MISLED BY THE FEW: A MECHANISTIC ANALYSIS OF DEMONSTRATION CONFLICTS IN IN-CONTEXT RULE INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

In-context learning enables large language models to perform novel tasks through few-shot demonstrations, including in real deployments where models must infer underlying rules from examples alone. However, demonstrations can naturally contain conflicting examples, making this capability vulnerable. Thus, understanding the dynamics under such conflicts is essential for reliable ICL deployment. In this work, we investigate this question on tasks requiring genuine demonstration reliance. We find that models suffer substantial performance degradation from a single demonstration with a corrupted rule. This systematic misleading behavior motivates our investigation of how models process conflicting evidence internally. Using linear probes and logit lens analysis, we discover that under corruption, models encode both correct and incorrect rules in intermediate layers but develop prediction confidence only in late layers, revealing a two-phase computational structure. We then identify attention heads for each phase underlying the reasoning failures: Vulnerability Heads in early-to-middle layers exhibit positional attention bias with high sensitivity to corruption, while Susceptible Heads in late layers significantly reduce support for correct predictions when exposed to the corrupted evidence. Targeted ablation validates our findings, with masking a small number of identified heads improving accuracy by up to 11.12% relative to the corrupted baseline.[1]

1 INTRODUCTION

In-context learning (ICL) enables large language models (LLMs) to adapt to new tasks through few-shot demonstrations without parameter updates (Brown et al., 2020). We refer to the ability to infer underlying patterns from demonstrations as in-context rule inference (Dong et al., 2022; Bai et al., 2023; Kossen et al., 2023). It underpins both theoretical studies of ICL on tasks like linear regression and analogical reasoning (Von Oswald et al., 2023; Coda-Forno et al., 2023; Zhang et al., 2024; Fu et al., 2025a), and real deployments where models must learn from few-shot examples alone. Applications include agents learning novel internal APIs with missing or outdated documentation (Patel et al., 2025), annotation of concept guidelines in specialized domains (Fonseca & Cohen, 2024), and machine translation conditioned on custom terminology and glossaries (Moslem et al., 2023).

Noisy demonstrations can provide conflicting evidence about the underlying rule. Prior work examines label correctness, the information supplied by demonstrations, and how model scale affects their use (Yoo et al., 2022; Min et al., 2022; Wei et al., 2023). However, studies of conflict resolution focus largely on text classification and question answering (Li et al., 2023; Xie et al., 2023; Halawi et al., 2023; Yu & Ananiadou, 2024), where models can use parametric knowledge for reliable zero-shot performance (Yang et al., 2022; Jiao et al., 2023). These findings do not directly translate to rule inference, which requires genuine demonstration reliance.

In this work, we first investigate the dynamics of in-context rule inference under conflicting demonstrations. We employ two tasks: Operator Induction (Zong et al., 2024) and Fake Word Inference (Aerni et al., 2025), where models perform both tasks well with ICL but can only achieve chance-level performance without demonstrations. These tasks isolate learning from mostly correct examples. Their

[1] https://anonymous.4open.science/r/Understanding-ICL-Demo-Conflict-E831.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

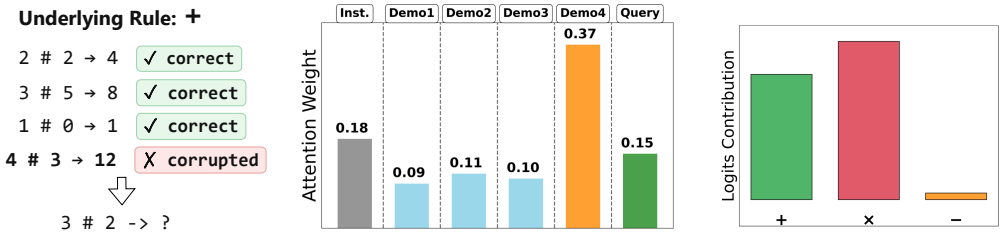


Figure 1: **Left:** The Operator Induction task where models are required to infer the underlying rule (+) from demonstrations, among which we introduce one corrupted example. **Middle:** Vulnerability heads show disproportionate attention allocation; they also show drastic output changes when the heavily-attended position corrupts. **Right:** Susceptible heads shows logit contributions favoring the corrupted operator ($\times$) over the correct one (+) despite minority corruption.

known rule spaces let us change one demonstration while holding the generating rule and remaining evidence fixed, then determine whether the model adopts the corrupted rule. Our experiments reveal that LLMs suffer substantial performance degradation from a single corrupted example among a vast majority of correct ones. We also observe systematic performance variance when corrupting different positions of demonstrations, known as the positional bias (Wang et al., 2024; Cobbina & Zhou, 2025).

During evaluation, we notice a strong signal that models are misled by the injected conflict. When corruption flips models from correct to wrong, the corrupted operator notably accounts for predictions despite appearing in only one position. Given this, we speculate that models encode both rules during the process, yet can fail to resolve the correct rule at the end. This drives us to study how conflicting evidence evolves and affects model behaviour during the internal processing. First, we train linear probes (Alain & Bengio, 2016) to track the emergence of each operator in internal representations, showing that the models encode both correct and misleading rule in intermediate layers. Then, we run logit lens (Nostalgebraist, 2020) to decode model predictions layer by layer, revealing that models develop strong confidence for both rules at the final layers. This temporal separation suggests a natural two-phase hypothesis: conflict creation and conflict resolution.

To test this hypothesis, we identify model components responsible for each phase. For the first phase, since demonstrations are independent and equivalent in our tasks, the positional bias indicates that specific components treat positions unequally. Building on prior work that attributes this bias to attention heads (Wang et al., 2024), we localize **Vulnerability Heads** (Figure 1 middle) that both attend disproportionately to certain positions and show high sensitivity when those positions are corrupted, finding their concentration in early-to-middle layers. For the second phase, we localize **Susceptible Heads** (Figure 1 right) whose preference for the correct over corrupted operator decreases the most when the corruption is introduced, finding their concentration in later layers. We test their causal contribution through targeted ablation, which improves accuracy by up to 11.12% relative to the corrupted baseline. Masking Vulnerability Heads also reduces the susceptibility of late-layer Susceptible Heads.

Our contributions are twofold. First, we establish a framework for studying the dynamics of conflict resolution during in-context rule inference. Second, we provide mechanistic evidence for a two-phase structure of reasoning under corruption, identifying specific model components that exhibit both correlational and causal relevance to conflict creation and conflict resolution.

2 Related Work

2.1 Conflict Resolution in In-Context Learning

Our work studies scenarios where LLMs encounter conflicts during in-context learning. Two types of conflicts emerge in ICL (Xu et al., 2024): context-memory conflicts where contextual information contradicts the model's parametric knowledge, and inter-context conflicts where multiple demonstrations contradict each other. Prior investigations predominantly target context-memory conflicts in tasks such as text classification and textual or visual question answering (Pan et al., 2021; Min et al., 2022; Xie et al., 2023; Halawi et al., 2023; Hua et al., 2025) with a parallel line of work

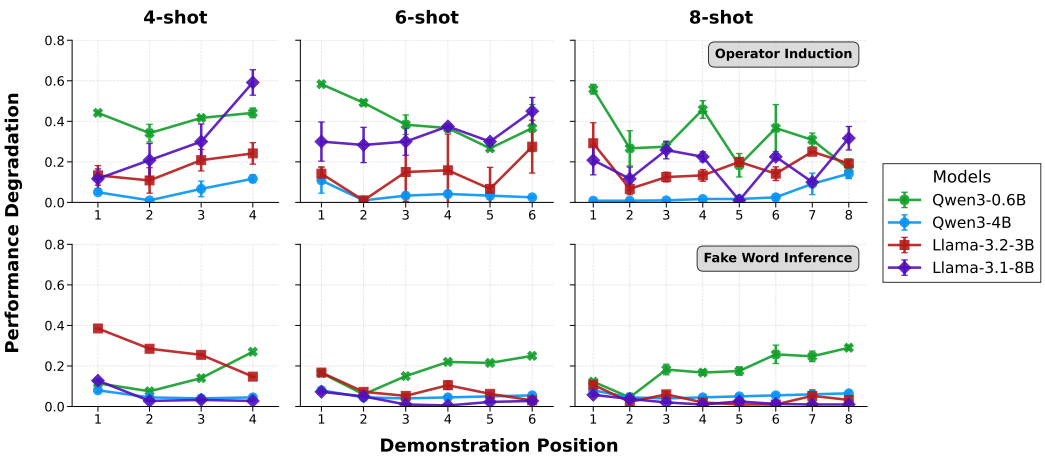


Figure 2: Performance degradation under single-position corruption across different large language models and tasks. Each point represents the decrease in accuracy when corrupting the demonstration at that specific position. Approximate position-wise accuracies are given in Appendix A.3.

on mitigation through adversarial analysis (Wang et al., 2023a) and demonstration selection (Li et al., 2024; Qin et al., 2024; Hu et al., 2024). These tasks share a common property: models can rely on parametric knowledge for reliable zero-shot performance, so demonstrations only refine an already-formed prior. In contrast, in-context rule inference requires genuine demonstration reliance: without demonstrations, models perform at chance (Von Oswald et al., 2023; Coda-Forno et al., 2023; Zong et al., 2024; Patel et al., 2025). This shifts the conflict resolution problem qualitatively: rather than reweighting context against a strong prior, models must arbitrate among demonstrations themselves. How they do so under inter-context conflict remains underexplored, and is the focus of our work.

