

Contrastive Attribution in the Wild: An Interpretability Analysis of LLM Failures on Realistic Benchmarks

Rongyuan Tan¹ * Jue Zhang² † Zhuozhao Li¹ † Qingwei Lin²

Saravan Rajmohan² Dongmei Zhang²

¹Southern University of Science and Technology ²Microsoft

juezhang@microsoft.com lizz@sustech.edu.cn

Abstract

Interpretability methods are increasingly invoked to diagnose Large Language Model (LLM) failures, but most evidence comes from short prompts or toy templates, leaving their utility on practical benchmarks unclear. We cast failure diagnosis as *contrastive attribution* over target–contrast token pairs, attributing the logit gap between incorrect and correct output tokens to input tokens and internal states, and introduce a batch-packed multi-target back-propagation scheme that scales cross-layer attribution graphs to long-context inputs. Instantiating this framework with LRP-style attribution, we evaluate it on IFEval, GAIA2, MATH, and EvalPlus across multiple model sizes and post-training checkpoints. Across coverage, interpretability, reliability, and dynamics, contrastive attribution surfaces recurring failure modes such as underweighting task-relevant tokens and overweighting distractors, demonstrating its practical value as a scalable diagnostic tool for realistic LLM failures.

1 Introduction

While the capabilities of Large Language Models (LLMs) have improved substantially in recent years, they continue to fail in diverse and consequential ways in deployment (Chang et al., 2024). Systematically debugging these failures is therefore crucial for characterizing internal signals associated with observed decisions and developing testable improvement hypotheses.

Existing analyses of LLM failures predominantly rely on *behavioral analysis* of inputs, outputs, and responses to perturbations. Such analyses can identify failures and categorize error types, but do not show how observed token preferences are supported internally. Similar behaviors can correspond to distinct internal attribution

patterns, including underweighting relevant context tokens, overweighting irrelevant tokens, or biases reflected in internal representations. This motivates interpretability-based analysis (Luo and Specia, 2024), which moves beyond behavior to expose how models form preferences and propagate influence across layers.

On the interpretability side, substantial progress has been made for transformer-based models (Ferrando et al., 2024), but applications to LLMs are largely restricted to toy settings, short inputs, or highly constrained prompt templates (Wang et al., 2023; Meng et al., 2022; Lindsey et al., 2025). Interpretability analyses of realistic failures, such as those arising from standard benchmarks or deployed systems, remain scarce, partly because such failures are open-ended and their input contexts can span thousands of tokens (e.g., in agentic tasks), exceeding the computational limits of many existing interpretability tools.¹

Rather than proposing another attribution algorithm, this paper asks a more pragmatic question: *Can interpretability-based analysis provide practical value for LLM failure diagnosis?* Answering this requires moving beyond illustrative case studies toward a population-level study that is explicit about four properties of the diagnostic pipeline: (i) *coverage*, which failures admit a clean explanation setup at all; (ii) *interpretability*, whether the resulting attribution, once obtained, makes the underlying failure pattern readily apparent; (iii) *reliability*, whether the resulting signals are robust to method choice, annotation, and causal probing; and (iv) *dynamics*, how those signals change as models scale or training progresses. Existing work largely treats these properties implicitly, leaving the practical value of attribution-based analysis underexplored.

To make these properties concrete, we instanti-

*Work done during an internship at Microsoft.

†Corresponding authors.

¹For example, the *TransformerLens* package (Nanda and Bloom, 2022) sets the default context length to 2048 tokens for Qwen3 models (Yang et al., 2025a).

ate the diagnostic framework with *contrastive attribution* (Yin and Neubig, 2022), which attributes the logit difference between an incorrect output token and a more correct alternative to input tokens and internal model states, using AttnLRP (Achtibat et al., 2024), a variant of the Layer-wise Relevance Propagation (LRP) framework (Bach et al., 2015), as the underlying attribution algorithm. To make cross-layer hidden-state attribution graphs tractable on long-context inputs (e.g., agentic traces in GAIA2 (Andrews et al., 2025) that can exceed 10k tokens), we introduce a batch-packed multi-target backpropagation scheme. We then conduct a systematic empirical study on hundreds of failures from four widely used benchmarks (IFEval, GAIA2, MATH, EvalPlus), using Qwen3 models up to 8B, a shared-protocol Llama-3 comparison, and OLMo post-training checkpoints. Our findings show that contrastive attribution surfaces recurring failure modes such as underweighting task-relevant tokens and overweighting distractors, and that scaling and continued training drive systematic, interpretable shifts in attribution.

Our contributions are summarized as follows:

- **Diagnostic framing.** We cast LLM failure diagnosis as *contrastive attribution* over target–contrast token pairs anchored at the earliest divergence from a correct trajectory, and treat it as a vehicle for studying when attribution methods help, rather than as an argument for a new attribution method.
- **Scalable cross-layer attribution graphs.** We introduce a batch-packed multi-target backpropagation scheme that recovers hidden-state-level relevance propagation in a single backward pass, enabling attribution-graph construction on long-context (10k+ token) inputs beyond the reach of prior LRP-based graph pipelines.
- **Population-level diagnostic study.** Across four widely used benchmarks (IFEval, GAIA2, MATH, EvalPlus) and multiple model families, model sizes and post-training checkpoints, we report a failure-pattern taxonomy and an attribution-outcome taxonomy, and characterize how attribution patterns evolve under model scaling and post-training stages.²

²Our code is available at: <https://github.com/microsoft/Debug-XAI>.

