From Reasoning to Answer: Empirical, Attention-Based and Mechanistic Insights into Distilled DeepSeek R1 Models

Jue Zhang[1]*, **Qingwei Lin**[1], **Saravan Rajmohan**[1], **Dongmei Zhang**[1]
[1] Microsoft

Abstract

Large Reasoning Models (LRMs) generate explicit reasoning traces alongside final answers, yet the extent to which these traces influence answer generation remains unclear. In this work, we conduct a three-stage investigation into the interplay between reasoning and answer generation in three distilled DeepSeek R1 models. First, through empirical evaluation, we demonstrate that including explicit reasoning consistently improves answer quality across diverse domains. Second, attention analysis reveals that answer tokens attend substantially to reasoning tokens, with certain mid-layer *Reasoning-Focus Heads* (RFHs) closely tracking the reasoning trajectory, including self-reflective cues. Third, we apply mechanistic interventions using activation patching to assess the dependence of answer tokens on reasoning activations. Our results show that perturbations to key reasoning tokens can reliably alter the final answers, confirming a directional and functional flow of information from reasoning to answer. These findings deepen our understanding of how LRMs leverage reasoning tokens for answer generation, highlighting the functional role of intermediate reasoning in shaping model outputs.

1 Introduction

Recent progress in Large Language Models (LLMs) has led to the development of Large Reasoning Models (LRMs), such as OpenAI's o1 (OpenAI, 2024) and DeepSeek's R1 (Guo et al., 2025), which generate explicit intermediate reasoning traces before producing final answers. This approach, rooted in early Chain-of-Thought (CoT) (Wei et al., 2022) prompting techniques, exemplifies a form of test-time scaling, which improves model performance by allocating additional computation during inference.

LRMs typically output two distinct segments: a *Reasoning* segment consisting of reasoning tokens often enclosed within `<think>` and `</think>`, and an *Answer* segment providing the final, self-contained response to the query. This separation gives rise to a fundamental question: *Do LRMs actually leverage the reasoning tokens to generate answers?* It is conceivable that the reasoning traces serve merely as post-hoc justifications, rather than functioning as essential components in answer generation (Lanham et al., 2023). This question is closely tied to active research areas including reasoning effectiveness (Ma et al., 2025), reasoning faithfulness (Chen et al., 2025), and model behavior monitoring (Baker et al., 2025). Despite growing efforts in enhancing the overall capabilities of LRMs, the interplay between reasoning and answer segments remains poorly understood.

To address this gap, we conduct a three-stage progressive analysis to examine how reasoning contributes to answer generation in Large Reasoning Models, with key insights summarized in Table 1. We focus on three distilled DeepSeek R1 models, i.e., R1-Llama-8B, R1-Qwen-7B, and R1-Qwen-1.5B (Guo et al., 2025), chosen for their accessible reasoning traces and moderate model sizes. We begin with an empirical evaluation comparing model performance with and without reasoning traces. Results show that including reasoning generally improves answer quality across diverse domains, extending prior work that focused primarily on the math and code domains (Ma et al., 2025).

While these results indicate that explicit reasoning *can* enhance answer quality, it remains unclear *how* answer tokens incorporate information from reasoning tokens. Since attention mechanisms govern information flow in transformer-based models, we next analyze the attention patterns between the reasoning and answer segments. We find that answer tokens attend substantially to reasoning tokens, and that certain *Reasoning-Focus Heads*

*Correspondence to: juezhang@microsoft.com.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 3986–4003
November 4-9, 2025

Investigation Directions	Key Observations and Insights
Empirical Study: Does Reasoning Improve Answer Quality? *(Section 3)*	• Explicit reasoning improves answer quality across diverse domains. • Gains are more pronounced in distilled R1 models than in the full R1 model.
Attention Analysis: How Does Answer Attend to Reasoning? *(Section 4)*	• Answer tokens strongly attend to reasoning tokens. • Certain mid-layer attention heads closely track reasoning progression. • Attention paths often terminate at the positions of reflection-related tokens. • Attention sink effects persist after reasoning-related model distillation. • <think> and </think> tokens function more likely as structural markers. • Potential application of reasoning-focus heads in reasoning failure debugging.
Mechanistic Intervention: Can Small Reasoning Changes Shift the Answer? *(Section 5)*	• Modifying the activations of key reasoning tokens can flip the answer. • Evidence suggests reasoning-to-answer information flow in mid-model layers.

Table 1: Summary of findings across empirical studies, attention analysis, and mechanistic interventions.

(RFHs) consistently track the reasoning process, including self-reflective steps. We further present a case study showing that RFHs can make the source of reasoning errors far more interpretable than head-averaged attention, showing their potential application in debugging failures in reasoning traces.

Finally, since strong attention alone does not guarantee functional dependence, we perform *mechanistic interventions* by perturbing reasoning traces in a controlled setting. These experiments reveal that small modifications to reasoning can indeed flip the answer output.

Our contributions are summarized as follows:

- We broaden the empirical investigation of the influence of reasoning on answer quality in LRMs to a wider range of domains beyond the commonly studied mathematics and code.
- We analyze attention patterns between reasoning and answer in LRMs, uncovering novel findings such as specific attention heads tracking the progression of reasoning during answer generation.
- We augment the attention analysis with a detailed mechanistic intervention, demonstrating that perturbations to the activations of reasoning tokens can directly influence the generated answers.

2 Related Work

Performance Impact of Reasoning in Large Reasoning Models. The advent of LRMs has spurred interest in analyzing the relationship between reasoning and model performance empirically (Muennighoff et al., 2025; Marjanović et al., 2025; Ma et al., 2025; Ballon et al., 2025; Su et al., 2025; Jahin et al., 2025). While most studies examine the correlation between reasoning length and answer quality, we do not constrain reasoning length, focusing instead on overall response quality. The study most related to ours is (Ma et al., 2025), which also compares models with and without reasoning. However, their emphasis still lies in efficiency, with evaluations confined to math and code. Our study extends this comparison to broader open-domains, demonstrating the general improvements in answer quality brought by explicit reasoning.

Measuring Faithfulness in Model Reasoning. A model's reasoning is considered faithful if it accurately reflects the model's internal decision-making process (Jacovi and Goldberg, 2020). A commonly used method to assess faithfulness is to alter the reasoning tokens and observe how these changes affect the model's output. Although some argue that this approach primarily evaluates self-consistency rather than true faithfulness (Parcalabescu and Frank, 2024), several recent studies (Lanham et al., 2023; Atanasova et al., 2023; Turpin et al., 2023) have employed it to assess the faithfulness of chain-of-thought reasoning in non-LRMs. For example, four manipulation strategies (i.e., early answering, introducing errors, paraphrasing, and adding filler tokens) were used to examine the impact on model predictions (Lanham et al., 2023). More recent work has extended such analyses to LRMs (Baker et al., 2025; Arcuschin et al., 2025; Chen et al., 2025; Chua and Evans, 2025; Marjanović et al., 2025). For instance, (Chen et al., 2025) used paired prompts, with and without a hint, to test whether the model explicitly acknowledges using the hint.

In our empirical evaluation, we also manipulate the reasoning content by comparing the with and without reasoning settings, a method loosely related to the early answering strategy in (Lanham et al., 2023). However, our objective differs: in-

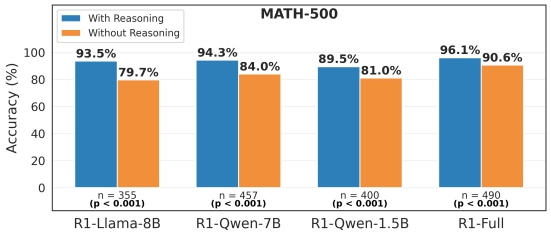
stead of measuring the answer flip rate to infer faithfulness, we focus on changes in overall accuracy to assess the functional contribution of reasoning.

Mechanistic Interpretability of Model Reasoning. Mechanistic interpretability aims to understand how neural networks operate by identifying the specific internal computations and structures responsible for their behavior (Olah et al., 2020; Meng et al., 2022; Geiger et al., 2021; Geva et al., 2021; Bereska and Gavves, 2024; Mueller et al., 2025). Understanding model reasoning has been a central focus within this field. Prior to the emergence of LRMs, several studies employed various mechanistic interpretability techniques to analyze chain-of-thought reasoning (Cabannes et al., 2024; Hou et al., 2023; Wang et al., 2024; Dutta et al., 2024; Tan, 2023; Li et al., 2024). For instance, attention analysis has been used to reconstruct the model's reasoning tree in multi-step reasoning tasks (Hou et al., 2023). In our work, we similarly leverage attention pattern analysis, but specifically to investigate how final answers attend to intermediate reasoning steps.