2.2 Mechanistic interpretability of in-context learning

Our approach to understanding conflict processing mechanisms follows the work of mechanistic interpretability, which seeks to reverse-engineer the computational framework underlying neural network behaviors (Olah et al., 2020; Elhage et al., 2021; Cammarata et al., 2021; Zhang et al., 2025b). These methods enable us to identify when and where models encode specific information, trace how representations evolve across layers, and localize model components responsible for particular behaviors (Yang et al., 2024; Zhang et al., 2025a; Meng et al., 2022).

Recent work applying mechanistic interpretability to ICL has revealed foundational insights about demonstration processing. Wang et al. (2023b) analyzes ICL through an information flow lens, identifying label words as anchors where semantic information aggregates in shallow layers before serving for predictions, and Liu et al. (2023) employs saliency maps to analyze the importance of sentence components during ICL. Notably, a thread of studies focus on functional attention heads for interpreting ICL, following the induction heads Olsson et al. (2022), which are crucial for pattern completion and copying behaviors in ICL. Studies have identified specific attention heads responsible for various ICL phenomena: mechanisms underlying few-shot learning capabilities (Wang et al., 2022), heads that promote incorrect predictions under adversarial contexts (Halawi et al., 2023), and components that contribute to positional bias (Han et al., 2024). This body of work establishes attention heads as primary computational substrates for ICL processing, providing the foundation for our component-level analysis of conflict creation and conflict resolution.

3 Dynamics of in-context rule inference under conflicting demonstrations

To observe how models behave under conflicts for in-context rule inference, we establish an intervention framework that introduces targeted conflicts at specific demonstration positions,

3.1 CORRUPTION-BASED INTERVENTION FRAMEWORK

Task setup. Without loss of generality, we formulate the task as where language models are instructed to infer latent rules from demonstrations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{k}$ and apply the learned rule to predict $\hat{y}_q = r^*(x_q)$ for query x_q. Clean demonstrations represent unanimous evidence for the underlying rule $r^* \in \mathcal{R}$, where $\mathcal{R}$ is the hypothesis space of possible rules. To introduce conflicting evidence among demonstrations, we intervene with a position-specific corruption at demonstration index $p \in \{1, \ldots, k\}$, by replacing the correct output $y_p = r^*(x_p)$ with $y_p^{\text{corrupt}} = r'(x_p)$ where $r' \neq r^*$. The model is then required to do the exact same task under the corrupted demonstration set as:

$$\mathcal{D}_p^{\text{corrupt}} = \{(x_1, y_1), \ldots, (x_p, y_p^{\text{corrupt}}), \ldots, (x_k, y_k)\} \tag{1}$$

This setup carries three fundamental principles. First, we require demonstration reliance, where models are expected to exhibit chance-level performance in zero-shot settings, ensuring their genuine reliance on demonstrations rather than on their parametric knowledge. Second, corruption leaves a clear majority of correct demonstrations. The generating rule fixes the target answer before corruption; the majority provides evidence for recovering that rule. Thirdly, we require the modularity of demonstrations during inference, where each demonstration is independent and provides equally weighted evidence of the underlying rule. Collectively, these three principles provide a valid foundation for studying reasoning dynamics through which LLMs inference rules with contradictions.

Datasets, models, and evaluation. Note that a wide variety of ICL tasks either involve interdependent demonstrations or potentially provide unequal weights per demonstration (Wei et al., 2021), we carefully adopt two tasks that adhere to our principled design criteria: Operator Induction (Zong et al., 2024) where models must infer the underlying mathematical operation from demonstrations to answer the final query, and Fake Word Inference (Aerni et al., 2025) where models learn mappings from synthetic vocabulary to real concepts (e.g., "blimontar glemorax" $\rightarrow$ "red hat"). We conduct experiments across four LLMs, including Qwen3 models (Qwen3-0.6B, Qwen3-4B) (Yang et al., 2025) and Llama3 models (Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct) (Dubey et al., 2024). We interpret model's performance gap between clean and corrupted runs as reflecting the difficulty in autonomously detecting and resolving conflicts. Details of datasets and the evaluation pipeline are included in Appendix A.

3.2 CONSISTENT PERFORMANCE DEGRADATION WITH SINGLE-POSITION CORRUPTIONS

Following our corruption framework, we systematically evaluate model performance with different numbers of demonstrations across all tasks. We first validate models' in-context learning capability for the tasks with no corruptions. Then, to observe performance dynamics under conflicts, we run position-specific corruption by corrupting only a single demonstration at each position within the demonstration sequence and examine the calibrated performance gap position by position, maintaining the majority rule. As shown in Table 2 at Appendix A, all models demonstrate reliable in-context performance with clean demonstrations. However, as shown in Figure 2, LLMs exhibit consistent and substantial performance degradation under conflicts. The degradation can reach as high as 58 percentage points for Operator Induction, with an average degradation of 16 percentage points across all cases. This degradation persists across model scales and architectures, and notably, performance degradation persists even as the number of demonstrations increases to 8, where correct samples maintain a dominant 7:1 advantage over conflicting ones. This pattern indicates that models are significantly obfuscated by conflicting, noisy information during in-context rule inference.

Models heavily adopt corrupted rules. Among Operator Induction predictions that change from correct to incorrect under corruption, 81% on Qwen3-4B and 71% on Llama-3.2-3B follow the corrupted operator, despite its appearing in only one demonstration. This systematic adoption motivates our analysis of how models encode and resolve conflicting rules.

The same behavior persists in Qwen3-32B. With 4-shot demonstrations, it achieves 100% clean accuracy on both tasks, yet a single corruption reduces Operator Induction accuracy by up to 44.4 percentage points; 76.3% of its errors follow the corrupted operator (Appendix A.4).

Positional bias emerges from corruption patterns. Moreover, our experimental design reveals notable positional bias (Hsieh et al., 2024; Cobbina & Zhou, 2025) as shown in Figure 2, where identical corruptions produce different performance impacts depending on their location within the demonstration sequence. This observed bias is particularly instructive in our framework, given the independence and equivalence nature of demonstrations. Under ideal conflict resolution, models should exhibit uniform vulnerability to corruption across all positions, yielding near-zero position bias. The authentic nature of this position bias, uncoupled from confounds of demonstration-specific and parametric knowledge reliance, provides a clean signal for our analysis.

4 Internal Mechanisms of Rule Inference under Conflicting Demonstrations

In the previous section, we present evaluation experiments that reveal clear LLM performance degradation under corruption. However, these evaluations provide little insight into when and how LLMs internally represent and resolve conflicting demonstrations during in-context learning. In this section, we aim to understand these questions by analyzing model internal representations.

4.1 LLMs encode underlying rules in intermediate layers

We use linear probes (Alain & Bengio, 2016) to identify where models encode the rule information. To investigate the evolution of rule encoding, we train linear probes to identify each possible underlying rule in the residual stream across all transformer layers. Details of the probe training setup and validation can be found in Appendix B.

We then evaluate model's encoded confidence of detecting the rules from demonstrations, measured by the predictions of linear probes, under two corruption scenarios: no corruption (all demonstrations are correct) and minority corruption (correct demonstrations outnumber corrupted ones). As shown in Figure 3, probing results reveal two patterns. First, when no corruption information is demonstrated at all, the model shows consistent evidence encoding with high confidence for the correct rule and low confidence for corrupted rules across layers. On the other hand, as corrupted demonstrations are introduced, probe confidence for corrupted rules rises significantly above baseline, indicating that models actively identify and encode the corrupting rule in the process. Notably, internal representations of both correct and incorrect rules emerge most prominently in early-to-middle layers, with confidence levels becoming more stable in later layers. This suggests that models encode multiple competing rules simultaneously during the intermediate processing stages, potentially creating an internal representational conflict that needs to be resolved to produce coherent final predictions.

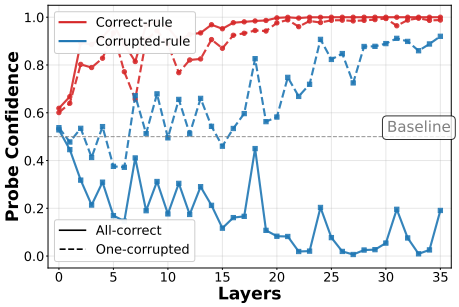


Figure 3: Linear probe confidence across model layers under different corruption scenarios. The Correct Probe detects model's encoding of ground-truth operator while the Corrupted Probe detects the corrupted rules.

4.2 LLMs resolve conflicting evidence in late layers

While probes reveal decodable information, they do not establish when models use it for predictions (Elazar et al., 2021). The logit lens complements these probes by measuring rule probabilities under the model's fixed output projection (Nostalgebraist, 2020). We prompt models to identify the underlying rule rather than compute the final answer. Low probabilities at intermediate layers do not imply that rule information is absent; it may not yet align with the output projection.

For consistency, we collect logit lens outputs under the same two corruption scenarios as linear probes. First, we notice a sharp transition in prediction probability occurring at late layers, where the model suddenly develops strong confidence for specific rules. In contrast, we observe that the model empirically exhibits minuscule prediction confidence across both correct and corrupted rules

in early layers, and thus we omit those layers in Figure 4. Secondly, similar to probing results, we observe the simultaneous emergence of confidence for both correct and corrupted rules under the only-one-corruption setup, further indicating an existing competition between conflicting rules.

4.3 A TWO-PHASE HYPOTHESIS OF RULE INFERENCE UNDER CORRUPTION

Synthesizing the linear probe and logit lens findings, we observe a temporal separation between rule encoding (early-middle layers) and prediction formation (late layers). This temporal structure prompts a hypothesis that model's reasoning under in-context corruptions involves two complementary computational phases, which we term *conflict creation* and *conflict resolution*.