2 Related Work

Analyzing LLM Failures. LLM failures have been studied primarily through behavioral analyses on standard benchmarks, revealing systematic weaknesses in natural language inference, reasoning, robustness, and bias (Chang et al., 2024), and through extensive work on *hallucination*, including empirical characterizations (Maynez et al., 2020), recent surveys (Ji et al., 2023; Huang et al., 2025; Zhang et al., 2025c), and dedicated benchmarks (Lin et al., 2022; Li et al., 2023; Bang et al., 2025). These studies are informative but largely treat LLMs as black boxes; when interpretability has been applied to such behaviors, it has predominantly been in synthetic or simplified settings such as subject–relation–object templates (Meng et al., 2022; Dai et al., 2022; Geva et al., 2021; Meng et al., 2023; Yao et al., 2024; Lindsey et al., 2025), which abstract away much of the open-endedness of realistic benchmark failures.

Two more recent threads target specific failure regimes. Reasoning failures in large reasoning models have been studied via error categorization (Song et al., 2025), process monitoring (Baker et al., 2025), sparse-feature interpretation (Galichin et al., 2025), and step-level debugging (Zhang et al., 2025a; Li et al., 2025); failures in LLM-powered agent systems have been studied via failure taxonomies (Cemri et al., 2025), localization (Zhang et al., 2025b), and debugging (Ma et al., 2025). These analyses largely operate at the level of reasoning or agent traces rather than at the underlying token-level decisions of the base LLM, which is the focus of this work.

Input Attribution. Input attribution methods estimate the contribution of individual input tokens to model predictions (Ferrando et al., 2024). Two large families are gradient-based attribution, including gradient norms (Simonyan et al., 2014), $\text{gradient} \times \text{input}$ (Denil et al., 2015), integrated gradients (Sundararajan et al., 2017), and Smooth-Grad (Smilkov et al., 2017); and perturbation-based methods, which estimate importance by ablating or noising inputs and measuring the resulting prediction change (Lundberg and Lee, 2017; Fong and Vedaldi, 2017; Covert et al., 2021).

While these methods operate directly on the input space, recent work has explored transformer-specific attribution approaches that exploit the model’s internal computation structure. Rather than estimating token importance solely from input-

output sensitivity, these methods trace how information is propagated and mixed across layers and attention heads.

Attention-based and context-mixing methods decompose final-layer token representations into layer-wise contributions by tracking how attention mixes information across tokens (Kobayashi et al., 2021; Ferrando et al., 2023), while Layer-wise Relevance Propagation (LRP) decomposes scalar quantities additively over inputs. Transformer-specific LRP variants include conservative-propagation LRP (Ali et al., 2022) and AttnLRP (Achtibat et al., 2024), which we use as the underlying attribution algorithm in this work; we further build on recent backpropagation-based efficiency improvements for LRP-style methods (Liu et al., 2025; Arras et al., 2025).

Causal Patching, Circuit Discovery, and Attribution Graphs. Beyond input attribution, prior work attributes behavior to internal states and identifies critical computation paths (*circuits*) via causal patching, including activation patching (Heimersheim and Nanda, 2024; Zhang and Nanda, 2024), attribution patching (Nanda, 2023; Kramár et al., 2024), and path patching (Wang et al., 2023; Goldowsky-Dill et al., 2023), all of which rely on paired clean and corrupted prompts whose construction is often nontrivial for real-world failures. A complementary line trains sparse autoencoders or transcoders to obtain locally linearized activations, enabling circuit discovery and attribution-graph construction (Dunefsky et al., 2024; Ameisen et al., 2025; Zhao et al., 2025; Dai et al., 2025), but these typically require a separate model-specific module and are restricted to short prompts. Decomposition-based attribution graphs constructed from LRP or attention-based approaches (Ferrando and Voita, 2024; Rosser et al., 2025) are more computationally efficient, motivating our AttnLRP-based attribution graphs for the long-context failure cases we study.

3 Methodology

3.1 Contrastive Attribution via LRP

Model. We consider a transformer-based autoregressive LLM (Vaswani et al., 2017) with vocabulary $\mathcal{V}$, hidden dimension d , and L transformer blocks. An embedding matrix $\mathbf{W}_E \in \mathbb{R}^{|\mathcal{V}| \times d}$ maps an input token sequence $\mathbf{t}_{\leq i} = (t_0, \dots, t_i)$ to embeddings $\mathbf{h}_{-1} = \mathbf{W}_E[\mathbf{t}_{\leq i}]$, which are propagated through the stack as $\mathbf{h}_l = \text{Transformer}_l(\mathbf{h}_{l-1})$

for $l = 0, \dots, L - 1$. An unembedding matrix $\mathbf{W}_U \in \mathbb{R}^{d \times |\mathcal{V}|}$ then maps the final hidden state $\mathbf{h}_{L-1}^{(i)}$ at position i to next-token logits:

$$\ell_{i+1} = \mathbf{h}_{L-1}^{(i)\top} \mathbf{W}_U \in \mathbb{R}^{|\mathcal{V}|}. \quad (1)$$