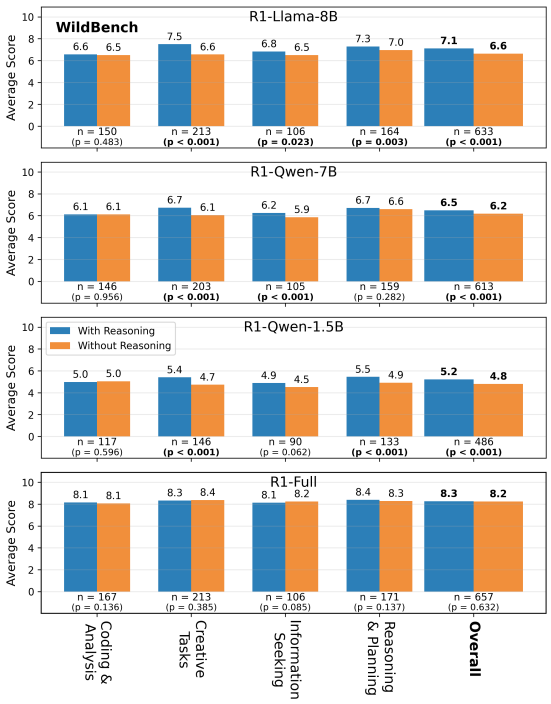
Following the development of LRMs, a number of mechanistic interpretability studies have explored reasoning-related features specific to these models (Galichin et al., 2025; Baek and Tegmark, 2025; Hazra et al., 2025; Ameisen et al., 2025; Venhoff et al., 2025). For instance, sparse features in the MLP layers have been identified and manipulated to influence reasoning behaviors (Galichin et al., 2025). In contrast to these approaches, our work centers on tracing the flow of information from reasoning steps to the final answer, emphasizing the study on the attention modules of LRMs.

3 Empirical Evaluation: Does Reasoning Improve Answer Quality?

We begin by empirically evaluating whether and how reasoning traces influence answer quality, treating the models as black boxes. This serves as an initial assessment of the impact of reasoning on final outputs. We adopt two datasets: MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024) for mathematics and WildBench (Lin et al., 2025) for real-world queries that cover diverse domains. Evaluations are conducted under two settings: **with and without reasoning traces**. For the latter, we adopt the suppression method from (Ma et al., 2025), using *"<think>\nOkay, I think I have finished thinking.\n</think>"* to bypass reasoning.[1]



(a) MATH-500: Answer accuracy across R1 model variants.



(b) WildBench: Average scores by task domain and model.

Figure 1: Answer quality comparison for DeepSeek R1 models under reasoning and non-reasoning settings, evaluated on the MATH-500 and WildBench datasets. All statistics are computed over $n = a$ samples, and p-values are obtained using a paired t-test. Statistically significant results with $p < 0.05$ are shown in bold.

Following the recommendations in (Guo et al., 2025), we append the instruction *"Please reason step by step, and put your final answer within \boxed{}."* to all math-related queries in the MATH-500 dataset and set the sampling temperature to 0.6 for all R1 model variants. We employ zero-shot prompting for both the MATH-500 and WildBench datasets and adopt the evaluation metrics defined in the original benchmarks. Additional experimental details are provided in Appendix A.

Figure 1 presents evaluation results for three dis-

[1]We also experimented with *"<think>\n\n</think>"*, but found it less effective in suppressing reasoning.

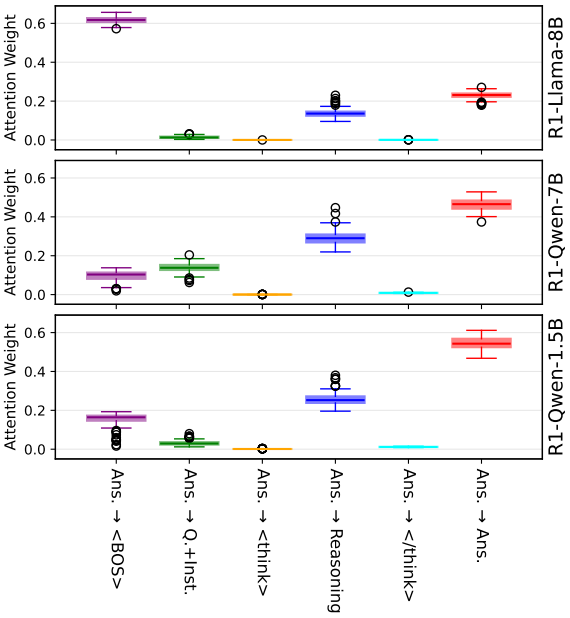


Figure 2: Decomposition of average attention weights from answer tokens to different prompt segments across three models (R1-Llama-8B, R1-Qwen-7B, R1-Qwen-1.5B) using the MATH-500 dataset.

tilled DeepSeek R1 models, alongside the full R1 model (R1-Full) for reference. Several key observations emerge. First, across nearly all models and domains, incorporating reasoning traces leads to improved answer quality, especially on the MATH-500 dataset, where distilled models show $\sim 10\%$ increase in accuracy, compared to $\sim 5\%$ for the full model. Second, in the general (non-math) domains, such as the "Overall" score in Figure 1(b) representing the average performance across all task categories in WildBench, the performance gains from reasoning are again more substantial for the distilled models, whereas R1-Full shows only marginal improvement. This suggests that the full R1 model may already possess sufficient general knowledge, making explicit reasoning during inference less impactful in general domains.

These results, in conjunction with similar trends observed in other math and code-related datasets (Ma et al., 2025), indicate that reasoning tokens contribute to answer quality across diverse tasks, with a more pronounced effect in distilled R1 models compared to the full model.

4 Attention Analysis: How Does Answer Attend to Reasoning?

Having established that reasoning traces enhance answer quality, we now turn to a more detailed analysis of the model's internal behavior. Since our goal is to understand how information flows from the reasoning to the final answer, we focus on the model's attention mechanisms. Specifically, we analyze how answer tokens attend to different parts of the prompt. Here, the term *"prompt"* refers to the input query and instructions plus the model's complete response, since all are required to generate the attention patterns. Concretely, we treat the *"prompt"* as a token sequence composed of six segments: <BOS>, *Query+Instruction (QI)*, <think>, *Reasoning*, </think>, and *Answer*, where <BOS> denotes the beginning-of-sentence token.

Our attention analysis builds on the answer quality traces introduced in the previous section, using 100 samples per dataset and per R1-distilled model. We first provide an overall analysis at the prompt segment level, followed by a more detailed analysis across model layers and attention heads. We also include a representative failure case study to illustrate how attention patterns may reveal the source of an incorrect answer. Due to space limitations, we present results for MATH-500 in the main text and defer the qualitatively similar results for WildBench to Appendix B.

4.1 By Prompt Segment

Figure 2 presents the decomposition of attention weights from answer tokens to various prompt segments. These weights are computed by first averaging over tokens within the *Answer* segment and then aggregating attention towards each destination segment. The results are further averaged across all model layers and attention heads. Based on this aggregated analysis, we observe the following:

- All three R1-distilled models allocate substantial attention from the *Answer* segment to the *Reasoning* segment (blue box-plots), although this cross-segment interaction is weaker than the intra-segment attention within *Answer* (red box-plots).
- The known attention sink phenomenon (Xiao et al., 2024; Gu et al., 2025; Barbero et al., 2025) appears in all three R1-distilled models, with noticeable attention allocated to the <BOS> token (purple box-plots). This indicates that the attention sink mechanism persists after reasoning distillation via supervised fine-tuning.
- Attentions to the <think> and </think> tokens (orange and cyan box-plots) are minimal, suggesting that their primary role is to demarcate different