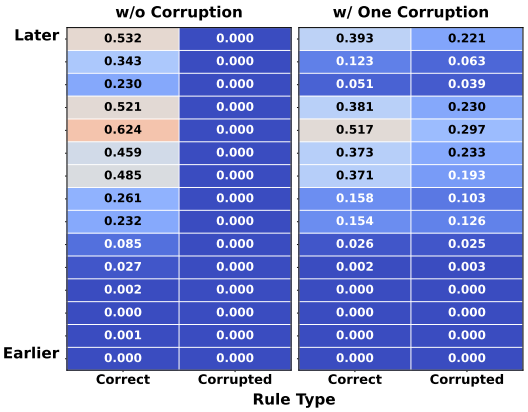


Figure 4: Logit lens predictions across corruption scenarios. Rows show different layers, columns represent correct and corrupted rules, and values indicate prediction probability for each rule decoded from layer-wise residual streams.

We characterize this two-stage hypothesis in more detail. In the first phase, certain model components create systematic weak points by encoding corrupted information, making the system vulnerable to conflicts. In the second phase, the conflict resolution process itself can fail when components are overly susceptible to minority corrupted evidence, leading to incorrect final predictions despite majority support for the correct rule. This framework predicts that early-layer heads should exhibit high sensitivity to positional corruption (creating vulnerabilities), while late-layer heads should demonstrate susceptibility patterns that undermine robust majority-based decisions (resolution failures). We test this hypothesis by identifying components associated with each failure mode and examining their causal contributions to reasoning under corruption.

4.3.1 LOCATING CONFLICT CREATION

We begin by analyzing the first phase of our hypothesis: how certain model components create systematic vulnerabilities under corruption. Under ideal reasoning, each demonstration should contribute equally to rule inference due to the demonstration-dependent nature and equivalent complexity across demonstrations, yielding uniform positional effects.

However, we observe systematic positional variances in performance degradation as shown in Figure 2. This positional bias is not new to research, with prior studies attributing it mostly to the disproportionate integration from demonstrations through the attention mechanism (Han et al., 2024; Peysakhovich & Lerer, 2023; Chen et al., 2023). Building on prior work, we characterize two measurable attributes of attention heads that potentially contribute to the position-variant degradation under corruption: (1) disproportionate attention allocation per position, and (2) high output sensitivity when heavily-attended positions are corrupted. In other words, these heads are prone to create systematic vulnerabilities when contradictory information appears at their heavily-attended positions. To quantify them, we follow the ICL prompt segmentation from (Cho et al., 2024), and focus our analysis on the final query forerunner token, which is identified critical for encoding input text representations during in-context learning.

Measuring positional attention allocation. For each attention head (l, h) at layer l and head h, we extract the attention score $\mathbf{A}_{l,h} \in \mathbb{R}^{S\times S}$ where S represents sequence length. Following the demonstration-query segmentation (Cho et al., 2024), we compute the attention allocation from the query forerunner token position $\mathbf{F}$ to each demonstration position $p \in \{1, 2, \ldots, k\}$:

$$A_{l,h}^{(p)} = \frac{1}{|D_p|} \sum_{t\in D_p} \mathbf{A}_{l,h}[\mathbf{F}, t] \tag{2}$$

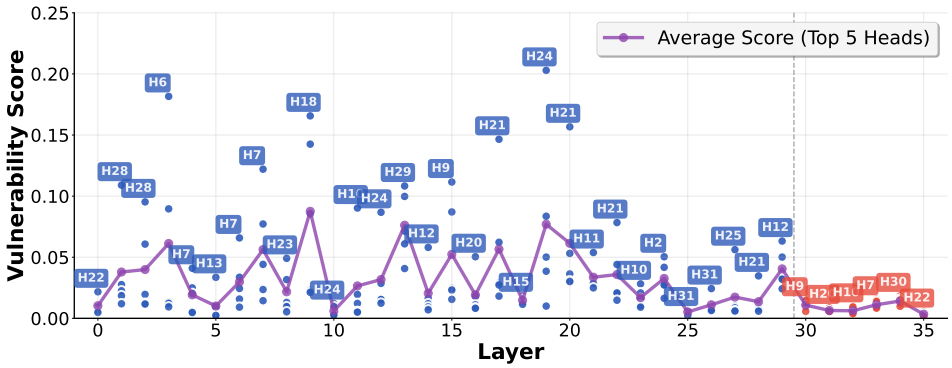


Figure 5: Distribution of Vulnerability Heads across model layers. The purple line shows the average vulnerability score (A × S) among the top 5 heads per layer. Individual heads are shown as dots.

where D_p denotes the set of token positions belonging to demonstration p, and $|D_p|$ represents the number of tokens in that demonstration. This averaging yields per-demonstration attention scores that capture how much each head attends to specific positions during query processing.

Measuring positional sensitivity. For each attention head (l, h) and demonstration position p, we compute sensitivity as the normalized change in head outputs when that position is corrupted. Specifically, we first extract the head's output $\mathbf{o}_{l,h}^{(\mathbf{F})}(\text{clean})$ at the query forerunner position $\mathbf{F}$ under all-correct conditions, then corrupt only position p with incorrect rule variants $r \in \mathcal{R}_{\text{wrong}}$ while keeping all other positions unchanged. We measure the output deviation $\mathbf{o}_{l,h}^{(\mathbf{F})}(\text{corrupted})$ for each corruption and compute the average normalized change. The sensitivity score is formulated as:

$$S_{l,h}^{(p)} = \frac{\left\| \mathbf{o}_{l,h}^{(\mathbf{F})}(\text{corrupt}) - \mathbf{o}_{l,h}^{(\mathbf{F})}(\text{clean}) \right\|_2}{\left\| \mathbf{o}_{l,h}^{(\mathbf{F})}(\text{clean}) \right\|_2 + \epsilon} \tag{3}$$

High sensitivity indicates heads whose outputs fluctuate substantially with conflicting demonstrations. A control for small clean-output norms is reported in Appendix E.2. Intuitively, for corrupted position p, we are interested in localizing those attention heads that attend heavily to p while being extremely sensitive in outputs to the single corruption at p. Before combining these attributes, we first assess whether they measure the same or distinct facets of attention head behaviour. To this end, we quantify their dependence using Pearson and Spearman correlations, with results shown in Appendix Figure 10. We observe weak associations across heads, indicating that our two characterizations are largely complementary rather than redundant. Thus, we multiply them as the *Vulnerability Score* and serve it as a filter for salient heads. Figure 5 shows the distribution of Vulnerability Heads for Qwen3-4B, with blue indicating early-to-middle layer heads and red indicating final-layer ones. We observe a clear stratification with vulnerability behavior concentrated in early layers, supporting the first phase of our two-stage framework. However, we point out that this score only reflects correlational patterns rather than causal impacts.

To validate the causal role of these heads in creating vulnerabilities, we further conduct targeted ablation experiments by masking identified vulnerability heads during inference. We run three rollouts for random corruption rules and compare against random head ablation. As shown in Table 1, ablating top vulnerability heads consistently improves performance under corruption, achieving up to 11.12% relative performance improvement. To contextualize the magnitude, we also measure the proportion of corruption-induced degradation recovered by the best ablation across head counts. Recovery ranges from moderate to substantial across models, with stronger cases reaching above 30% on Qwen3-4B and 40% on Llama-3.2-3B. This downstream effect validates our identified heads and the causal contribution of these heads to failures under demonstration corruption.

4.3.2 Locating Conflict Resolution

We next investigate the second phase of our hypothesis: how certain model components cause the failure of conflict resolution. Under the corruption framework, robust conflict resolution requires

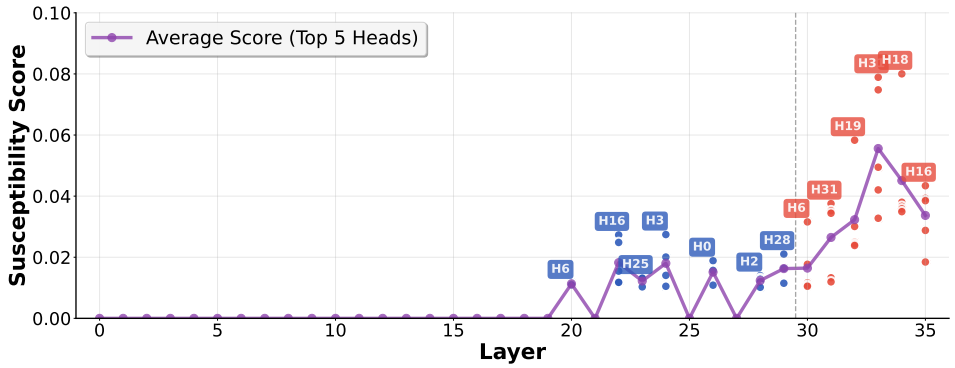


Figure 6: Distribution of susceptible heads across model layers. The purple line shows average susceptibility scores among top 5 heads per layer. Individual heads shown as dots.

maintaining support for correct predictions even when minority corrupted evidence appears, and we characterize the susceptibility to corruption as the opposite pattern of being easily swayed. Intuitively, and also established by previous studies, attention heads are the primary substrate for mediating how models selectively incorporate information from different demonstrations (Olsson et al., 2022; Halawi et al., 2023; Han et al., 2024). Thus, we focus our analysis on identifying attention heads that exhibit this susceptibility pattern.