Contrastive attribution objective. We formulate failure analysis as a token-level *contrastive attribution* problem. Given an incorrect target token t_{tgt} and a contrast alternative t_{con} , we attribute the logit difference between them to input token embeddings and hidden states across layers. The contrastive logit difference is

$$\begin{aligned} \Delta\ell &:= \ell_{i+1}(t_{\text{tgt}}) - \ell_{i+1}(t_{\text{con}}) \\ &= \mathbf{h}_{L-1}^{(i)\top} (\mathbf{W}_U[:, t_{\text{tgt}}] - \mathbf{W}_U[:, t_{\text{con}}]), \end{aligned} \quad (2)$$

where $\mathbf{W}_U[:, t]$ denotes the unembedding vector for token t . This contrastive formulation closely reflects practical debugging scenarios by comparing internal support for a specific incorrect token against a plausible alternative and removes shared, failure-irrelevant computation, yielding more salient and interpretable explanations (Yin and Neubig, 2022).

Layer-wise Relevance Propagation. To attribute $\Delta\ell$ to intermediate representations, we employ Layer-wise Relevance Propagation (LRP) (Bach et al., 2015), an additive decomposition: a scalar R_j is decomposed over input features $\mathbf{x} = \{x_i\}_{i=1}^N$ as $R_j = \sum_{i=1}^N R_{i \leftarrow j}$, and when feature i contributes to multiple downstream quantities, its total relevance aggregates as $R_i = \sum_j R_{i \leftarrow j}$. Applying LRP with $\Delta\ell$ as the attribution target and propagating backward through all transformer layers yields a relevance vector $\mathbf{R}_i^{(l)} \in \mathbb{R}^d$ associated with each hidden state $\mathbf{h}_l^{(i)}$, with

$$\mathbf{R}_i^{(l)} = \sum_j \mathbf{R}_{\mathbf{h}_l^{(i)} \leftarrow \mathbf{h}_{l+1}^{(j)}}, \quad (3)$$

where $\mathbf{R}_{\mathbf{h}_l^{(i)} \leftarrow \mathbf{h}_{l+1}^{(j)}}$ is the relevance propagated from $\mathbf{h}_{l+1}^{(j)}$. These relevances define an **Attribution Graph** whose nodes are $\mathbf{R}_i^{(l)}$ and whose edges correspond to propagated relevances.

LRP variant. Because transformers are highly non-linear, LRP relies on module-specific propagation rules that locally linearize model components. Different rules yield different LRP variants. We

instantiate the framework with *AttnLRP* (Achtibat et al., 2024), which trades strict layer-wise relevance conservation for greater attribution sharpness on transformer modules; a direct empirical comparison against gradient-based attribution and other LRP variants in the failure-diagnosis setting is provided in Appendix H.

3.2 Efficient Attribution Graph Construction

Recent work shows that LRP relevance for input embeddings and hidden states can be obtained via a modified *gradient* $\times$ *input* formulation in a single backward pass (Achtibat et al., 2024; Arras et al., 2025). This yields hidden-state relevance scores but not the *propagation structure* between layers for the attribution graph; naively recovering this structure would require backpropagating each hidden-state relevance in layer $l + 1$ as a separate attribution target, which is computationally prohibitive given the cardinality of $\{\mathbf{R}_i^{(l)}\}$.

To address this, we propose a *batch-packed multi-target backpropagation* scheme that reuses the batch dimension to pack multiple attribution targets into a single backward pass, following recent work on transcoder-based attribution graphs (Hanna et al., 2025). This scheme exploits GPU vectorization to recover layer-wise relevance propagation efficiently; details and empirical efficiency gains are in Appendix A and Appendix B, respectively. We further prune attribution targets with relevance below a fixed threshold and remove negligible edges (Appendix C).

3.3 Attribution Granularity

While the above formulation supports neuron-level analysis (Achtibat et al., 2024), we aggregate relevance at the hidden-state level by summing over hidden dimensions, $R_i^{(l)} = \sum_{k=1}^d \mathbf{R}_{i,k}^{(l)}$, and use this aggregate both to analyze input-token attribution patterns via heatmaps (Figure 2) and to build attribution graphs on long-context failures that one-shot neuron-level construction (Ameisen et al., 2025) cannot reach. Important subgraphs may be optionally refined by propagating fine-grained relevance within the selected subgraph; we leave systematic neuron-level analysis to future work.

4 Experiments

4.1 Experimental Setup

Benchmarks. To ensure our benchmark choices align with widely used ones while remaining rel-

evant amid rapidly improving model capabilities, we follow the recent recommendations from HuggingFace (Fourrier et al., 2025). Specifically, we adopt GAIA2 (Andrews et al., 2025) for long-horizon agentic tasks, IFEval (Zhou et al., 2023) for instruction-following, MATH (Hendrycks et al., 2021) for math reasoning, and EvalPlus (Liu et al., 2023, 2024) for code generation.

Models. We primarily use the open-source Qwen3 model series (Yang et al., 2025a) for attribution analyses, and the Olmo-3-7B-Thinking and OLMo3-7B-Instruct post-training paths (Olmo et al., 2025) to study the evolution of attribution patterns across training checkpoints. Due to computation resource constraints, we restrict the model sizes to up to 8B. Benchmark-specific model configurations are summarized in Table 1.

