shen. 2025. Syzygy of thoughts: Improving llm cot with the minimal free resolution. *CoRR*, abs/2504.09566.

Yule Liu, Jingyi Zheng, Zhen Sun, Zifan Peng, Wenhan Dong, Zeyang Sha, Shiwen Cui, Weiqiang Wang, and Xinlei He. 2025. Thought manipulation: External thought can be efficient for large reasoning models. *CoRR*, abs/2504.13626.

Xindi Luo, Zequn Sun, Jing Zhao, Zhe Zhao, and Wei Hu. 2024. Knowla: Enhancing parameter-efficient finetuning with knowledgeable adaptation. In *NAACL*, pages 7153–7166.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025. Reasoning models can be effective without thinking. *CoRR*, abs/2504.09858.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *NeurIPS*.

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

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2025. Introducing openai o1.

Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Iqbal khan, and Arsalan Shahid. 2024. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *CoRR*, abs/2408.13296.

Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025. Optimizing test-time compute via meta reinforcement fine-tuning. *CoRR*, abs/2503.07572.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Ben Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *CoRR*, abs/2503.16419.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025a. Towards large reasoning models: A survey of reinforced reasoning with large language models. *CoRR*, abs/2501.09686.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. 2025. Towards thinking-optimal scaling of test-time compute for LLM reasoning. *CoRR*, abs/2502.18080.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Rongzhi Zhu, Xiangyu Liu, Zequn Sun, Yiwei Wang, and Wei Hu. 2025. Mitigating lost-in-retrieval problems in retrieval augmented multi-hop question answering. *CoRR*, abs/2502.14245.

A Experimental Details

The primary large reasoning model (LRM) used in our experiments is QwQ-32B (QwQ-32B), utilizing the version downloaded from Hugging Face on March 31, 2025. For all experiments conducted in this study, we employed greedy decoding by setting the do_sample parameter to False, and the maximum number of tokens allowed for generation was set to 20480. All experiments were performed on a single NVIDIA A800 GPU.

To determine the correctness of the model's output, we first extracted the final answer using regular expressions. We then calculated the number of exact matches (EM) with the golden answer. For instances initially judged as incorrect by this automated process, we conducted manual verification to check for any misjudgments by the matching script. Finally, the samples that are exact matches and those manually verified as correct are combined to calculate the final accuracy rate.

B Attention Scores across Prompt Sections on GSM8K

This section provides supplementary results for the attention distribution analysis on the GSM8K dataset, complementing the MATH500 results presented in the main text. Figure 7 illustrates the summed attention scores directed towards the "user" section and the "think" section of the prompt for the NT, ET, and IT modes when processing questions from the GSM8K dataset.

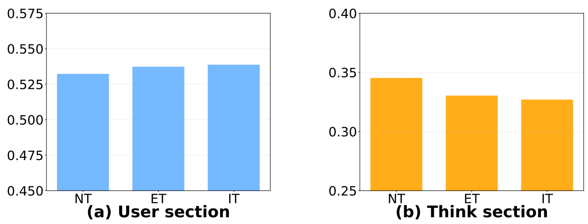

Figure 7: Attention scores across prompt sections for different thinking modes on GSM8K.