Measuring susceptibility via logit attribution. We employ logit attribution (Wang et al., 2022) to quantify each attention head's contribution to final predictions under the same two corruption scenarios as above. For each query, we construct two contrastive contexts: a clean context where all demonstrations are correct, and a minority corruption context where only one demonstration is corrupted. We extract each attention head's output at the query forerunner position and compute its attribution to the correct rule token through the unembedding matrix. For head (l, h), we denote the logit attribution gap between the correct and corrupted rule tokens as $\Delta_{l,h}^{\text{clean}}$ and $\Delta_{l,h}^{\text{corrupt}}$ under the two contexts respectively, and define the *Susceptibility Score* as their difference:

$$\text{Susc}_{l,h} = \Delta_{l,h}^{\text{clean}} - \Delta_{l,h}^{\text{corrupt}}. \tag{4}$$

A high positive score indicates that the head's preference for the correct rule drops substantially when a single corrupted demonstration is introduced, suggesting problematic susceptibility rather than robust conflict resolution. We term heads with high scores *Susceptible Heads*. Figure 6 from Qwen3-4B reveals that the most salient susceptible heads are predominantly concentrated in late layers. Across all four models, the top-10 Susceptible Heads have a later mean normalized layer than the top-10 Vulnerability Heads (Appendix D.1). We test causal influence through targeted ablation (Table 1). Susceptible Head ablation yields relative accuracy gains of up to 10.33%. Across both head families and all tested head counts, targeted ablation improves 23/24 configurations, averaging 3.87%, while random ablation averages -2.52%. Targeted ablation outperforms random ablation in 20/24 configurations ($p < 0.05$; Appendix E.1).

5 Discussion

5.1 The generalizability of identified heads

To evaluate whether identified heads capture generalizable conflict-resolution mechanisms, we measure the consistency of both Vulnerability and Susceptible Heads across the Operator Induction and Fake Word Inference task. On the first hand, we find that susceptible heads exhibit significant overlap across tasks, with 10 of the top 20 heads from Qwen3-4B shared between both tasks, as visualized in Figure 11 at Appendix E. This observation suggests that late-layer conflict resolution relies on generalizable mechanisms. To verify the generalizability, we conduct a cross-task ablation study: ablating top 10 susceptibility heads identified from Fake Word Inference for Operator Induction. This cross-task ablation improves performance under corruption by 3.80% in average across models, indicating a potential mitigation solution under conflict demonstrations across different rule inference tasks. On the other hand, we observe that vulnerability heads exhibit lower cross-task overlap (4

Table 1: Attention head ablation results showing relative performance improvement (%) on Operator Induction when masking different numbers and types of attention heads. Random ablation consistently degrades performance, while ablating identified Vulnerability and Susceptible heads improves it.

Models	Type of Heads	# of Ablated Heads		
		5	8	10
Qwen3-0.6B	Vulnerability	1.58±1.57	4.91±1.55	8.49±1.58
	Susceptible	10.33±1.48	8.41±1.18	8.21±1.50
	Random	-0.85±0.42	-2.34±0.38	-4.67±0.51
Qwen3-4B	Vulnerability	1.94±0.87	0.88±0.66	1.58±0.90
	Susceptible	1.58±0.90	1.94±0.87	1.94±0.87
	Random	-0.44±0.20	-2.07±0.85	-4.24±1.03
Llama-3.2-3B	Vulnerability	-0.03±1.47	11.12±1.34	8.21±1.38
	Susceptible	0.93±1.46	4.34±1.42	1.33±1.50
	Random	-0.67±0.39	-2.15±0.52	-4.38±0.45
Llama-3.1-8B	Vulnerability	3.89±1.39	4.63±1.39	1.97±1.44
	Susceptible	1.30±1.43	2.00±1.41	1.38±1.49
	Random	-0.93±0.41	-2.56±0.36	-4.92±0.53

of top 20 heads from Qwen3-4B), suggesting that attention allocation and sensitivity to conflicting evidence is more task-specific.

5.2 Empirical synergy between the two phases

We also investigate in the synergy among these components within the model's computational pathway. Specifically, we examine whether ablating vulnerability heads reduces the susceptibility of late-layer heads by measuring changes in their logit contributions, and compare these effects against random head ablations. As shown in Figure 12 at Appendix E, ablating vulnerability heads reduces the susceptibility scores of top susceptible heads (e.g. L30H29 and L31H5 for Qwen3-4B on Operator Induction) significantly more than random ablation. This case study suggests an empirical synergy between intermediate-layer vulnerability creation and late-layer susceptibility, through the mediation of the two types of heads.

6 Conclusion

We study how LLMs infer rules from conflicting demonstrations in two tasks that require learning from examples. Our representation analyses support a two-phase account: models encode competing rules in intermediate layers, while rule preferences become readable through the output projection in late layers. We identify Vulnerability Heads with positional attention bias and high sensitivity to corruption, and Susceptible Heads whose support for correct predictions falls under corruption. The top-10 Susceptible Heads occur later on average than the top-10 Vulnerability Heads in all four models. Targeted ablation improves accuracy in 23 of 24 configurations, with gains of up to 11.12% relative to the corrupted baseline, supporting the causal contribution of the identified heads to failures under corruption. To summarize, the controlled rule spaces let us distinguish adoption of the corrupted rule from other errors while keeping the generating rule and remaining evidence fixed. We combine this behavioral control with representation analysis and targeted ablation to connect performance degradation to specific internal components.

The behavioral failure also persists in Qwen3-32B, extending the finding beyond the four models used for head-level analysis. Further tests in larger models and richer rule-inference settings, including agents learning unfamiliar APIs, would clarify the scope of the identified mechanisms. Such evaluations would test whether positional vulnerability and late-layer susceptibility persist when demonstrations differ in relevance or depend on one another.

ETHICS STATEMENT

This work studies how language models process conflicting in-context demonstrations using controlled arithmetic and synthetic word-mapping tasks. The experiments use publicly available models and do not involve human participants or private personal data.

REPRODUCIBILITY STATEMENT

Appendix A describes task construction, corruption procedures, prompts, decoding settings, answer extraction, and statistical evaluation. Appendices B and C describe probe training and logit lens analysis. Appendix E reports head-selection controls and supplementary ablation results. The codebase is linked in the abstract.

AI USE STATEMENT

Generative AI assisted with writing, reference checks, figures, and formatting. The authors are responsible for the final content.

REFERENCES

Michael Aerni, Joshua Swanson, Kristina Nikolić, and Florian Tramèr. Modal aphasia: Can unified multimodal models describe images from memory? *arXiv preprint arXiv:2510.21842*, 2025.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve circuits. *Distill*, 6(1):e00024–006, 2021.

Yongqiang Chen et al. Positional information matters for invariant in-context learning: A case study of simple function classes. *arXiv preprint arXiv:2311.18194*, 2023.

Hakaze Cho, Mariko Kato, Yoshihiro Sakai, and Naoya Inoue. Revisiting in-context learning inference circuit in large language models. *arXiv preprint arXiv:2410.04468*, 2024.

Kwesi Cobbina and Tianyi Zhou. Where to show demos in your prompt: A positional bias of in-context learning. *arXiv preprint arXiv:2507.22887*, 2025.

Julian Coda-Forno, Marcel Binz, Zeynep Akata, Matt Botvinick, Jane Wang, and Eric Schulz. Meta-in-context learning in large language models. *Advances in Neural Information Processing Systems*, 36:65189–65201, 2023.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits, December 2021. URL `https://transformer-circuits.pub/2021/framework/index.html`. Transformer Circuits series.

Marcio Fonseca and Shay B Cohen. Can large language models follow concept annotation guidelines? a case study on scientific and financial domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8027–8042, 2024.

Shaopeng Fu, Liang Ding, Jingfeng Zhang, and Di Wang. Short-length adversarial training helps llms defend long-length jailbreak attacks: Theoretical and empirical evidence. *arXiv preprint arXiv:2502.04204*, 2025a.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv preprint arXiv:2508.15260*, 2025b.

Danny Halawi, Jean-Stanislas Denain, and Jacob Steinhardt. Overthinking the truth: Understanding how language models process false demonstrations. *arXiv preprint arXiv:2307.09476*, 2023.

Hanzhang Han et al. Unibias: Unveiling and mitigating llm bias through internal attention and ffn manipulation. *arXiv preprint arXiv:2405.20612*, 2024.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, et al. Found in the middle: Calibrating positional attention bias improves long context utilization. *arXiv preprint arXiv:2406.16008*, 2024.

Jingyu Hu, Weiru Liu, and Mengnan Du. Strategic demonstration selection for improved fairness in llm in-context learning. *arXiv preprint arXiv:2408.09757*, 2024.

Tianze Hua, Tian Yun, and Ellie Pavlick. How do vision-language models process conflicting information across modalities? *arXiv preprint arXiv:2507.01790*, 2025.

Difan Jiao, Yilun Liu, Zhenwei Tang, Daniel Matter, Jürgen Pfeffer, and Ashton Anderson. Spin: Sparsifying and integrating internal neurons in large language models for text classification. *arXiv preprint arXiv:2311.15983*, 2023.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*, 2025.

Jannik Kossen, Yarin Gal, and Tom Rainforth. In-context learning learns label relationships but is not conventional learning. *arXiv preprint arXiv:2307.12375*, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Jierui Li, Vipul Raheja, and Dhruv Kumar. Contradoc: understanding self-contradictions in documents with large language models. *arXiv preprint arXiv:2311.09182*, 2023.

Lvxue Li, Jiaqi Chen, Xinyu Lu, Yaojie Lu, Hongyu Lin, Shuheng Zhou, Huijia Zhu, Weiqiang Wang, Zhongyi Liu, Xianpei Han, et al. Debiasing in-context learning by instructing llms how to follow demonstrations. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 7203–7215, 2024.

Fuxiao Liu, Paiheng Xu, Zongxia Li, Yue Feng, and Hyemi Song. Towards understanding in-context learning with contrastive demonstrations and saliency maps. *arXiv preprint arXiv:2307.05052*, 2023.