Failure Case Collection. To generate failure cases for analysis, we use different model configurations across benchmarks (Table 1) with greedy decoding for reproducibility. We evaluate the full EvalPlus and IFEval sets due to their small size and randomly sample subsets from the larger GAIA2 and MATH sets. Further collection details are in Appendix D.

Contrast Token Pair Identification. After collecting failure cases, we apply a post-processing step to identify contrast token pairs for attribution analysis. This step can be challenging for two reasons: (i) there may be multiple plausible candidates, or none at all, for the target error token; and (ii) multiple choices may exist for the corresponding contrast token.

We define the *target token* as the **earliest generated token** at which the model departs from a correct trajectory. This is the earliest externally observable divergence from an accepted/correct trajectory, following prior agent-failure attribution work (Zhang et al., 2025b), which likewise localizes failure at the earliest error step. To identify this token reliably, we employ a two-stage pipeline: (1) four independent LLMs each propose the target token, and (2) human annotators validate cases with majority (≥ 2) agreement among LLMs. Inter-agreement among LLM proposers under a relaxed ± 3 -token window exceeds 86% on the three non-agentic benchmarks, and human approval of majority-agreed cases also exceeds 86%, confirming that the identification is reliable (see subsection E.2 for details).

Table 1: Summary of benchmarks and models, number of failure cases, proportion of clean cases used for analysis, and average input token count. Models highlighted in **bold** are used for failure trace generation.

Benchmark	Target Capability	Model	Failure Cases (Clean Case Rate)	Avg. Token Count
IFEval	Instruction following	Qwen3- 0.6B /1.7B/4B	201 (60.7%)	62.6
GAIA2	Agentic; long-context	Qwen3- 4B	100 (40.0%)	12374.1
MATH	Math	Qwen3- 0.6B	62 (59.7%)	136.5
EvalPlus	Coding	Qwen3- 0.6B	270 (50.4%)	106.1

The *contrast token* is selected as the highest-ranked alternative during top- k rollout, using both the models that generated the failure traces and stronger auxiliary models. Crucially, we impose a **strict recovery criterion**: the contrast token must, when substituted at the target position, *recover* the correct trajectory. While this requirement could be relaxed by allowing alternative contrast tokens (even if the underlying model cannot recover from the failure due to limited capability), we enforce it to obtain clean, high-confidence contrastive pairs. Further discussion of this design choice, along with additional experiments using alternative contrast tokens, is provided in [subsection E.4](#). An ablation removing manual and/or stronger-model recovery shows the largest coverage changes on MATH and GAIA2, while M-IA and URT remain the largest categories under every tested configuration. Thus, the recovery mechanisms expand coverage without defining the main qualitative findings; full results are in [subsection E.5](#).

Finally, we impose a logit-difference threshold $\Delta\ell > 1$: this both guards against numerical instability (int8 vs. FP16 inference can induce logit discrepancies up to ~ 0.5) and ensures a meaningfully strong preference (odds ratio $e^1 \approx 2.7$), and removes 18.3% of cases overall. The resulting per-benchmark coverage rates are reported in [Table 1](#), with a per-stage breakdown of how many cases are filtered at each criterion shown in [Table 7](#) of [subsection E.4](#); further procedural details are in [Appendix E](#).

Attribution Analysis. We apply two attribution stages: input attribution via AttnLRP, producing heatmaps of each token’s signed contribution to the logit difference ([Figure 2](#)); and attribution-graph construction via our batch-packed multi-target backpropagation, discarding nodes with $|R| < 0.01$ and pruning edges at 0.85 per-layer cumulative mass. To avoid attention-sink artifacts ([Xiao et al., 2024; Cancedda, 2024](#)), we exclude BOS and other special tokens (gray in [Figure 2](#)) from the normalized heatmap. Clean failure cases are analyzed manually, with inter-annotator agreement as-

sessed on a representative subset ([Appendix G](#)). Reliability checks in the appendices support these analyses: target-token identification shows high LLM–human agreement, post-calibration annotation agreement reaches 88.2%/83.3%, and LRP-based attributions are more perturbation-efficient than gradients in flipping erroneous predictions ([subsection E.2, Appendix H](#)).

4.2 Attribution Analysis of Incorrect Token Preference

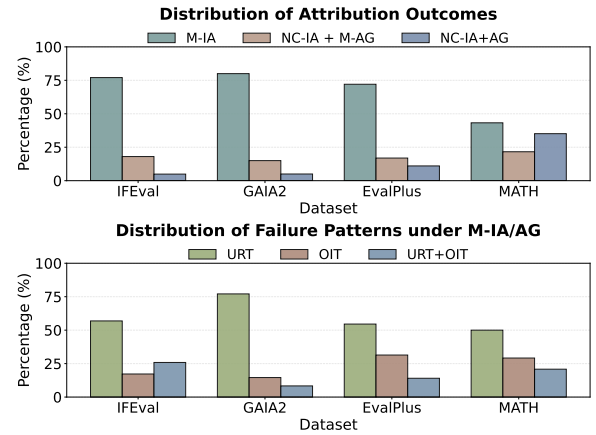


Figure 1: Distribution of attribution outcomes (top) and failure patterns (bottom) across benchmarks.