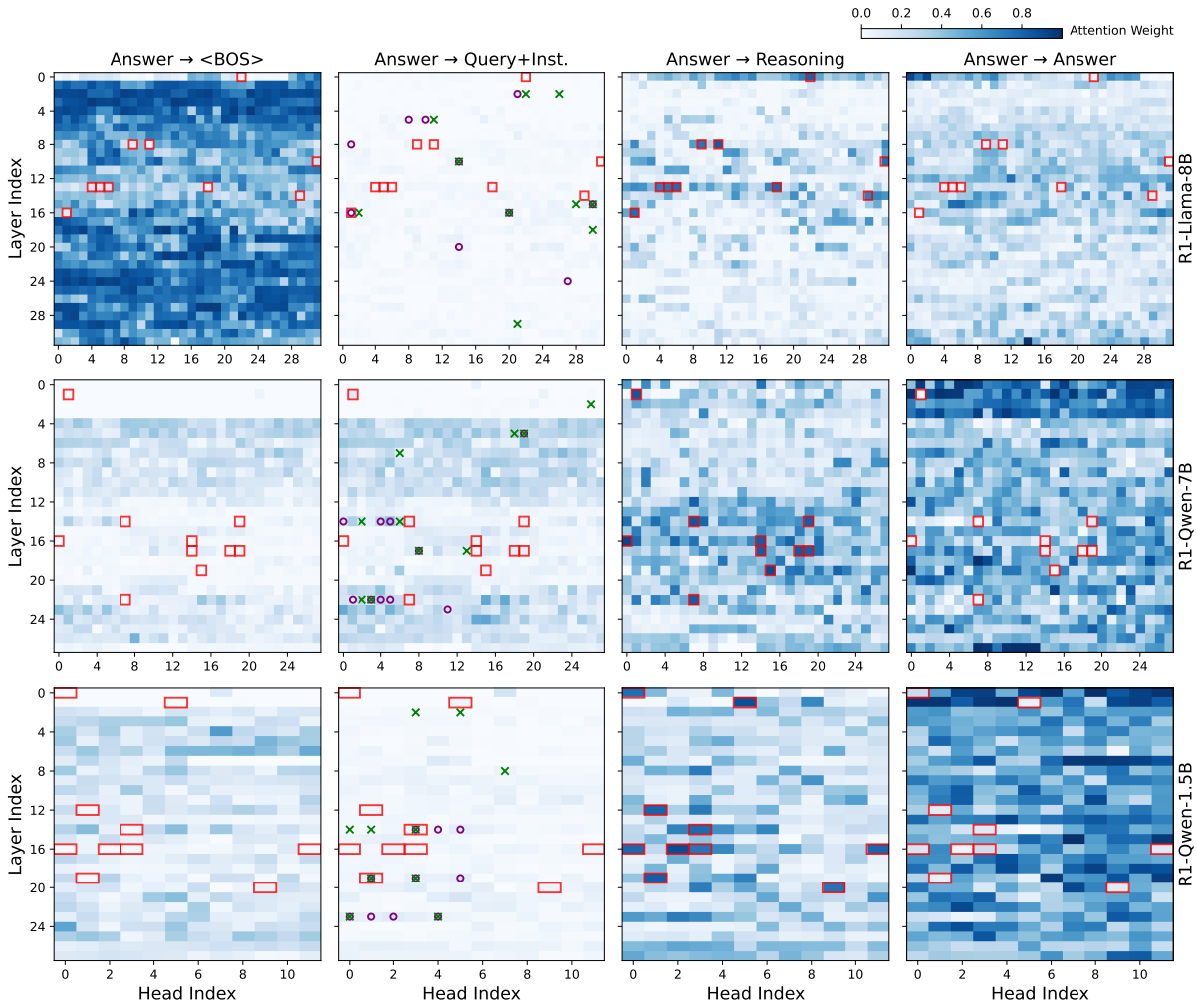


Figure 3: Decomposition of answer attention to different prompt segments across layers and heads for three models (R1-Llama-8B, R1-Qwen-7B, and R1-Qwen-1.5B) using the MATH-500 dataset. Heatmaps show the attention weights from the *Answer* segment to four prompt segments: <BOS>, *Query+Instruction*, *Reasoning*, and *Answer*. Red boxes highlight the top 10 attention heads with the highest weights for *Answer* → *Reasoning*. Additionally, in the second column the top 10 retrieval and induction heads are annotated with "○" and "×", respectively.

prompt segments rather than to store or summarize preceding information.

- Lastly, attention to the *Query+Instruction* segment (green box-plots) is relatively low compared to the *Reasoning* and *Answer* segments. This may be attributed to the greater token distance between the *QI* and *Answer* segments, as well as the shorter average query length in the MATH-500 dataset.

Overall, this segment-level attention analysis reveals that reasoning tokens receive substantial attention from answer tokens, suggesting their significant role in the answer generation process.

4.2 Across Model Layers and Attention Heads

Figure 3 shows the decomposition of attention weights from the *Answer* segment to other prompt segments across model layers and attention heads. The top 10 attention heads with the highest attention weights for the *Answer* → *Reasoning* are highlighted with red boxes. Notably, these heads are primarily concentrated in the middle layers of the models, i.e., **Layers 8–16** for R1-Llama-8B, **Layers 14–22** for R1-Qwen-7B, and **Layers 12–20** for R1-Qwen-1.5B. This pattern aligns with the prevailing understanding that middle layers in LLMs are primarily responsible for comprehension and reasoning by processing information from lower layers while shaping it for decision-making and generation in later layers (Ju et al., 2024). We therefore refer to these heads as **Reasoning-Focus Heads (RFHs)**, a term whose relevance will become more apparent in subsequent case studies.

While some attention heads in early layers (e.g.,

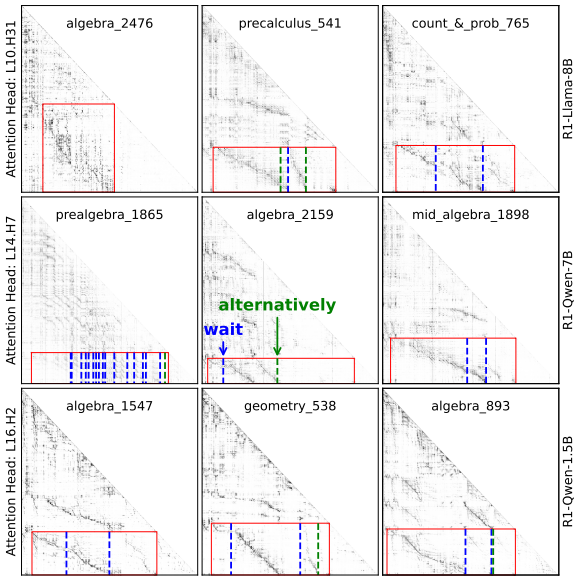


Figure 4: Attention patterns from selected top attention heads of the three distilled R1 models on sample cases from the MATH-500 dataset. The horizontal and vertical axes represent key and query token indices, respectively. Labels on the left y-axis denote the selected reasoning-focused heads (e.g., *L10.H31"* refers to head 31 in Layer 10). The *Answer* $\rightarrow$ *Reasoning* region is highlighted with red boxes. Vertical lines indicate the positions of the tokens "wait" and "alternatively".

Head 22 at Layer 0 for R1-Llama-8B) also receive large attention of *Answer* $\rightarrow$ *Reasoning*, closer inspection reveals that these typically exhibit a near-uniform focus on preceding tokens such as punctuation and prepositions. More details are provided in the Appendix C. This behavior can result in larger aggregate attention to longer segments like *Reasoning*, due to a length bias rather than genuine reasoning focus. In contrast, the middle-layer heads are less affected by this length-based artifact.

For comparison, the second column of Figure 3 presents the top 10 induction heads (Olsson et al., 2022) and retrieval heads (Wu et al., 2024), with implementation details provided in the Appendix D. The minimal overlap among these and the newly identified RFHs suggests the latter may capture novel patterns not accounted for by known induction or retrieval mechanisms.

To dive into these reasoning-focused heads, we select one representative head per model and visualize their attention patterns across three sample cases from the MATH-500 dataset, as shown in Figure 4. The positions of key reflection-related tokens, such as *"wait"* and *"alternatively"*, are also marked with vertical lines.

By focusing on the attention region of *Answer* $\rightarrow$ *Reasoning* (highlighted by red boxes), we observe:

- A prominent *attention trajectory* emerges in most cases. This trajectory typically starts at the top-left of the red box and moves diagonally downward (e.g., *"precalculus_541"*), indicating that as answer generation begins (top of the box), RFHs focus on the beginning of the *Reasoning* segment (left side of the box). As generation progresses, attention shifts accordingly along the reasoning tokens, producing the observed sloped pattern.

- These trajectories often terminate near the reasoning reflection tokens on the horizontal axis (e.g., *"algebra_1547"*, *"mid_algebra_1898"*, and *"algebra_893"*). This alignment reflects the model's awareness that such tokens often appear after a solution, signaling a moment for verification or alternative solution. Together with the alignment at the start of the reasoning, this suggests that RFHs closely track the reasoning process and synchronize answer generation with it.