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Yasmin Moslem, Rejwanul Haque, John Kelleher, and Andy Way. Adaptive machine translation with large language models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pp. 227–237, 2023.

Nostalgebraist. Interpreting gpt: The logit lens. LessWrong post, 2020. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, March 2020. doi: 10.23915/distill.00024.001. URL `https://distill.pub/2020/circuits/zoom-in`. OpenAI.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Liangming Pan, Wenhu Chen, Min-Yen Kan, and William Yang Wang. Attacking open-domain question answering by injecting misinformation. *arXiv preprint arXiv:2110.07803*, 2021.

Bhrij Patel, Ashish Jagmohan, and Aditya Vempaty. Learning api functionality from in-context demonstrations for tool-based agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 18318–18336, 2025.

Alexander Peysakhovich and Adam Lerer. Attention sorting combats recency bias in long context language models, 2023.

Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. In-context learning with iterative demonstration selection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7441–7455, 2024.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Jiongxiao Wang, Zichen Liu, Keun Hee Park, Zhuojun Jiang, Zhaoheng Zheng, Zhuofeng Wu, Muhao Chen, and Chaowei Xiao. Adversarial demonstration attacks on large language models. *arXiv preprint arXiv:2305.14950*, 2023a.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*, 2023b.

Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M Kakade, Hao Peng, and Heng Ji. Eliminating position bias of language models: A mechanistic approach. *arXiv preprint arXiv:2407.01100*, 2024.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in english? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15366–15394, 2024.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*, 2023.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for llms: A survey. *arXiv preprint arXiv:2403.08319*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shu Yang, Shenzhe Zhu, Liang Liu, Lijie Hu, Mengdi Li, and Di Wang. Exploring the personality traits of llms through latent features steering. *arXiv preprint arXiv:2410.10863*, 2024.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 3081–3089, 2022.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. Ground-truth labels matter: A deeper look into input-label demonstrations. *arXiv preprint arXiv:2205.12685*, 2022.

Zeping Yu and Sophia Ananiadou. How do large language models learn in-context? query and key matrices of in-context heads are two towers for metric learning. *arXiv preprint arXiv:2402.02872*, 2024.

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

Jiayi Zhang, Shu Yang, Junchao Wu, Derek F Wong, and Di Wang. Understanding and mitigating political stance cross-topic generalization in large language models. *arXiv preprint arXiv:2508.02360*, 2025a.

Lin Zhang, Wenshuo Dong, Zhuoran Zhang, Shu Yang, Lijie Hu, Ninghao Liu, Pan Zhou, and Di Wang. Eap-gp: Mitigating saturation effect in gradient-based automated circuit identification. *arXiv preprint arXiv:2502.06852*, 2025b.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

Yongshuo Zong, Ondrej Bohdal, and Timothy M Hospedales. Vl-icl bench: The devil in the details of benchmarking multimodal in-context learning. *CoRR*, 2024.

Amir Zur, Atticus Geiger, Ekdeep Singh Lubana, and Eric Bigelow. Are language models aware of the road not taken? token-level uncertainty and hidden state dynamics. *arXiv preprint arXiv:2511.04527*, 2025.

A Details of Corruption-based Intervention Framework

A.1 Corruption-based Experimental Design

Our evaluation protocol systematically examines model robustness to conflicting demonstrations through controlled corruption at specific positions within the demonstration sequence.

A.1.1 Data Scale and Coverage

Taking 4-shot ICL as an example, we evaluate the following:

Operator Induction Tasks.

- Query samples: 60 unique queries per task
- Shots per query: 4 demonstrations
- Corruption positions: 4 positions
- Rollouts per position: 3 independent trials with different random seeds
- Total evaluations: 720 comparisons ($60 \times 4 \times 3$)

Fake Word Inference Task.

- Query samples: 100 unique target pairs (color-object combinations)
- Shots per query: 4 demonstrations
- Corruption positions: 4 positions
- Rollouts per position: 3 independent trials with different random seeds
- Total evaluations: 1,200 comparisons ($100 \times 4 \times 3$)

Note that in our tasks the choice of corrupted rules can be flexible based on the hypothesis rule space. This design yields robust statistical estimates across position-specific corruption patterns. Table 2 validates our task selection criteria for studying demonstration-dependent conflict resolution. All tasks exhibit near-chance 0-shot performance, confirming genuine demonstration reliance. Performance substantially improves with few-shot demonstration, demonstrating reliable ICL capability. This pattern of models' failing without demonstrations yet succeeding with them intrinsically satisfies our principled framework requirement that models must genuinely rely on contextual evidence rather than parametric knowledge.

Table 2: Clean (Uncorrupted) in-context learning performance across tasks and models. Models exhibit reliable ICL capabilities on both Operator Induction and Fake Word Inference.

Task	Model	0-shot	4-shot	6-shot	8-shot
Operator Induction	Qwen3-0.6B	0.35	0.84$\pm$0.03	0.88$\pm$0.01	0.88$\pm$0.01
	Qwen3-4B	0.37	0.99$\pm$0.01	1.00$\pm$0.00	1.00$\pm$0.00
	Llama-3.2-3B	0.42	0.83$\pm$0.05	0.69$\pm$0.08	0.75$\pm$0.02
	Llama-3.1-8B	0.32	0.90$\pm$0.02	0.92$\pm$0.02	0.91$\pm$0.01
Fake Word Inference	Qwen3-0.6B	0.03	0.95$\pm$0.02	0.88$\pm$0.01	0.91$\pm$0.02
	Qwen3-4B	0.03	0.99$\pm$0.01	1.00$\pm$0.00	1.00$\pm$0.00
	Llama-3.2-3B	0.07	0.82$\pm$0.01	0.84$\pm$0.02	0.84$\pm$0.02
	Llama-3.1-8B	0.08	0.93$\pm$0.01	0.93$\pm$0.01	0.93$\pm$0.01

A.1.2 Corruption Mechanism

We implement systematic single-position corruption where exactly one demonstration at position p (e.g., $p \in \{1, 2, 3, 4\}$ for 4-shot scenarios) receives a corrupted rule while maintaining the majority rule principle. For Operator Induction tasks, corruption involves replacing the correct mathematical

operator with an alternative operator (+, -, ×), ensuring the corrupted demonstration exhibits a different underlying rule while maintaining surface-level plausibility. For Fake Word Inference, we first select all demonstrations showing the correct fake word mapping (e.g., all 4 demonstrations mapping the fake color word to the target real color), then substitute exactly one demonstration's real word with an incorrect alternative from the same category (e.g., replacing "purple" with "blue" while keeping the fake word unchanged). This ensures the correct mapping maintains a 3:1 majority over the corrupted alternative, providing dominating evidence for conflict resolution while testing whether models will be obfuscated by the only corrupted evidence.

A.2 Model Response Evaluation Protocol

A.2.1 Inference Setup

We employ the vLLM framework (Kwon et al., 2023) for efficient batched inference on NVIDIA A100 80GB GPUs. For generation, we use greedy decoding (temperature = 0.0) with a maximum output length of 2,048 tokens. For Qwen3 models, we enable the extended thinking mode via the chat template to allow chain-of-thought reasoning.

A.2.2 Prompt Design

Operator Induction. Taking 4-shot in-context rule inference as an example, we employ standard formatting adapted from (Zong et al., 2024). The prompt instructs models to output their final answer after the `####` delimiter. Below is an example of corrupted prompt:

```
The text contains two digit numbers and a ? representing the
mathematical operator. Induce the mathematical operator (addition,
multiplication, minus) according to the results of the in-context
examples and calculate the result. Think step by step, then write
your final answer after ####

Support Set:
8 ? 6 =
Answer: 14

3 ? 5 =
Answer: 8

7 ? 2 =
Answer: 5

9 ? 4 =
Answer: 13

Question:
6 ? 3 = ?

Answer:
```

Fake Word Inference. For the Fake Word Inference task, we present demonstrations mapping fake words to real color-object combinations. Each demonstration follows the format "`[fake_phrase] means [real_phrase]`". In this corrupted example, one demonstration (position 2) shows an incorrect color mapping while the majority maintain the correct mapping. Below is an example of corrupted prompt:

```
You will see examples showing what fake words mean in terms of
real colors and objects. Learn the mapping from fake words to
real words, then answer the question using real words only.
Think step by step, then write your final answer after ####
```

```
Support Set:
zandolex vundelka means purple sheep
zandolex glemorax means blue hat
zandolex plintovar means purple scarf
zandolex thovaline means purple flower

Question:
What color is zandolex frivenmox?

Answer:
```

A.2.3 Answer Extraction and Evaluation

Operator Induction Evaluation. For Operator Induction tasks, we employ a priority-based regex extraction approach to identify the final numerical answer from model responses. The extraction follows this priority order: (1) numbers appearing after the #### delimiter; (2) numbers after explicit answer markers such as "Final answer:" or "Answer:"; (3) numbers immediately following an equals sign; and (4) the last number in the response as a fallback. Correctness is determined by exact match between the extracted integer and the ground truth answer.

Fake Word Inference Evaluation. For Fake Word Inference, we extract the final answer segment after the #### delimiter and perform exact string matching against the ground truth color (e.g., "purple"). This deterministic evaluation ensures consistent assessment of mapping accuracy.

A.2.4 Statistical Design

Majority Rule Verification. Our corruption mechanism maintains the majority rule principle: in 4-shot scenarios with single-position corruption, correct demonstrations maintain at least a 3:1 majority over corrupted ones, providing a theoretically sound basis for expecting successful conflict resolution under ideal conditions. This applies to both Operator Induction tasks (correct operator appears in 3 of 4 demonstrations) and Fake Word Inference (correct mapping appears in 3 of 4 examples).