We first present attribution analysis results across four benchmarks, using the same model employed for failure trace generation. We categorize attribution outcomes as follows:

- **Manifested by Input Attribution (M-IA):** The failure is clearly observable in the input attribution heatmap.
- **No Clue from Input Attribution (NC-IA) + Manifested by Attribution Graph (M-AG):** The failure is not evident from input attribution but becomes apparent via attribution graph analysis.
- **No Clue from Either Analysis (NC-IA + AG):** Neither input attribution nor the attribution graph yields explanatory signal, demanding finer-grained analysis.



Figure 2: Examples of input attribution heatmaps. Colors show per-example max-absolute normalized signed relevance: positive values support the erroneous target, whereas negative values support the contrast. (a) URT: Qwen3-0.6B underweights the instruction token “comma”. (b) OIT: Qwen3-0.6B overweightes an irrelevant token “(a”. (c) NC-IA: input attribution alone offers limited explanation, motivating attribution graph analysis.

The resulting categorization for the four benchmarks is shown in the top panel of Figure 1. IFEval, GAIA2, and EvalPlus exhibit similar patterns: in the majority of cases, input attribution alone suffices to reveal failure signatures. In contrast, for MATH, a substantial fraction of failures remain unexplained even after attribution graph analysis. This limitation arises because our current attribution graphs operate at the level of aggregated hidden states, whereas numerical reasoning errors often require finer-grained, neuron-level attribution analysis (Lindsey et al., 2025): for instance, the *Algebra_9* case in MATH, where Qwen3-0.6B prefers “0” over “1” after “ $(1.0175)^{20} \approx 1.4$ ” (i.e., “1.40 vs. 1.41”), remains unresolved at hidden-state granularity and motivates neuron-level tracing.

For cases that are explainable by attribution analysis (i.e., those categorized as M-IA and NC-IA + M-AG), we further identify the following failure patterns:

- **Underweight Relevant Tokens (URT):** The model assigns insufficient relevance to critical input tokens. Figure 2(a) shows an example from IFEval, where Qwen3-0.6B fails to assign negative relevance to the keyword “comma”, which is essential for producing the correct output. In con-

trast, Qwen3-4B assigns clear negative relevance to this token, aligning with correct behavior.

- **Overweight Irrelevant Tokens (OIT):** The model overemphasizes misleading or semantically uninformative tokens. Note that URT does not imply OIT, as AttnLRP does not enforce relevance conservation. As illustrated in Figure 2(b), an EvalPlus example shows Qwen3-0.6B assigning high relevance to an irrelevant token “(a”, whereas Qwen3-8B assigns much less relevance to that token.
- **URT + OIT:** Failures characterized by both underweighting relevant tokens and overweighting irrelevant ones.

Across all four benchmarks (Figure 1, bottom), underweighting relevant tokens is the dominant failure mode; overweighting irrelevant tokens contributes substantially, particularly for EvalPlus and GAIA2, and is often accompanied by underweighting.

We illustrate the value of attribution graphs with a representative NC-IA + M-AG case in Figure 2(c): the output should begin with “<<” to repeat the request, but the model outputs “First” with

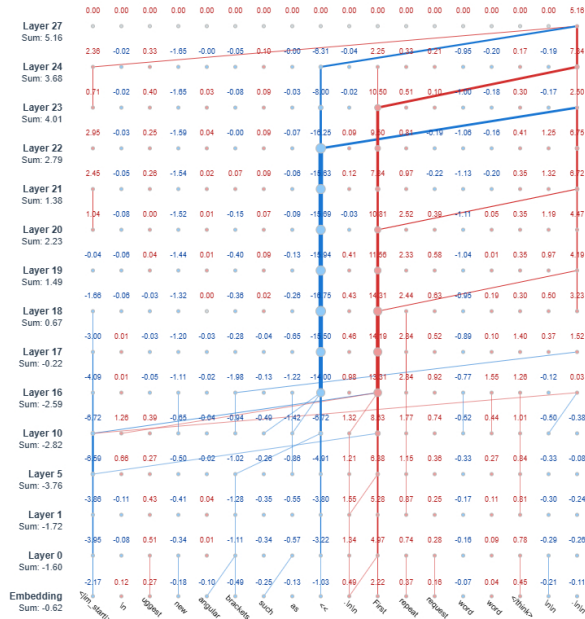


Figure 3: Abridged attribution graph for the NC-IA + M-AG case presented in Figure 2(c); the expanded attribution graph is provided in Figure 10.

a large logit difference of 5.25, even though “<<” and “brackets” carry substantial negative input relevance. The attribution graph in Figure 3 provides complementary layer-wise relevance-flow evidence for this case. The competing tokens grow at similar rates until Layer 16, where “<<” jumps from 6.72 to 14.0 via aggregated relevance from earlier tokens; beyond Layer 16, however, “First” accumulates multiple large increments at the final-token residual stream (Layers 18, 20, 23) that outweigh the contributions of “<<” (Layers 22, 24), producing the positive final logit difference. This case shows how attribution graphs reveal layer-wise relevance-propagation structure that is not visible from input attribution alone. Additional examples and a systematic statistical analysis are provided in Appendix I and Appendix J, respectively.

4.3 Attribution Shifts with Model Scaling

We next investigate whether scaling model size resolves the failures observed in smaller models and, if so, whether such corrections are supported by interpretability evidence.