- Beyond the main trajectory, several parallel attention paths are also observed (e.g., in *"precalculus_541"*, *"algebra_2159"*, and *geometry_538"*). These often originate from and terminate at reflection-related tokens, suggesting they correspond to alternative solutions or verification steps in the *Reasoning* segment.[2] The presence of multiple simultaneous trajectories suggests that RFHs recognize multiple solution paths and attempt to incorporate them during answer generation.

- Not all attention trajectories are interrupted by reflection tokens. For instance, in *"algebra_1547"* and *"count & prob._765"*, attention continues despite encountering the first reflection token. This is because reflections can occur in the middle of a solution to re-evaluate specific steps.

- Lastly, the above observations also apply to the cases with no reflection tokens (e.g., *"algebra_2476"*) or with excessive amount of reflection tokens (e.g., *"prealgebra_1865"*).

In summary, our fine-grained case study across selected RFHs reveals that the distilled R1 reasoning models closely follow the *Reasoning* segment during answer generation. They not only align the

[2]In the *"algebra_2159"* case, the second trajectory starts at a point not marked with a vertical line, due to the model generating a reflection without using explicit keywords like "wait" or "alternatively" instead saying "let us double-check...".

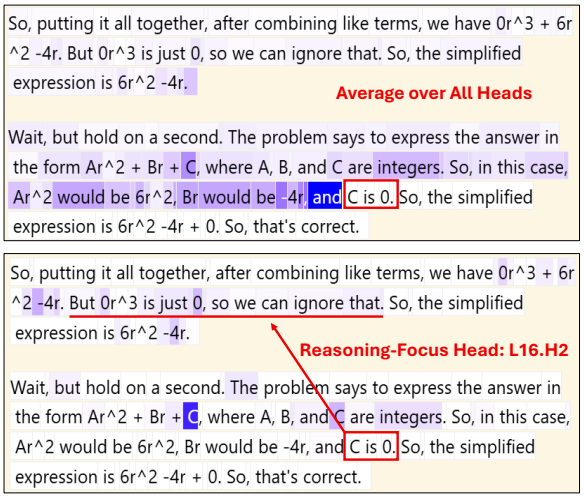


Figure 5: Visualization of attention weights for tokens preceding "*C is 0*" in the reasoning trace of R1-Qwen-1.5B on *"algebra_1547"*. The top panel shows attention averaged over all heads, where the focus is largely on nearby tokens. The bottom panel isolates RFH "L16.H2", which highlights a strong attention link to the phrase "*But* $0r^3$ *is just 0, so we can ignore that*". Color scales of the top and bottom panels are normalized independently, and if comparing the absolute values of the attention weights, the RFH assigns roughly five times more attention to this phrase than the head-average view.

generation process with the reasoning structure but also exhibit sensitivity to reflective cues, reinforcing their deep integration of the reasoning content.

4.3 Case Study: Debugging Reasoning Failures with RFHs

Further investigations suggest that Reasoning-Focus Heads (RFHs) may also provide a useful lens for understanding the model's reasoning behavior within the *Reasoning* segment, particularly when diagnosing failures in reasoning traces. Figure 5 illustrates an application of RFH "L16.H2" in R1-Qwen-1.5B to investigate a failure on the MATH-500 problem *"algebra_1547"* (see Figure 4 for the overall attention pattern). The problem is: "*Simplify* $4(3r^3 + 5r - 6) - 6(2r^3 - r^2 + 4r)$*, and express your answer in the form* $Ar^2 + Br + C$." The ground-truth answer is $6r^2 - 4r - 24$. The model's final prediction, however, is $6r^2 - 4r$, having dropped the constant term. Tracing the chain-of-thought reveals an earlier statement, "*C is 0*," which propagates to the incorrect final result; yet this statement is not justified by the surrounding verbalized reasoning tokens.

To understand where "*C is 0*" originates, we inspect the tokens attended to by the tokens comprising this statement. Figure 5 visualizes attention weights for tokens preceding "*C is 0*" under two settings: (i) averaged over all attention heads and (ii) restricted to the identified RFH. In the head-average setting, the majority of the attention mass falls on nearby tokens, providing little insight into the origin of the statement. By contrast, when focusing on the RFH, it becomes apparent that the erroneous conclusion arises because the model strongly attends to the phrase "*But* $0r^3$ *is just 0, so we can ignore that.*" This phrase correctly concerns the vanishing coefficient of the r^3 term, but the model conflates it with the constant term, incorrectly inferring "*C is 0*". In short, the RFH view makes the source of confusion salient, whereas the head-average view obscures it by focusing local context.

This observation highlights the potential of RFHs as a practical interpretability tool for model debugging: by isolating reasoning-focus heads, we can more easily identify the origin of reasoning errors and trace them back to specific points in the model's internal computation. Additional visualizations of the full reasoning-trace attention maps for this example, are provided in Appendix E.

5 Mechanistic Intervention: Can Small Reasoning Changes Shift the Answer?

As strong attention does not guarantee functional dependence, in this section we perform targeted interventions on the reasoning tokens and trace how these modifications affect the model's output, leveraging mechanistic interpretability tools. Specifically, we use Activation Patching (or Causal Tracing) (Meng et al., 2022), which involves running the model on both clean and corrupted versions of a prompt. We intervene in the corrupted run by replacing certain token activations with those from the clean run, then evaluate whether this correction nudges the output closer to the correct answer. By systematically patching activations at various layers for reasoning and answer tokens, we identify which activations are causally important—i.e., those whose restoration substantially increases the likelihood of the correct outcome.

5.1 Controlled Experiment Settings

Designing controlled settings for conducting activation patching is not trivial (Heimersheim and Nanda, 2024). Here, we introduce a **Contextual Object Comparison** reasoning task. The format of task query is defined as: "Considering [context],

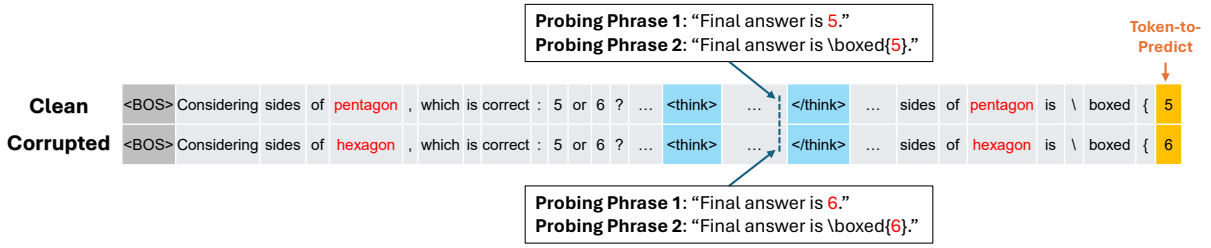


Figure 6: Example of aligned clean and corrupted prompts for activation patching. To ensure comparability, the reasoning and answer segments are padded to equal lengths in both clean and corrupted prompts. A probing phrase is inserted at aligned token positions in the reasoning segment using a consistent format. Activation patching is applied to the probing phase, and its effect is measured by the shift in model output at the token-to-predict position.

which is [comparator]: [A] or [B]?", with "context" determining how objects A and B are compared. For example, as shown in Figure 6, a clean query might be "Considering *sides of pentagon*, which is correct: 5 or 6?", while the corresponding corrupted query is "Considering *sides of hexagon*, which is correct: 5 or 6?". Since these yield different answers, we can examine how patching specific activations causes the output to flip. To ensure generality, we generate dozens of such query pairs across a variety of domains using the OpenAI o1 model. The generation prompt and additional details on data curation are provided in Appendix F.

Using the clean and corrupted query pairs, we instruct reasoning models to generate both reasoning traces and answer tokens, and define the clean and corrupted prompts as the full token sequences comprising the query, instruction, and all generated tokens (mirroring the earlier definition used in the attention analysis). Unlike prior controlled settings (e.g., the Indirect Object Identification task (Wang et al., 2022)), our setup presents two specific challenges. First, since the reasoning and answer tokens are generated, the resulting prompt pairs often differ in length. Second, their formatting can diverge, e.g., reasoning traces may or may not conclude with a phrase like *"**Final Answer**:..."*.