Paired Experimental Control. Critical to our methodology is the paired nature of evaluations: for each (query, position, rollout) tuple, the clean baseline and corrupted conditions share the same underlying demonstration structure, differing only in the corruption applied at the target position. We use position-specific random seeds to ensure reproducibility across experimental runs. We report performance degradation as the difference between clean baseline accuracy and corrupted accuracy, with statistical significance assessed via paired t-tests across rollouts.

Statistical Significance. For corruption-induced degradation, all 8 model×task combinations are significant at $p < 0.001$. The weakest case is Qwen3-4B on 8-shot Operator Induction with $t = 4.66$ (still at $p < 10^{-5}$), reflecting the small absolute degradation on this already near-saturated setting. For the attention head ablation results in Table 1, improvements of Vulnerability and Susceptible head ablation over random head ablation are significant ($p < 0.05$) in most cases (20/24 entries), with the strongest effects at 8 and 10 ablated heads where significance holds across all models.

A.3 Accuracy under single-position corruption

Table 3 reports approximate accuracies for the 4-, 6-, and 8-shot settings in Figure 2. The index p counts demonstrations in prompt order: $p = 1$ is the first demonstration, and $p = k$ is the last demonstration before the query. In each column, only demonstration p is corrupted; the remaining demonstrations are correct.

We reconstruct corrupted accuracy by subtracting the mean drop plotted in Figure 2 from the corresponding clean mean in Table 2. The clean means are rounded to two decimal places, so the reconstructed accuracies inherit that rounding. All entries are percentages, shown to one decimal place.

Table 3: Approximate accuracy (%) under single-position corruption, reconstructed from the clean means in Table 2 and the plotted mean drops in Figure 2. Clean denotes uncorrupted accuracy; a dash indicates that $p > k$.

			Corrupted demonstration position p							
Model	k	Clean	1	2	3	4	5	6	7	8
Operator Induction										
	4	84.0	39.8	49.8	42.3	39.8	–	–	–	–
Qwen3-0.6B	6	88.0	29.7	38.8	49.7	51.3	61.3	51.3	–	–
	8	88.0	32.2	61.3	60.5	42.2	69.7	51.3	57.2	71.3
	4	99.0	94.0	98.0	92.3	87.3	–	–	–	–
Qwen3-4B	6	100.0	89.2	99.0	96.7	95.8	96.7	97.5	–	–
	8	100.0	99.2	99.2	99.0	98.3	98.3	97.5	90.8	85.8
	4	83.0	69.7	72.2	62.2	58.8	–	–	–	–
Llama-3.2-3B	6	69.0	54.8	68.0	54.0	53.2	62.3	41.5	–	–
	8	75.0	45.8	68.3	62.5	61.7	55.0	60.8	50.0	55.8
	4	90.0	78.3	69.2	60.0	30.8	–	–	–	–
Llama-3.1-8B	6	92.0	62.0	63.7	62.0	54.5	62.0	47.0	–	–
	8	91.0	70.2	79.3	65.2	68.5	90.0	68.5	81.0	59.3
Fake Word Inference										
	4	95.0	83.5	87.5	81.0	68.0	–	–	–	–
Qwen3-0.6B	6	88.0	71.5	82.0	73.0	66.0	66.5	63.0	–	–
	8	91.0	78.8	86.5	72.8	74.3	73.5	65.3	66.3	62.0
	4	99.0	91.0	94.5	95.0	94.5	–	–	–	–
Qwen3-4B	6	100.0	92.0	95.5	96.0	95.5	95.0	94.5	–	–
	8	100.0	92.0	95.5	96.0	95.5	95.0	94.5	94.0	93.5
	4	82.0	43.5	53.5	56.5	67.3	–	–	–	–
Llama-3.2-3B	6	84.0	67.3	76.8	78.8	73.5	77.8	81.0	–	–
	8	84.0	73.3	81.8	78.0	82.0	83.0	83.0	78.8	80.8
	4	93.0	80.3	90.3	89.8	90.3	–	–	–	–
Llama-3.1-8B	6	93.0	85.8	88.0	92.0	92.5	90.8	90.3	–	–
	8	93.0	87.3	89.3	91.0	92.0	90.5	91.8	92.0	92.0

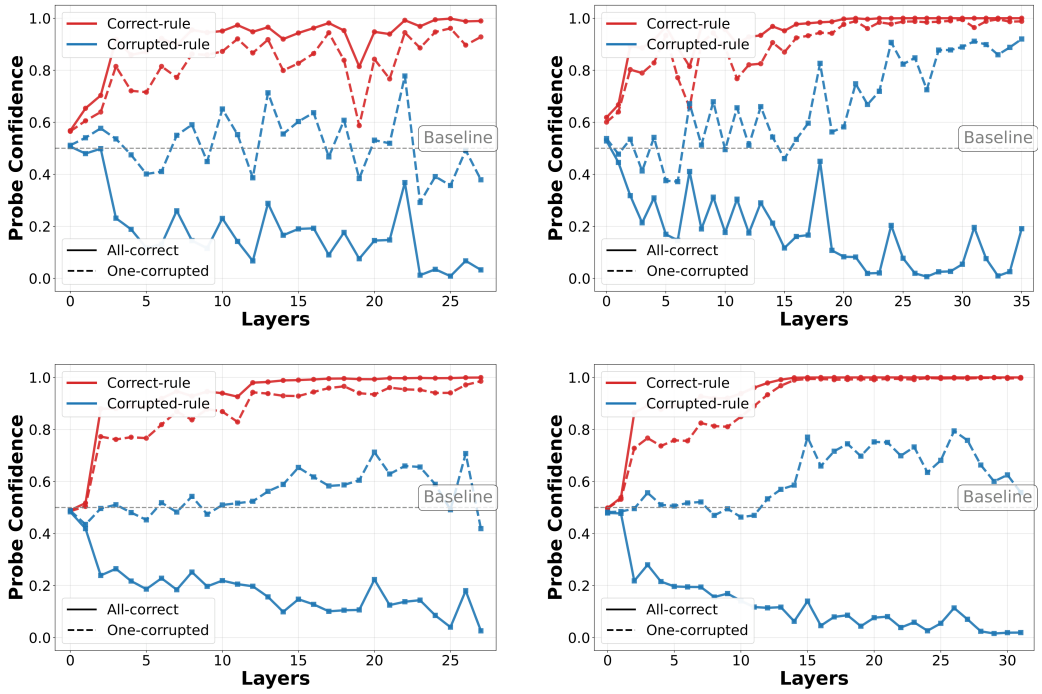


Figure 7: Linear probe confidence across model layers for Qwen3-0.6B, Qwen3-4B, Llama-3.1-8B-Instruct, and Llama-3.2-3B-Instruct from left to right, up to down. **Red solid**: Correct-rule probe confidence in all-correct scenarios. **Red dashed**: Correct-rule probe confidence in one-corrupted scenarios. **Blue solid**: Corrupted-rule probe confidence in all-correct scenarios. **Blue dashed**: Corrupted-rule probe confidence in one-corrupted scenarios. The baseline represents chance-level performance.

A.4 Behavioral evaluation on Qwen3-32B

We extend the 4-shot behavioral evaluation to Qwen3-32B using the same single-position corruption protocol. Table 4 reports mean accuracy over three independent seeds. Clean accuracy is 100% on both tasks. On Operator Induction, corrupting the last demonstration reduces accuracy to 55.6%, a decrease of 44.4 percentage points; 76.3% of Operator errors follow the corrupted operator. Fake Word Inference shows smaller, position-dependent decreases, with no decrease at the last position. This evaluation extends the behavioral result to 32B parameters; the head-level analyses use the four models described in the main experiments.

Table 4: Qwen3-32B accuracy (%) on 4-shot tasks, averaged over three seeds. The corrupted position p is counted from the first demonstration ($p = 1$) to the last ($p = 4$), as in Appendix A.3.

Task	Clean	$p = 1$	$p = 2$	$p = 3$	$p = 4$
Operator Induction	100.0	100.0	95.6	62.2	55.6
Fake Word Inference	100.0	95.0	95.7	94.7	100.0

B Linear Probe Analysis Details

B.1 Probe Training Methodology

We train linear probes as binary classifiers to detect the presence of specific mathematical operators (+, -, ×) in model internal representations. Each probe learns to answer "Is the target rule present in the demonstrations?" rather than performing multi-class operator identification. This design provides a principled approach to measure rule encoding strength across different corruption scenarios.

For each target operator, we generate 600 balanced training samples, containing 300 positive samples for the target rule. Take the Operator Induction task under 4-shot ICL as an example, we compose the positive samples 75 each with 1/4, 2/4, 3/4, 4/4 target demonstrations, and target rules are completely absent in the 300 negative samples. Then, demonstrations are randomly shuffled to eliminate positional effects, ensuring probes learn rule content rather than position patterns. We train simple layer-wise linear probes with L1-regularization (Alain & Bengio, 2016) on the residual streams of the LLMs. Specifically, for each layer l, we predict:

$$\hat{y}^{(i)} = \sigma(\mathbf{w}^T \mathbf{h}_l^{(i)} + b) \tag{5}$$

where $\mathbf{h}_l^{(i)}$ is the representation at layer l for sample i, and we optimize over the regularized objective.