To this end, we apply Qwen3-1.7B and -4B to the IFEval failures identified for Qwen3-0.6B. As shown in Figure 4, larger models consistently yield more negative logit differences for the contrast token pairs: most points lie below $y = x$ in both the 0.6B vs. 1.7B and 0.6B vs. 4B comparisons,

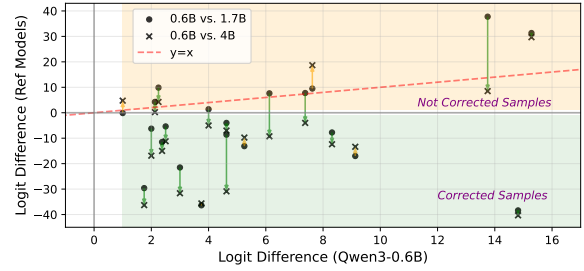


Figure 4: Logit-difference comparisons between Qwen3-0.6B and larger models on IFEval failures. Points below $y=x$ are corrected samples (green region); remaining failures are the yellow region.

and a similar but smaller shift is visible between 1.7B and 4B. The majority of failures are corrected (green region), while a subset remains uncorrected (yellow).

To check whether these corrections are supported by attribution evidence, we run input attribution for all three models on the same cases and compute normalized relevance over three prompt segments (*Instruction*, *Query*, and *Answer*). Total relevance becomes increasingly negative as model size grows (Figure 5, top); the Instruction segment in particular flips from non-negative in 0.6B to clearly negative in 1.7B and 4B, and the Query and Answer segments shift in the same direction, indicating that larger models assign stronger relevance to task requirements and less to superficial continuation from prior answer tokens.

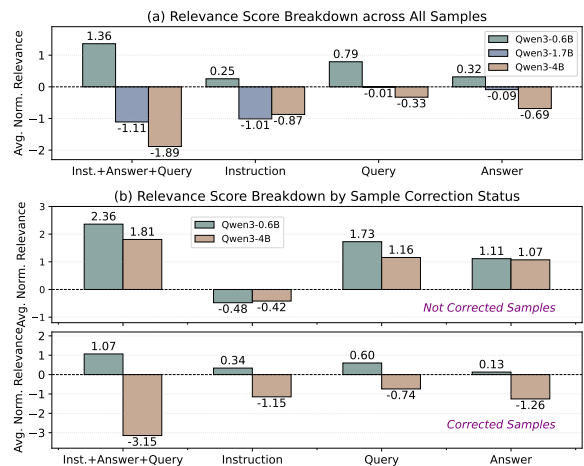


Figure 5: Input attribution relevance score breakdown across prompt segments. (a) Average normalized relevance over all samples. (b) Relevance breakdown by correction status.

Separating corrected from uncorrected samples (Figure 5, bottom) sharpens the picture: corrected

samples show negative relevance across all three segments, while uncorrected samples retain nearly identical distributions across model sizes. Failure correction in larger models is therefore accompanied by systematic, interpretable shifts in attribution rather than uniform per-token rescaling.

4.4 Attribution Shifts with Model Family

We additionally compare model-specific failure collections for Qwen3 model family and Llama-3 model family on IFEval, MATH, and EvalPlus.

Across both families and all three benchmarks, M-IA is the largest attribution outcome and URT the largest failure pattern in Figure 7. The remaining mass is distributed differently between NC-IA+M-AG and NC-IA+AG, and between OIT and URT+OIT, across models and benchmarks. Thus, the dominant categories replicate, but the full distributions do not imply equivalence between families or isolate a model-family effect.

4.5 Evolution of Failure Attribution Patterns Across Training

Beyond model scaling, we examine how attribution evolves during post-training by analyzing input attribution across checkpoints of the Olmo-3-7B-Think series (Olmo et al., 2025).

We identify IFEval failures using an early supervised fine-tuning (SFT) checkpoint SFT_1000 and run input attribution on these same cases across later checkpoints: eight more SFT steps, one Direct Preference Optimization (DPO) checkpoint, and three Reinforcement Learning with Verifiable Rewards (RLVR) checkpoints. Logit differences decrease steadily as training progresses (Figure 8).

Across checkpoints, segment-wise relevance breakdowns (Figure 6) show two large drops in the *Query* and *Answer* segments, between SFT_1000 and SFT_2000 and between SFT_8000 and SFT_10000, that align closely with the corresponding logit-difference reductions in Figure 8; the *Instruction* segment instead decreases more steadily across all checkpoints. The post-DPO checkpoint further pushes relevance in all three segments toward more negative values relative to the final SFT checkpoint (SFT_43000). SFT induces larger attribution shifts than RLVR, likely because IFEval probes instruction-following that SFT targets directly (RLVR primarily targets multi-step reasoning) and because SFT precedes RLVR in the pipeline, leaving less remaining failure mass for RLVR to correct. These results provide checkpoint-

level evidence that training progressively corrects failures through systematic, interpretable shifts in attribution.

A complementary comparison of aligned SFT, DPO, and RL stages across the OLMo3-7B-Thinking and OLMo3-7B-Instruct paths is provided in Appendix F.

4.6 Practical Implications

Beyond its diagnostic role, contrastive attribution offers actionable hypotheses for covered cases through a selective, partly manual workflow. Heatmaps localize prompt segments for inspection, while tested perturbations (Appendix H) flip predictions away from the error token with an average of 1.7 masked tokens for AttnLRP, without establishing causal repair. Potential applications, including training-time monitoring of attribution dynamics and token-importance signals for alignment objectives such as DPO (Rafailov et al., 2023; Yang et al., 2025b), are discussed in Appendix K.