To resolve these issues, we implement a prompt alignment procedure that standardizes the ending phrases in both the reasoning and answer segments across clean and corrupted prompts. Alignment details are provided in Appendix G. Figure 6 presents an example of aligned prompts, which now share the same token length and consistent formatting in answer and reasoning segments. We focus on two common concluding phrase formats in reasoning, reflecting frequently observed patterns.

With the aligned prompts, our activation patching experiments are conducted by replacing the activations of reasoning tokens in the clean prompt with those from the corrupted prompt. Since most reasoning tokens in the clean and corrupted prompts differ, we perform activation patching on the reasoning tokens within the probing phase, as well as on a few final answer tokens. This allows us to observe how these answer tokens leverage the newly introduced activation information after patching. To quantify the effect of the intervention, we use the *logit difference* (Heimersheim and Nanda, 2024), normalizing it by the raw logit differences of the clean and corrupted prompts (see Appendix H for details). Thus, a value approaching 1 indicates that the answer has flipped, while a value near 0 suggests no impact on the final answer.

5.2 Intervention Results

Figure 7 shows the impact of patching residual streams from the clean prompt into the corrupted prompt at each layer for selected reasoning and answer tokens, using the example shown in Figure 6. We observe that patching the answer-flipping token in reasoning (in red) can increase the normalized logit difference by up to about 0.5. This indicates that such an intervention is highly effective in flipping the answer in the corrupted prompt, providing further evidence that the activations of reasoning tokens can influence the model's final output.

Furthermore, by comparing the distributions across model layers for the answer-flipping reasoning token and the final two answer tokens (i.e., "*boxed*" and "{"), we observe that the patching effect on the reasoning token diminishes in later layers, while at similar layers, the patching impact on the answer tokens begins to emerge. This pattern is reminiscent of the Indirect Object Identification task (Wang et al., 2022), where attention heads in these transitional model layers act as information movers—transporting the value of the

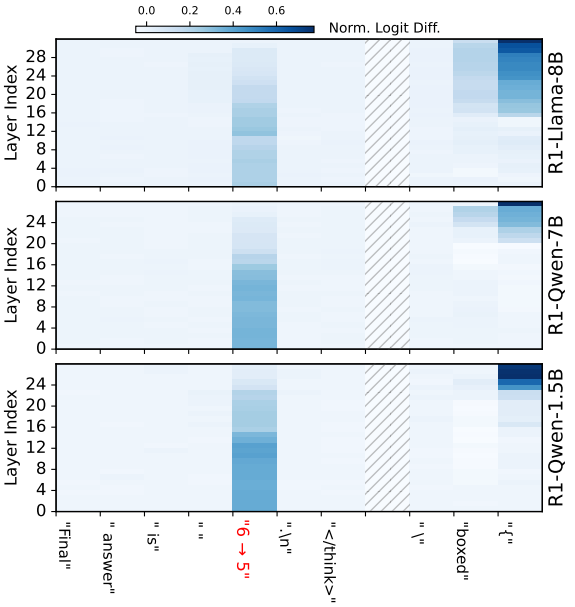


Figure 7: Effect of residual stream patching on normalized logit difference for the target token prediction. The heatmaps show the impact of injecting residual streams from the clean prompt into the corrupted prompt at various token positions, measured across model layers. Residual patching is applied at each token in Probing Phrase 1 within the *Reasoning* segment, as well as selected ending tokens in the *Answer* segment.

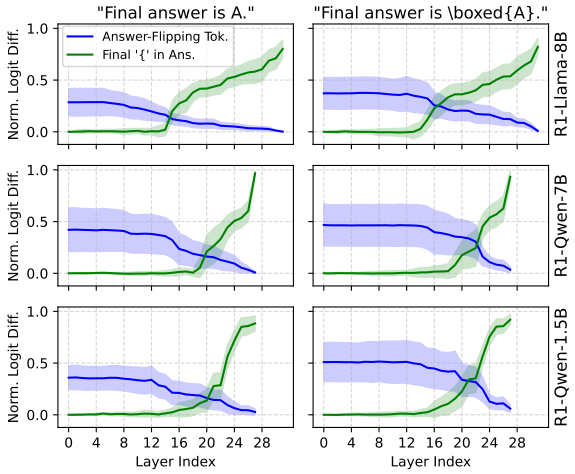


Figure 8: Aggregated results showing how the normalized logit difference evolves across model layers when residual stream patching is applied to specific tokens. Patching is performed at the answer-flipping token in the *Reasoning* segment (blue) and at the preceding answer prediction token (i.e., "{") in the *Answer* segment (green). The left and right columns represent two distinct probing phrases. Shaded regions indicate the distribution across all test samples.

answer-flipping token and embedding it into the residual stream at the position of the final answer token (e.g., "{").

The above observation generalizes to other test cases within this contextual objection comparison scenario. In Figure 8, we illustrate how the normalized logit difference evolves across model layers when residual stream patching is applied to the answer-flipping reasoning token (blue) and the preceding answer prediction token (green). Two distinct probing phrases are considered, with the shaded bands representing variation across test samples. We observe that these transitional model layers emerge consistently across diverse cases and different probing phrases.

Notably, an astute reader may recognize that these transitional layers approximately correspond to the middle layers identified in our earlier attention analysis, reinforcing the presence of reasoning-focused attention heads. Finally, we find that the patching effect is more pronounced for *Probing Phase 2*, which includes the "\boxed{" tag. This heightened effect may arise from the recurrence of the same tag in the answer prediction token, forming a pattern of "...ab...a→b", where a and b correspond to "\boxed{" and the final answer token, respectively. This recurrence potentially activates the induction head (Olsson et al., 2022), thereby facilitating additional information propagation.

6 Conclusion

We presented a multi-faceted investigation into how reasoning traces influence answer generation in large reasoning models, focusing on distilled variants of DeepSeek R1. Our study combined empirical evaluation, attention analysis, and mechanistic intervention to assess whether and how models leverage reasoning tokens during inference. We find that explicit reasoning improves answer quality across diverse domains. From the attention analysis, answer tokens consistently attend to reasoning segments; moreover, within this analysis, *Reasoning-Focus Heads* (RFHs) emerge as heads that track the reasoning process (including self-reflection) and make failure modes interpretable than head-averaged views. Finally, mechanistic interventions show that small changes to reasoning activations can flip final outputs.

Together, these results provide converging evidence for a functional dependence between reasoning and answers, shedding light on the internal dynamics of LRMs and informing efforts to improve faithfulness, controllability, and monitoring.

Limitations

Model Scope. Our analyses focus on three distilled variants of DeepSeek R1 due to their accessibility and tractable size. Although these models exhibit consistent trends across tasks, our findings may not generalize to other LRMs such as the full R1 model or OpenAI o1. Future work should examine whether similar reasoning-to-answer dependencies exist across a broader range of LRMs.

Dataset Coverage and Scale. Although our empirical evaluation spans both math (MATH-500) and open-domain tasks (WildBench), the number of test samples per model is moderate due to computational constraints. While qualitative trends are consistent, more extensive benchmarking would help strengthen statistical confidence in observed effects, particularly for open-ended domains.

Controlled Interventions. Our mechanistic interventions rely on clean-corrupted prompt pairs in a controlled, synthetic reasoning format. While this design allows for precise attribution of answer changes, it may not capture the full range of naturalistic perturbations or adversarial reasoning errors encountered in real-world deployments. Generalizing the intervention results to unconstrained inputs remains an open challenge.

Depth of Mechanistic Analysis. Our mechanistic interpretability work operates at the residual stream level using activation patching. While this reveals causal influence from reasoning tokens to answers, we do not perform in-depth circuit-level tracing or path attribution to identify the precise internal substructures responsible for this information flow. Future work could build on our findings to discover and characterize the underlying circuits that mediate reasoning integration.

Ethics Statement

This work analyzes publicly available models and datasets (DeepSeek R1 variants, MATH-500, WildBench) for research purposes only. No personally identifiable or sensitive data is used. Our goal is to improve transparency and understanding of model reasoning, not to deploy models in real-world applications. While we employ interpretability tools like activation patching, we caution that such methods must be used responsibly. We release code at this URL to support reproducibility.

Acknowledgments

We thank Fangkai Yang, Xiaoting Qin, Zhitao Hou, and Yi Ren for their insightful discussions and feedback. We are also grateful to Bo Qiao for assistance in setting up the experimental environment. We further thank the anonymous reviewers for their careful reading and helpful suggestions, which greatly improved the quality of this work. Finally, we gratefully acknowledge the developers of the open-source `transformer_lens` package (Nanda and Bloom, 2022), which provided a crucial foundation for our analysis.