B.2 Two-Scenario Evaluation Protocol

We evaluate trained probes under two controlled scenarios to measure rule encoding dynamics: all Correct and only one corrupted. In the first scenario, all 4 demonstrations exhibit the same target rule (e.g., addition); in the second scenario, 3 demonstrations show the target rule, while 1 demonstration shows conflicting rule (e.g., multiplication or minus)

For each scenario, we extract representations from all layers and compute probe confidence as the measurement for model's encoded confidence of each rule. This yields four evaluation curves: Correct-rule probe confidence in all-correct scenario, correct-rule probe confidence in one-corrupted scenario, corrupted-rule probe confidence in all-correct scenario, and corrupted-rule probe confidence in one-corrupted scenario. To encourage the model to generate the encoded rule instead of the target answer directly, we modify the prompt to elicit direct operator prediction rather than numerical calculation. Specifically, for the Operator Induction tasks, we replace the query format from "6 ? 3 = ?" to:

```
What mathematical operation does ? represent?
Choose from: plus, minus, multiplication

Answer:
```

Figure 7 illustrates these probe confidence patterns across three different model sizes. The results consistently demonstrate our key findings across model architectures. The probe analysis reveals two key mechanistic insights. First, in one-corrupted scenarios, both correct and corrupted rule probes exhibit above-baseline confidence, demonstrating that models encode multiple competing rules simultaneously rather than selecting a single rule early in processing. Second, rule encoding emerges prominently in early-to-middle layers (layers 5-20), with confidence levels stabilizing in later layers.

B.3 Decoding majority and minority operators

The binary probes in Figure 3 predict whether an operator appears anywhere in the demonstrations. This leaves open whether representations also distinguish the operator supported by the three correct demonstrations from the operator introduced by the corrupted one.

We test this distinction on Qwen3-4B in the 4-shot Operator Induction setting with one corrupted demonstration. At each layer, we train two separate three-way classifiers on the residual stream: one predicts the majority operator, and the other predicts the minority operator. The classes are addition, subtraction, and multiplication. We evaluate both classifiers on unseen queries using balanced accuracy.

The majority probe reaches 100% balanced accuracy at every reported layer, starting at layer 5. Minority-operator accuracy rises from 70.8% at layer 5 to 99.2% at layer 21 (Table 5). These probes extend the presence test by identifying each operator's role in the context. Their predictions measure decodable information; the logit lens in Appendix C measures rule probabilities through the model's fixed output projection.

Table 5: Balanced accuracy (%) of probes that identify the majority and minority operators in Qwen3-4B. L denotes the layer.

Target	L5	L10	L18	L19	L20	L21
Majority (correct)	100.0	100.0	100.0	100.0	100.0	100.0
Minority (corrupted)	70.8	76.3	81.0	85.6	96.9	99.2

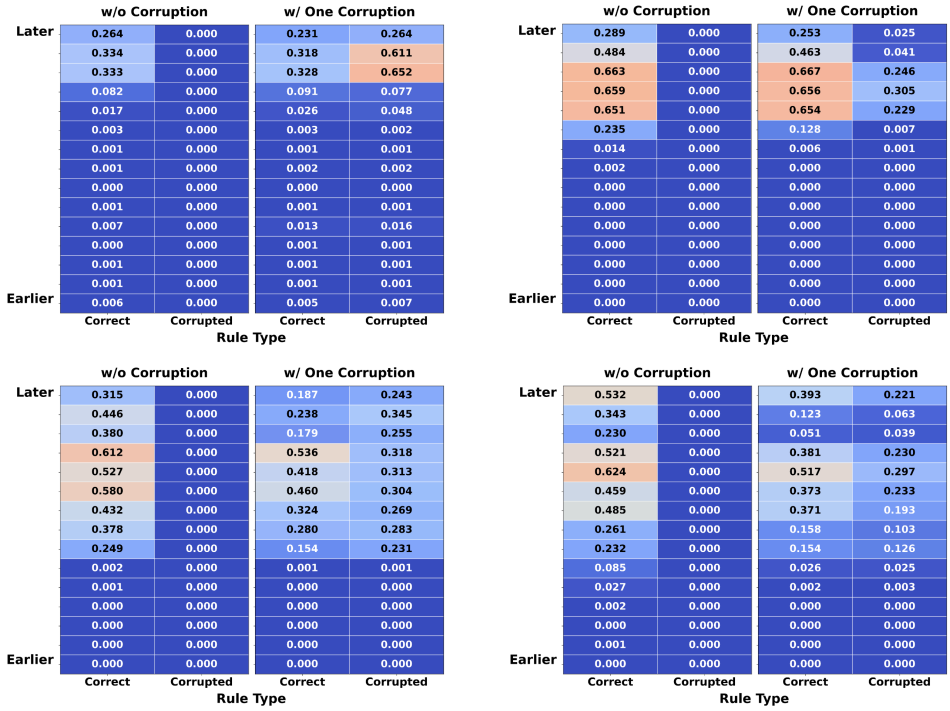


Figure 8: Logit lens results across model layers for Qwen3-0.6B, Qwen3-4B, Llama-3.1-8B-Instruct, and Llama-3.2-3B-Instruct from left to right, up to down.

C Logit Lens Analysis Details

We employ logit lens analysis (Nostalgebraist, 2020) to track rule probabilities across layers under conflicting demonstrations. The logit lens projects intermediate representations from each transformer layer through the final layer normalization and language modeling head to observe predicted token probabilities at different processing stages.

For a hidden state $\mathbf{h}_\ell$ at layer ℓ, the logit lens computes:

$$\mathbf{p}_\ell = \text{softmax}(\mathbf{W} \cdot \text{LayerNorm}(\mathbf{h}_\ell))$$

where $\mathbf{W}$ is the unembedding matrix of the language modeling head and $\mathbf{p}_\ell$ represents the probability distribution over vocabulary tokens as predicted from layer ℓ.

Then, we focus specifically on correct or corrupted rules prediction by analyzing the probability assigned to the words of rules at each layer. Our analysis tracks two primary quantities as below:

Correct Rule Probability The probability assigned to the word of ground-truth rule at layer ℓ:

$$P_\ell(\text{correct}) = \max_{t \in T_{\text{correct}}} \mathbf{p}_\ell[t]$$

where T_{correct} includes all tokens that match or prefix the word of correct rules (Wendler et al., 2024).

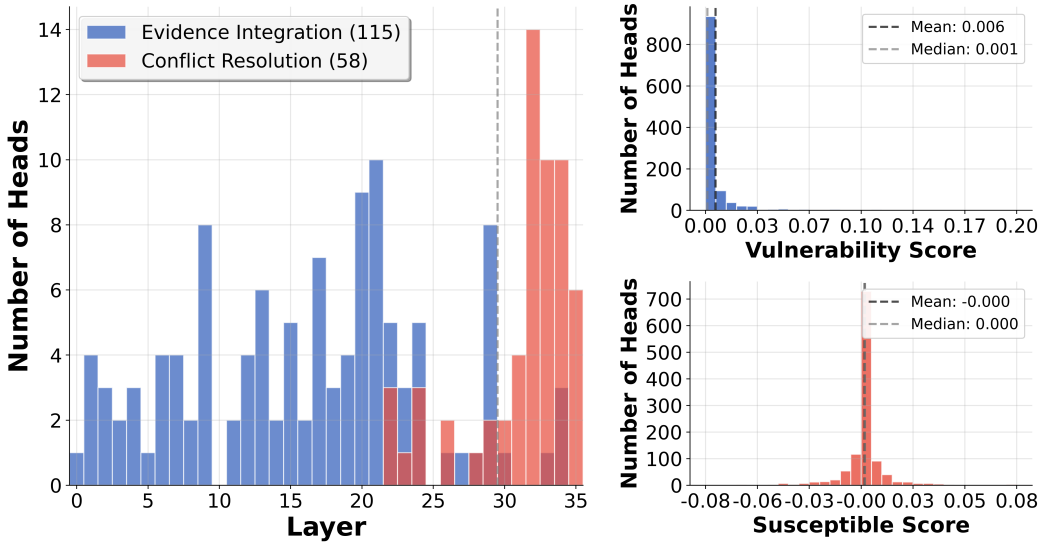


Figure 9: Distribution statistics of vulnerability and susceptible heads for Qwen3-4B model.

Corrupted Rule Probability The probability assigned to the operator introduced through minority corruption:

$$P_\ell(\text{corrupted}) = \max_{t \in T_{\text{corrupted}}} \mathbf{p}_\ell[t]$$

We use the same clean and single-corruption scenarios as the binary probes. Figure 8 reports logit lens results for all four models. Prediction probabilities rise in late layers, showing when rule information becomes readable through the fixed output projection.

D THE STATISTICS OF TWO TYPES OF HEADS

For Qwen3-4B, the layer-wise distribution in Figure 9 shows vulnerability heads predominantly in intermediate layers before 30 and susceptible heads concentrated in final layers, consistent with our two-phase hypothesis. More importantly, the score distributions reveal that our metrics are highly selective rather than universal—the vast majority of attention heads exhibit near-zero scores (median values approach zero for both metrics), while only a small subset shows high vulnerability or susceptibility. This concentration of high scores among a minority of heads validates that our identification methodology successfully isolates functionally specialized components rather than capturing generic attention behaviors. Combined with our causal ablation results, these statistics provide strong evidence that the identified heads play specific, non-redundant roles in creating and failing to resolve conflicts during rule inference.

D.1 LAYER ORDERING ACROSS MODELS

For each model, we select the ten highest-scoring heads in each family on 4-shot Operator Induction and compare their mean layer positions. We normalize layer positions to $[0, 1]$, with 0 at the input side and 1 at the output side. Susceptible Heads have a later mean position than Vulnerability Heads in all four models (Table 6), although the layer boundaries vary across architectures.

Table 6: Mean normalized layer of the top-10 heads in each family on 4-shot Operator Induction.