5 Conclusion

We study the practical value of contrastive attribution for diagnosing realistic LLM benchmark failures. Using a target–contrast objective and batch-packed cross-layer attribution graphs, we analyze cases from IFEval, GAIA2, MATH, and EvalPlus across multiple model sizes and post-training stages. Our results show that attribution analysis consistently reveals recurring failure patterns and captures how these patterns shift with model family, scaling and training, demonstrating its promise as a scalable diagnostic framework for real-world LLM evaluation.

Limitations

Hidden-state granularity ceiling on neuron-intensive failures. A core limitation surfaced by our study is the granularity ceiling of hidden-state-level attribution. As shown in the top panel of Figure 1, a substantial fraction of MATH failures fall into the NC-IA + AG category: neither input attribution nor our cross-layer attribution graph yields explanatory signal. The *Algebra_9* case discussed in §4.2 illustrates why: explaining why Qwen3-0.6B prefers “0” over “1” after “ $(1.0175)^{20} \approx 1.4$ ” plausibly requires tracing the underlying power computation at the neuron level (Lindsey et al., 2025), which our hidden-state aggregation deliberately abstracts away to keep cross-layer attribution

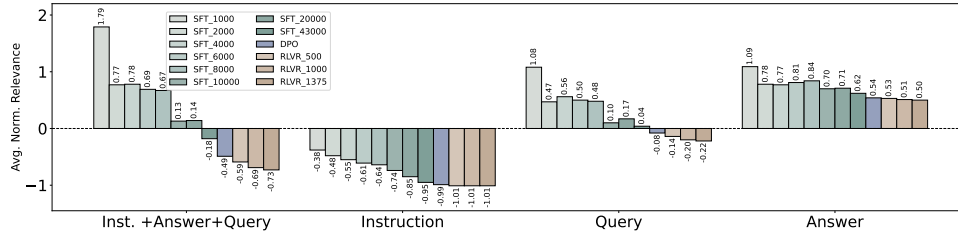


Figure 6: Input attribution relevance score breakdown across training checkpoints by prompt segment. Relevance is shown for *Instruction*, *Query*, and *Answer* segments.

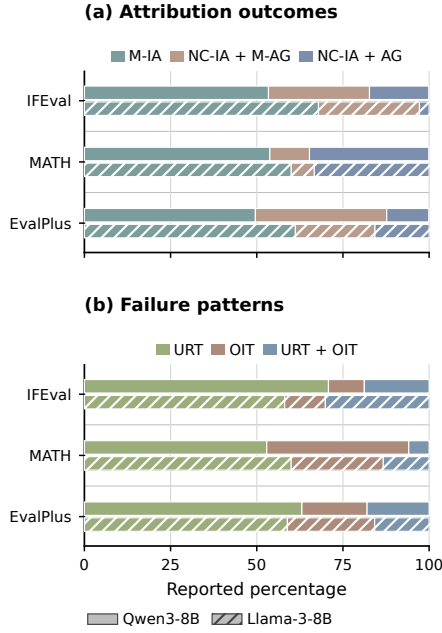


Figure 7: The complete attribution-outcome distribution (top) and complete failure-pattern distribution (bottom) for Qwen3-8B and Llama-3-8B on IFEval, MATH, and EvalPlus. Colors denote categories; solid and hatched bars denote Qwen3-8B and Llama-3-8B, respectively. Attribution percentages use retained cases, whereas failure-pattern percentages use explained cases. Each model uses its own retained failure set under the same protocol, so the comparison is descriptive rather than paired.

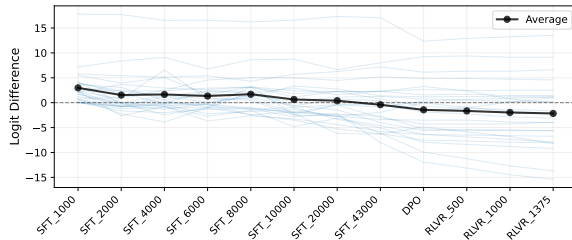


Figure 8: Evolution of logit differences across training checkpoints for Olmo-3-7B-Think on IFEval failure cases identified at SFT_1000. Each line corresponds to an individual sample, with the bold line denoting the average.

graphs tractable on long-context inputs. Addressing this limitation will require selectively refining NC-IA + AG cases at the neuron level and pairing decomposition-based attribution with causal probes such as activation patching (Heimersheim and Nanda, 2024) and neuron-level tracing (Ameisen et al., 2025; Lindsey et al., 2025).

Manual annotation bottleneck and single-token scope. The current pipeline relies on human inspection of attribution heatmaps and attribution graphs to categorize failures into M-IA / NC-IA + M-AG / NC-IA + AG and to identify URT, OIT, and URT + OIT patterns. This limits throughput and reproducibility, and constrains the study to four benchmarks, models up to 8B, and single-token contrastive decisions. Clean-case coverage (40–60% per benchmark, Table 1) is similarly bounded by our strict contrast-recovery criterion, which trades coverage for higher attribution signal quality. Future extensions should automate failure-pattern classification from annotated attribution outputs and extend contrastive attribution from single-token decisions to phrase- or step-level units, better capturing multi-token failures such as multi-step arithmetic in MATH or multi-line code in EvalPlus.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China Grant No. 62202216, the Guangdong Basic and Applied Basic Research Foundation Grant No. 2023A1515010244, Guangdong Key Program (No. 2023QN10X613), and the Shenzhen Science and Technology Program Grant 20231121101752002. This work was also supported by Center for Computational Science and Engineering at Southern University of Science and Technology.