References

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. 2025. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits*.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. 2025. Chain-of-thought reasoning in the wild is not always faithful. In *Workshop on Reasoning and Planning for Large Language Models*.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–294, Toronto, Canada. Association for Computational Linguistics.

David D. Baek and Max Tegmark. 2025. Towards understanding distilled reasoning models: A representational approach. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. 2025. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *Preprint*, arXiv:2503.11926.

Marthe Ballon, Andres Algaba, and Vincent Ginis. 2025. The relationship between reasoning and performance in large language models – o3 (mini) thinks harder, not longer. *Preprint*, arXiv:2502.15631.

Federico Barbero, Álvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar

Veličković, and Razvan Pascanu. 2025. Why do llms attend to the first token? *Preprint*, arXiv:2504.02732.

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic interpretability for ai safety - a review. *Transactions on Machine Learning Research*.

Vivien Cabannes, Charles Arnal, Wassim Bouaziz, Alice Yang, Francois Charton, and Julia Kempe. 2024. Iteration head: A mechanistic study of chain-of-thought. In *Advances in Neural Information Processing Systems*, volume 37, pages 109101–109122. Curran Associates, Inc.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. 2025. Reasoning models don't always say what they think. *Preprint*, arXiv:2505.05410.

James Chua and Owain Evans. 2025. Are deepseek r1 and other reasoning models more faithful? *Preprint*, arXiv:2501.08156.

Alan Cooney and Neel Nanda. 2023. Circuitsvis. `https://github.com/TransformerLensOrg/CircuitsVis`.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. 2024. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *Transactions on Machine Learning Research*.

Andrey Galichin, Alexey Dontsov, Polina Druzhinina, Anton Razzhigaev, Oleg Y. Rogov, Elena Tutubalina, and Ivan Oseledets. 2025. I have covered all the bases here: Interpreting reasoning features in large language models via sparse autoencoders. *Preprint*, arXiv:2503.18878.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. *Preprint*, arXiv:2012.14913.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2025. When attention sink emerges in language models: An empirical view. In *The Thirteenth International Conference on Learning Representations*.

D. Guo, D. Yang, H. Zhang, et al. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638.

Dron Hazra, Max Loeffler, Murat Cubuktepe, Levon Avagyan, Liv Gorton, Mark Bissell, Owen Lewis, Thomas McGrath, and Daniel Balsam. 2025. Under the hood of a reasoning model.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *Preprint*, arXiv:2404.15255.

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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. 2023. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4919, Singapore. Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Afrar Jahin, Arif Hassan Zidan, Yu Bao, Shizhe Liang, Tianming Liu, and Wei Zhang. 2025. Unveiling the mathematical reasoning in deepseek models: A comparative study of large language models. *Preprint*, arXiv:2503.10573.

Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. 2024. How large language models encode context knowledge? a layer-wise probing study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8235–8246, Torino, Italia. ELRA and ICCL.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. 2023. Measuring faithfulness in chain-of-thought reasoning. *Preprint*, arXiv:2307.13702.

Jiachun Li, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao. 2024. Focus on your question! interpreting and mitigating toxic cot problems in commonsense reasoning. In *Proceedings of the 62nd Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 9206–9230.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2025. Wildbench: Benchmarking LLMs with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations*.

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

Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, Nicholas Meade, Dongchan Shin, Amirhossein Kazemnejad, Gaurav Kamath, Marius Mosbach, Karolina Stańczak, and Siva Reddy. 2025. Deepseek-r1 thoughtology: Let's think about llm reasoning. *Preprint*, arXiv:2504.07128.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. 2025. The quest for the right mediator: Surveying mechanistic interpretability through the lens of causal mediation analysis. *Preprint*, arXiv:2408.01416.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Workshop on Reasoning and Planning for Large Language Models*.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

OpenAI. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Letitia Parcalabescu and Anette Frank. 2024. On measuring faithfulness or self-consistency of natural language explanations. *Preprint*, arXiv:2311.07466.

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

Juanhe (TJ) Tan. 2023. Causal abstraction for chain-of-thought reasoning in arithmetic word problems. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 155–168, Singapore. Association for Computational Linguistics.

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. In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965. Curran Associates, Inc.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. 2025. Understanding reasoning in thinking language models via steering vectors. In *Workshop on Reasoning and Planning for Large Language Models*.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small. *arXiv e-prints*, arXiv:2211.00593.

Yiqun Wang, Sile Hu, Yonggang Zhang, Xiang Tian, Xuesong Liu, Yaowu Chen, Xu Shen, and Jieping Ye. 2024. How large language models implement chain-of-thought?

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval head mechanistically explains long-context factuality. *Preprint*, arXiv:2404.15574.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

A Implementation Details for Empirical Evaluation

To generate the evaluation results presented in Section 3, we first deploy the three distilled R1 models locally, while utilizing the Azure-hosted API endpoint for the full DeepSeek R1 model. Following the recommendations from DeepSeek (Guo et al., 2025), we append the instruction *"Please reason step by step, and put your final answer within \boxed{}."* to all math-related queries (i.e., the MATH-500 dataset) and set the temperature to 0.6 for all R1 variants. Zero-shot prompting is employed for both the MATH-500 and WildBench datasets. Due to resource limitations, we cap the maximum output length at 10k tokens for the distilled models and 32k tokens for the full R1 model.

After collecting the model outputs, we perform post-processing by discarding responses that are incomplete or do not conform to the required format. For example, Table 2 presents the detailed sample counts after each filtering step when evaluating on the MATH-500 dataset. Specifically, "+Finished" indicates the number of samples that complete within the token limit, "+ThinkFormat" refers to the additional filtering step that ensures the response contains the correct `<think>` tag, "+AnswerFormat" verifies that the final answer is wrapped in "`\boxed{}`", and "+SameId" ensures that the same set of samples is used for both the reasoning and non-reasoning settings.

To examine how model randomness affects format conformance, we perform two runs with identical settings for the three distilled R1 models on the MATH-500 dataset and report the sample counts after the first and second attempts as "once" and "twice" in Table 2. As shown, a second attempt successfully recovers additional samples, particularly for Llama-8B with reasoning, where the number of valid samples increases by approximately 10%.

The same data filtering procedure is applied to the WildBench dataset, with the exception that math-related queries are excluded due to the lack of precise ground-truth answers. Furthermore, to mitigate variance caused by limited data in certain sub-task types, we aggregate the WildBench task domains into four broader categories (Lin et al., 2025) using the following mapping:

- "Coding & Analysis": "Coding & Debugging", "Data Analysis"
- "Creative Tasks": "Brainstorming", "Creative Writing", "Editing", "Role Playing"
- "Information Seeking": "Advice Seeking", "Information Seeking"
- "Reasoning & Planning": "Reasoning", "Planning"

The resulting number of valid samples (after the first model run) used in our evaluation is shown in Figure 1.

For model output evaluation, we follow the methodology provided with the MATH-500 and WildBench datasets. Specifically, MATH-500 is evaluated using rule-based answer extraction and matching, whereas WildBench relies on an LLM judge ("GPT-4o-20240513" in our case). Statistical significance of the difference between the with- and without-reasoning conditions is assessed using a paired t-test.

Finally, to evaluate the effect of temperature, we experimented with greedy decoding (i.e., temperature = 0) and observed results that were largely consistent with those in Figure 1. For example, Table 3 reports the accuracy results on the MATH-500 dataset under this decoding strategy.

B Attention Analysis for WildBench

Here we present attention analysis results on the WildBench dataset, complementing the corresponding results on the MATH-500 dataset discussed in the main text. Segment-level attention patterns are shown in Figure 9, while detailed attention head-level results are provided in Figure 10. The overall trends are consistent with those observed in MATH-500, with two notable differences. First, the aggregated attention from answer to reasoning segments (blue box-plots) exhibits greater variation in WildBench, likely due to its broader domain diversity and corresponding variability in reasoning patterns. Second, the top ten attention heads focusing on reasoning (red boxes in Figure 10) differ slightly from those in MATH-500. Eight of these heads overlap with the top heads in MATH-500, while the remaining two, though not in the top ten for MATH-500, still exhibit relatively high answer-to-reasoning attention in that dataset.