Model	Vulnerability	Susceptible
Qwen3-0.6B	0.378	0.893
Qwen3-4B	0.323	0.963
Llama-3.2-3B	0.570	0.967
Llama-3.1-8B	0.474	0.913

E Additional Results

E.1 Distribution of head ablation effects

Table 7 summarizes the full Operator Induction grid in Table 1. Each targeted family contributes 12 configurations: four models with 5, 8, or 10 heads ablated. The random control contributes another 12 configurations. We compute the mean, median, and range across configuration means, giving each configuration equal weight. All changes are relative to the unablated corrupted accuracy.

Table 7: Distribution of relative accuracy changes (%) across models and head counts. Positive counts indicate improvement over the unablated corrupted baseline.

Selection	Mean	Median	Minimum	Maximum	Positive
Vulnerability	4.10	2.93	−0.03	11.12	11/12
Susceptible	3.64	1.94	0.93	10.33	12/12
Targeted pooled	3.87	1.96	−0.03	11.12	23/24
Random	−2.52	−2.25	−4.92	−0.44	0/12

Targeted ablation improves 23 of 24 configurations; random ablation reduces accuracy in all 12 controls. Paired comparisons of targeted and random ablation are significant in 20/24 configurations ($p < 0.05$), including all models and both targeted families at head counts 8 and 10. These tests compare the two interventions; the positive counts in Table 7 describe the direction of change from the unablated baseline.

E.2 Stress test of vulnerability score design

Section 4.3.1 characterizes Vulnerability Heads via two complementary attributes: attention allocation (Eq. 2) and corruption sensitivity (Eq. 3), combined multiplicatively into the Vulnerability Score. Here we conduct a stress test to verify that this combination is more selective than either component alone for identifying causally relevant heads.

To check whether small clean-output norms inflate sensitivity, we rank all 1,152 Qwen3-4B heads by sensitivity and inspect the top 10. Their clean output norms have a mean percentile rank of 56.4 among all heads; their unnormalized output changes under corruption have a mean percentile rank of 93.6. The selected heads thus show large output changes without unusually small clean-output norms on average.

Why both components are needed. Attention allocation answers *which heads read from which positions*, while corruption sensitivity answers *which heads change their output when that position is corrupted*. Each is informative but insufficient on its own: high attention to a position is necessary for processing the demonstration but not sufficient for vulnerability, and high sensitivity can also reflect general reactivity to any input perturbation rather than focused vulnerability tied to a specific position. Combining them isolates heads that exhibit both properties simultaneously.

Ablation comparison. We rerun the ablation experiments of Table 1 using attention-only and sensitivity-only rankings as alternative head selection criteria. Table 8 reports relative performance improvements under these two baselines.

Both single-component baselines yield substantially weaker and less consistent improvements than the combined Vulnerability Score in Table 1, often producing negative effects comparable to random ablation. Quantitatively, ablation under the combined Vulnerability Score yields significant ($p < 0.01$) causal effects in 9 of 12 (model, head-count) configurations, compared with 5/12 for attention-only and 3/12 for sensitivity-only. This confirms that attention allocation and corruption sensitivity capture complementary properties, and that their joint use is essential for isolating heads that causally underlie reasoning failures under demonstration corruption.

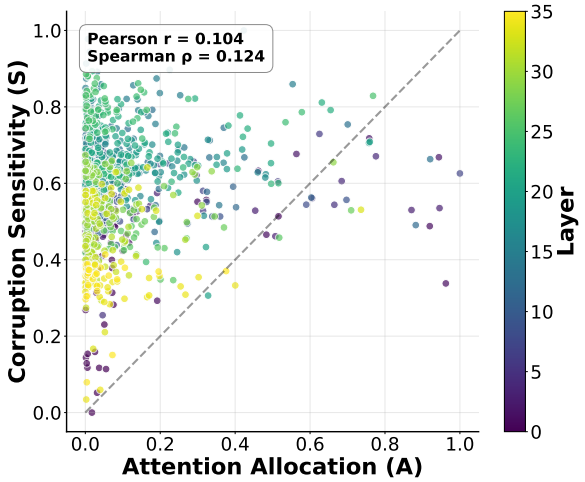


Figure 10: Weak correlation revealed between attention score allocation (A) and corruption sensitivity (S).

Table 8: Ablation comparison between Vulnerability Score (combined), attention-only, and sensitivity-only head selection criteria on Operator Induction. The combined score consistently identifies heads with stronger and more consistent causal influence.

Model	Criterion	5	8	10
Qwen3-0.6B	Attention-only	-1.11±0.65	-1.23±0.38	-3.78±1.58
	Sensitivity-only	-1.24±0.55	-1.24±0.55	-3.00±0.61
Qwen3-4B	Attention-only	0.78±0.11	0.22±0.00	-0.32±0.25
	Sensitivity-only	0.11±0.22	0.67±0.16	0.33±1.20
Llama-3.2-3B	Attention-only	0.56±1.57	5.67±1.40	5.89±1.35
	Sensitivity-only	-0.08±1.55	1.44±1.52	2.78±1.47
Llama-3.1-8B	Attention-only	-0.56±0.42	1.32±0.66	-1.27±1.19
	Sensitivity-only	-0.78±0.46	-1.06±0.97	-2.88±0.39

E.3 Reasoning Uncertainty under Demonstration Conflicts

Beyond classification accuracy, we also examine whether models exhibit calibrated reasoning uncertainty when facing demonstration conflicts. Since our rule inference tasks employ Chain-of-Thought (CoT) reasoning, we track next-token prediction entropy during CoT generation as a proxy for the model's internal uncertainty, following recent work showing that entropy reliably reflects and monitors reasoning confidence (Fu et al., 2025b; Kang et al., 2025; Zur et al., 2025).

We measure the next-token entropy for Qwen3-4B on the Operator Induction queries under clean, corrupted, and corrupted-with-intervention conditions. As shown in Table 9 at Appendix E, demonstration conflicts significantly increase reasoning uncertainty: entropy rises from 0.0952 to 0.1161 nats, indicating the model are more obfuscated when facing conflicting information. Moreover, ablating identified heads partially restores confidence: masking the top-5 Vulnerability Heads and Susceptible Heads both reduce the next-token entropy, approaching the clean baseline. This demonstrates that our identified components causally contribute not only to prediction accuracy but also to elevated reasoning uncertainty, providing additional validation of their mechanistic roles through an orthogonal evaluation metric.

E.4 Vulnerability Heads are Relevant to Positional Bias

Beyond performance improvement, we also notice that ablating vulnerability heads reduces positional bias in corruption sensitivity. We formally define positional bias as $\text{PB} = \text{Var}_{p=1}^{k}[\Delta\text{Acc}_p]$, where ΔAcc_p represents the accuracy decrease when position p is corrupted. As shown in Table 10, removing

Table 9: Average next-token prediction entropy during rule inference tasks under different conditions on Qwen3-4B. Lower entropy indicates higher model reasoning confidence.

Condition	Mean Entropy	Std
Clean	0.0952	0.0239
Corrupted	0.1161	0.0217
Corrupted + Vulnerability Heads masked	0.1148	0.0167
Corrupted + Susceptible Heads masked	0.1120	0.0205

vulnerability heads reduces positional variance moderately across models. Since vulnerability heads exhibit disproportionate attention to certain positions while showing high sensitivity to corruption at those positions, their removal likely mitigates the uneven impact of corruption across demonstration sequences, leading to more uniform vulnerability patterns.

Table 10: Relative positional variance reduction (%) when ablating vulnerability heads.

Models	# of Ablated Heads		
	5	8	10
Qwen3-0.6B	14.26	12.55	12.35
Qwen3-4B	18.45	22.50	19.73
Llama-3.1-8B	20.45	16.12	15.30
Llama-3.2-3B	13.50	16.80	11.95

E.5 Why Logits Attribution over Activation Patching for Locating Susceptible Heads

Activation patching is another widely-used technique to identify causally important model components by targetedly replacing their activations and observing performance changes (Meng et al., 2022; Zhang & Nanda, 2023). While our experimental setup naturally supports activation patching, where we have structured clean and corrupted demonstrations, we employ logits attribution to localize Susceptible Heads for the following reason. Activation patching fails to isolate conflict resolution mechanisms due to the temporal separation of conflict detection and conflict resolution functions. For example, patching heads that encode conflict with clean activations simply relieves the necessity of conflict resolution, consequently creating systematic false positives. In contrast, logits attribution respects this functional separation by measuring each head's direct contribution given the evidence it actually processes.

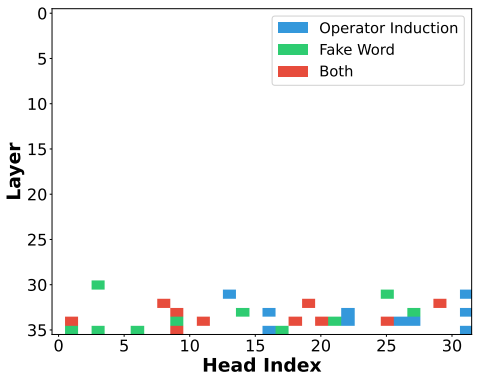


Figure 11: Cross-task overlap of Susceptibility Heads between Operator Induction and Fake Word Inference tasks.

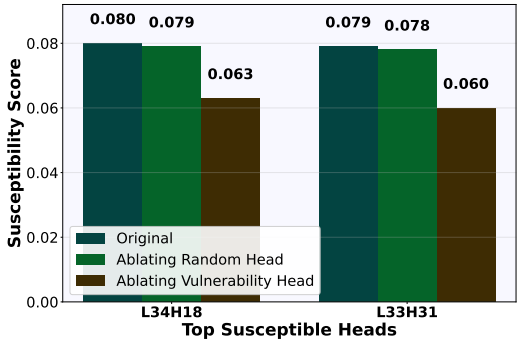


Figure 12: Ablating top vulnerability heads significantly reduces the susceptibility score of top susceptible heads.