C Attention Patterns for Selected Attention Heads in Early Model Layers

In Figure 11, we present attention patterns from three selected attention heads in the early layers of the three distilled R1 models. The sample cases are identical to those shown in Figure 4. A comparison between Figure 11 and Figure 4 reveals that

Setting	Num	+Finished (once / twice)	+ThinkFormat (once / twice)	+AnswerFormat (once / twice)	+SameId (once)
R1-Llama-8B-withR	500	412 / 448	379 / 433	373 / 429	355
R1-Llama-8B-withoutR	500	490 / 498	489 / 498	465 / 490	355
R1-Qwen-7B-withR	500	470 / 483	470 / 483	466 / 479	457
R1-Qwen-7B-withoutR	500	496 / 499	490 / 497	483 / 492	457
R1-Qwen-1.5B-withR	500	421 / 454	421 / 454	415 / 448	400
R1-Qwen-1.5B-withoutR	500	483 / 494	474 / 493	459 / 484	400
R1-Full-withR	500	500 / -	499 / -	493 / -	490
R1-Full-withoutR	500	500 / -	499 / -	491 / -	490

Table 2: Step-by-step sample counts after applying the data filtering process on the MATH-500 dataset. "+Finished" indicates the number of samples that complete within the token limit, "+ThinkFormat" enforces the presence of the `<think>` tag, "+AnswerFormat" ensures that the final answer is enclosed within "`\boxed{}`", and "+SameId" aligns the sample set across reasoning and non-reasoning conditions. Results are reported after two independent runs ("once" and "twice") to illustrate the effect of model randomness on format conformance.

Model Type	Accuracy (withR / withoutR)	p-value
R1-Llama-8B	94.1% / 77.5%	$p < 0.001$
R1-Qwen-7B	95.4% / 84.8%	$p < 0.001$
R1-Qwen-1.5B	95.6% / 86.8%	$p < 0.001$

Table 3: Greedy decoding results for MATH-500.

although these early-layer heads also assign high attention from answer tokens to reasoning tokens, their attention maps exhibit no discernible structure, instead showing a near-uniform distribution. This suggests that such patterns are not indicative of genuine reasoning focus, but rather reflect artifacts related to the token length.

D Implementation Details for Obtaining Induction and Retrieval Heads

The induction heads for the three distilled R1 models are identified using the `detect_head()` function from the `transformer_lens` Python library (Nanda and Bloom, 2022). Sample prompts include *"one two three one two three one two three"*, *"1 2 3 4 5 1 2 3 4 1 2 3 1 2 3 4 5 6 7"*, and *"green ideas sleep furiously; green ideas don't sleep furiously"*. Default settings are used, and alternative configurations were also tested, yielding negligible differences in output.

For the retrieval heads, we adopt the open-source implementation provided with (Wu et al., 2024), with minor modifications to support the three distilled R1 models. We use the default configuration, setting the detection length to 5000 (i.e., `-e 5000`) to accommodate our GPU constraints.

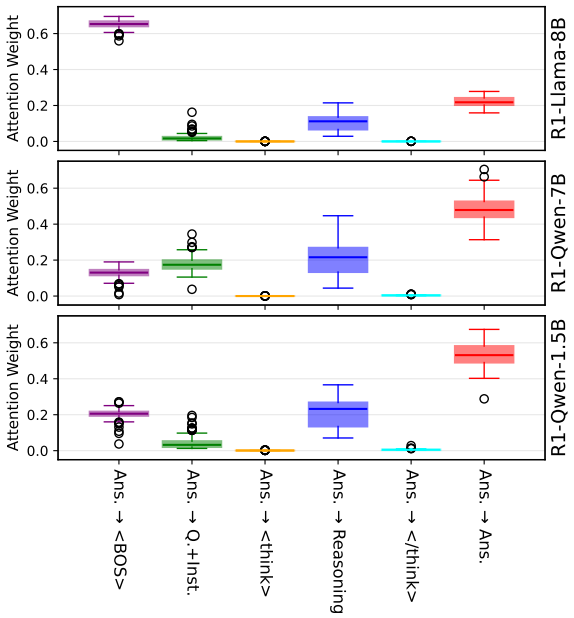


Figure 9: Decomposition of average attention weights from answer tokens to different prompt segments across three models (R1-Llama-8B, R1-Qwen-7B, R1-Qwen-1.5B) using the WildBench dataset.

E Additional Case Details for Debugging Reasoning Failures with RFHs

Figure 5 in the main text visualizes attention weights for a limited set of tokens preceding the error phrase "*C is 0.*" Here, we extend this analysis by visualizing attention over *all* reasoning tokens in the *Reasoning* segment (Figure 12) and by introducing an additional setting that averages over the top 10 Reasoning-Focus Heads (RFHs) identified in R1-Qwen-1.5B (Figure 3). In these visualizations, the top, middle, and bottom panels correspond to (i) the all-head-average view, (ii) the average over the top-10 RFHs, and (iii) the view for the specific

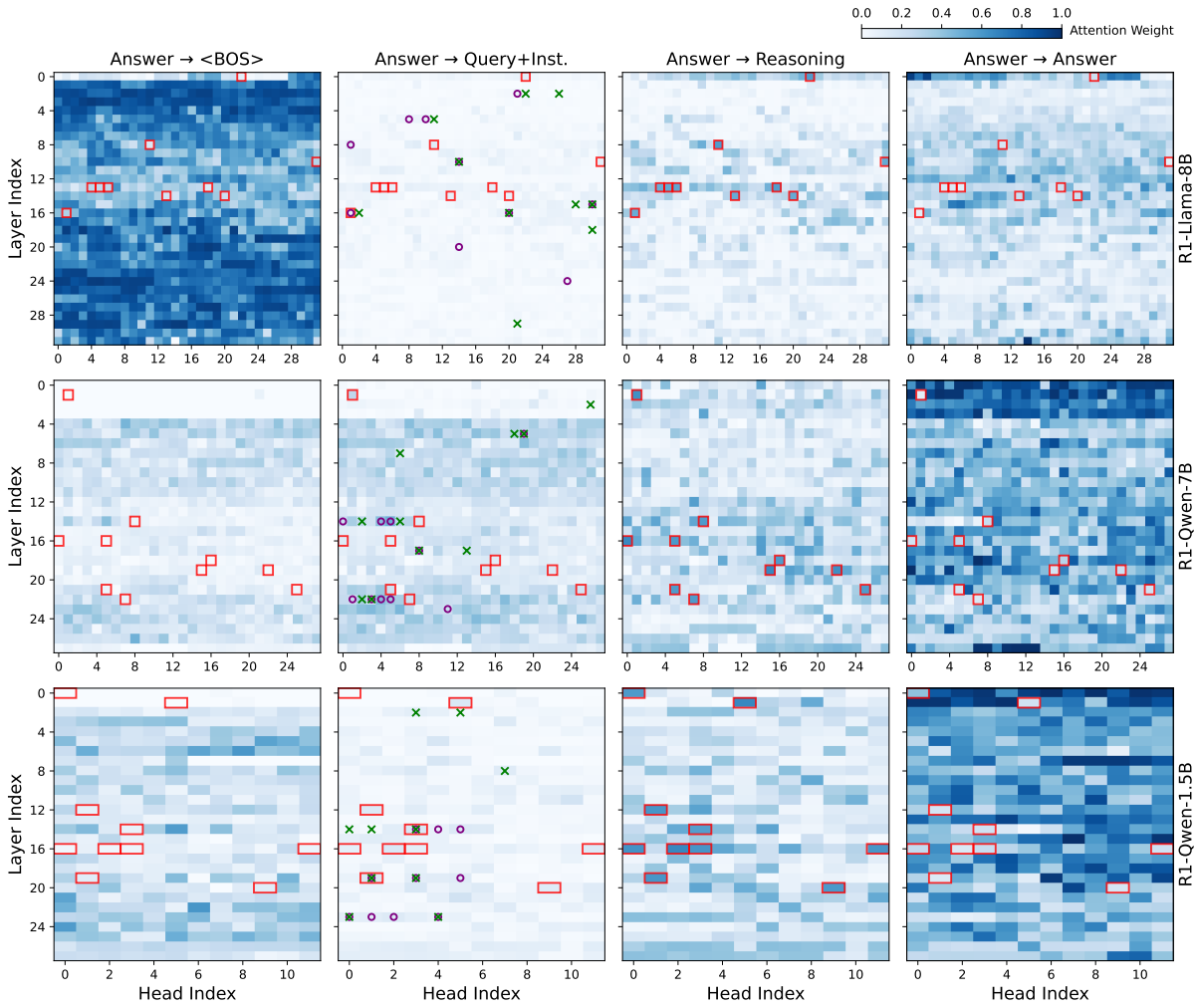


Figure 10: Decomposition of answer attention to different prompt segments across layers and heads for three models (R1-Llama-8B, R1-Qwen-7B, and R1-Qwen-1.5B) using the WildBench dataset. Heatmaps show the attention weights from the *Answer* segment to four prompt segments: <BOS>, *Query+Instruction*, *Reasoning*, and *Answer*. Red boxes highlight the top 10 attention heads with the highest weights for *Answer* → *Reasoning*. Additionally, in the second column the top 10 retrieval and induction heads are annotated with "○" and "×", respectively.

RFH "L16.H2," respectively. All visualizations in Figures 5 and 12 are obtained by utilizing the `circuitsvis` package (Cooney and Nanda, 2023).

Comparing these views highlights the value of RFHs for interpretability. While the head-average view primarily attends to local context, the RFH views reveal that the model strongly links the error phrase "*C is 0*" to the statement "*But* $0r^3$ *is just 0, so we can ignore that.*", indicating that the model confuses the vanishing cubic coefficient with the constant term. Interestingly, the RFH view also attends to the correct constant term coefficient "-24", suggesting that the relevant information is available but is ultimately misprocessed. This observation hints that the subsequent feedforward module in the transformer block may play a critical role in propagating or distorting the attended information, potentially leading to the incorrect conclusion that $C = 0$. Exploring how these downstream modules transform attention-derived signals presents an important direction for future work.

F Data Curation for Contextual Object Comparison Scenario

Table 4 presents the prompt used to generate sample query pairs for the Contextual Object Comparison scenario described in the main text. The prompt enforces two main constraints: i) candidate answers 'A' and 'B' must be single-token words or numbers. This simplifies evaluation by focusing on the prediction of a single token and facilitates the computation of logit differences between the two candidates. ii) the query domains are designed to be diverse. Approximately 30% of query

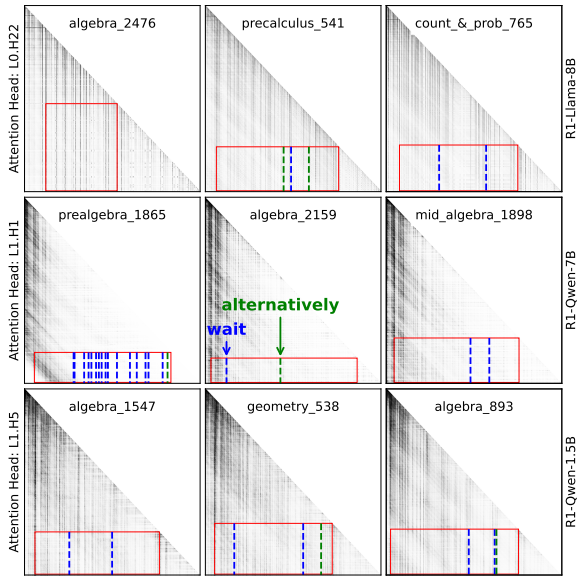


Figure 11: Attention patterns from selected attention heads in the early layers of the three distilled R1 models on sample cases from the MATH-500 dataset. The horizontal and vertical axes represent key and query token indices, respectively. Labels on the left y-axis denote the selected reasoning-focused heads (e.g., *L0.H22"* refers to head 22 in Layer 0). The *Answer* $\rightarrow$ *Reasoning* region is highlighted with red boxes. Vertical lines indicate the positions of the tokens "wait" and "alternatively".

pairs involve numerical answers, 10% are binary choice queries, and the remainder involve arbitrary one-token words. This distribution mitigates bias arising from varying types of candidate objects.

We begin by generating 200 query pairs using OpenAI o1. To ensure that the candidate answers consist of a single token, we apply the tokenizers of the three R1 distilled models and retain only those pairs that meet this criterion, resulting in 103 valid query pairs. Each model then generates a response for these pairs, with the appended instruction *"Please reason step by step (but not overthinking), and put your final answer within \boxed"*. We also constrain the maximum output length to 3,000 tokens to fit within our computational limits. After generation, we parse the responses and discard those that exceed the token budget. Ultimately, we obtain 59 query pairs for R1-Llama-8B, 67 for R1-Qwen-7B, and 22 for R1-Qwen-1.5B. The smaller number for R1-Qwen-1.5B arises from its longer reasoning outputs, which often exceed the token limit. Increasing the budget to 5,000 tokens did not resolve this issue.

G Alignment Procedure for Clean and Corrupted Prompts

To align the clean and corrupted prompts, we first standardize the concluding phrases in the *Answer* and *Reasoning* segments. This involves removing existing variations and replacing them with a consistent format. For the *Answer* segment, we adopt the consistent concluding phrase *"Thus, the {comparator} {condition} is \boxed{"*, where the first and second placeholders are replaced with the corresponding comparator and condition for each sample, respectively. For the *Reasoning* segment, we use two different concluding phrases, illustrated in Figure 6, which also serve as probing phrases in the activation patching experiments.

After standardization, we equalize the token lengths of clean and corrupted prompts through *segment-wise* token padding. Left padding is applied within each segment using the model's default padding token. This alignment may alter the model's original output, potentially changing the predicted answer token. To control for this, we discard samples where the absolute logit difference exceeds 5 after alignment.

H Definition of Normalized Logit Difference

We first define the logit difference LD for each prompt as follows,

$$LD = logit(A) - logit(B), \tag{1}$$

where $logit(A)$ is the logit value for the candidate answer token 'A'. With this, we then define the Normalized Logit Difference (NLD) after patching as:

$$NLD(LD) = \frac{LD - LD_{corrupted}}{LD_{clean} - LD_{corrupted}}, \tag{2}$$

where LD_{clean} and $LD_{corrupted}$ are the logit difference for the original clean and corrupted prompts before patching, respectively, and LD is the logit difference of the corrupted prompt after patching.

You are an experienced query data generator. I want you to generate 200 query pairs.

On the query format:
Each query pair ('Query-a' and 'Query-b') should have the following consistent format:
- Query-a: Considering {condition_a}, which is {comparator}: {A} or {B}?
- Query-b: Considering {condition_b}, which is {comparator}: {A} or {B}?

Instructions for query pair generation:
- {condition_a} and {condition_b} are two distinct conditions so that the answer for 'Query-a' should always be 'A' while the answer for 'Query-b' is always 'B'.
- {condition_a} and {condition_b} should have the same token length. By token length, I mean the number of tokens if passing {condition_a/b} to an LLM tokenizer.
- {comparator} can be any comparison-related word, e.g., "greater".
- {comparator} should be the same for both queries in the query pair.
- Candidate answers of {A} and {B} need to be one-token word or number.
- If candidate answers of {A} and {B} are words, they MUST be simple short 1-token words. For example, 'cheetah' is not allowed for the candidate answers, as it is split into 3 tokens in most LLMs.
- If candidate answers of {A} and {B} are numbers, they MUST be single-digit numbers (i.e., < 10).
- For diversity, I want the total query pairs to cover diverse domains. Avoid to simply replace the candidate answers without major changes on the corresponding conditions.
- About 30% of query pairs should have the candidate answers of {A} and {B} to be single-digit numbers (i.e., < 10).
- About 10% of query pairs should be binary choice queries, i.e., {A} and {B} are 'yes' and 'no'.

On the output format:
- Output should be organized as a table form.
- The output table should have columns of "query_a", "answer_a", "query_b", "answer_b", "domain", "explanation".
- "domain" field provides the query domain.
- "explanation" provides a brief explanation of why 'query_a/b' should have distinct answers of 'answer_a' and 'answer_b'.

Table 4: Prompt used for generating query pairs in the Contextual Object Comparison scenario.

Figure 12: Visualization of attention weights across the entire *Reasoning* segment for three settings: (top) average over all heads, (middle) average over the top 10 Reasoning-Focus Heads (RFHs) identified in R1-Qwen-1.5B, and (bottom) the specific RFH "L16.H2". While the head-average view primarily focuses on local tokens, the RFH views highlight strong attention to the phrase "*But* $0r^3$ *is just 0, so we can ignore that.*", revealing that the model confuses the vanishing cubic term with the constant term when generating "*C is 0*". Interestingly, the RFH view also shows strong attention to the correct constant term coefficient "-24," suggesting that the necessary information is present but may be misprocessed by the downstream feedforward module, ultimately leading to the incorrect conclusion that $C = 0$.