We thank Haibin Lai for support with experimental annotation, Bo Qiao for his invaluable assistance in maintaining and supporting the experimental in-

frastructure, and Bowen Zhang and Yangfan Qiao for their helpful suggestions on our demo pipeline.

References

- Reduan Achitbat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. 2024. [Attnlrp: Attention-aware layer-wise relevance propagation for transformers](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. 2022. [XAI for transformers: Better explanations through conservative propagation](#). In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 435–451. PMLR.
- 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, and 8 others. 2025. [Circuit tracing: Revealing computational graphs in language models](#). *Transformer Circuits Thread*.
- Pierre Andrews, Amine Benhalloum, Gerard Moreno-Torres Bertran, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Romain Froger, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, Kunal Malkan, Dheeraj Mekala, Pierre Ménard, Grégoire Mialon, Ulyana Piterbarg, Mikhail Plekhanov, Mathieu Rita, and 5 others. 2025. [Are: Scaling up agent environments and evaluations](#). *Preprint*, arXiv:2509.17158.
- Leila Arras, Bruno Puri, Patrick Kahardipraja, Sebastian Lapuschkin, and Wojciech Samek. 2025. [A close look at decomposition-based xai-methods for transformer language models](#). *Preprint*, arXiv:2502.15886.
- Shir Ashury-Tahan, Yifan Mai, Elron Bandel, Michal Shmueli-Scheuer, and Leshem Choshen. 2026. [Errormap and erroratlas: Charting the failure landscape of large language models](#). *Preprint*, arXiv:2601.15812.
- Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.
- 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.
- Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. 2025. [Hallulens: Llm hallucination benchmark](#). *Preprint*, arXiv:2504.17550.
- Nicola Cancedda. 2024. Spectral filters, dark signals, and attention sinks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4792–4808.
- Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. [Why do multi-agent llm systems fail?](#) *Preprint*, arXiv:2503.13657.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. [A survey on evaluation of large language models](#). *ACM Trans. Intell. Syst. Technol.*, 15(3).
- Ian Covert, Scott M. Lundberg, and Su-In Lee. 2021. [Explaining by removing: A unified framework for model explanation](#). *J. Mach. Learn. Res.*, 22:209:1–209:90.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. [Knowledge neurons in pretrained transformers](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.
- Xinnan Dai, Kai Guo, Chung-Hsiang Lo, Shenglai Zeng, Jiayuan Ding, Dongsheng Luo, Subhabrata Mukherjee, and Jiliang Tang. 2025. [Graphghost: Tracing structures behind large language models](#). *Preprint*, arXiv:2510.08613.
- DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. [Deepseek-v3.2: Pushing the frontier of open large language models](#). *Preprint*, arXiv:2512.02556.
- Misha Denil, Alban Demiraj, and Nando de Freitas. 2015. [Extraction of salient sentences from labelled documents](#). *Preprint*, arXiv:1412.6815.
- Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. 2024. Transcoders find interpretable LLM feature circuits. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

- Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. [Explaining how transformers use context to build predictions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2023, Toronto, Canada, July 9-14, 2023, pages 5486–5513. Association for Computational Linguistics.
- Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. [A primer on the inner workings of transformer-based language models](#). Preprint, arXiv:2405.00208.
- Javier Ferrando and Elena Voita. 2024. [Information flow routes: Automatically interpreting language models at scale](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 17432–17445. Association for Computational Linguistics.
- Ruth C. Fong and Andrea Vedaldi. 2017. [Interpretable explanations of black boxes by meaningful perturbation](#). In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3449–3457. IEEE Computer Society.
- Clémentine Fourrier, Thibaud Frere, Guilherme Penedo, and Thomas Wolf. 2025. [The llm evaluation guidebook](#).
- 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.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. [Transformer feed-forward layers are key-value memories](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. [Localizing model behavior with path patching](#). Preprint, arXiv:2304.05969.
- Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. 2025. [Circuit-tracer: A new library for finding feature circuits](#). In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 239–249, Suzhou, China. Association for Computational Linguistics.
- 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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. [A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions](#). *ACM Trans. Inf. Syst.*, 43(2).
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Comput. Surv.*, 55(12).
- Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2021. [Incorporating residual and normalization layers into analysis of masked language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4547–4568. Association for Computational Linguistics.
- János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. [Atp*: An efficient and scalable method for localizing LLM behaviour to components](#). *CoRR*, abs/2403.00745.
- Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. [HaluEval: A large-scale hallucination evaluation benchmark for large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.
- Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Xiao Liang, Zhi-jiang Guo, and 2 others. 2025. [From system 1 to system 2: A survey of reasoning large language models](#). Preprint, arXiv:2502.17419.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.
- Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermy, Andy Jones, and 8 others. 2025. [On the biology of a large language model](#). *Transformer Circuits Thread*.

- Fengyuan Liu, Nikhil Kandpal, and Colin Raffel. 2025. [Attribot: A bag of tricks for efficiently approximating leave-one-out context attribution](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. [Is your code generated by chat-GPT really correct? rigorous evaluation of large language models for code generation](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jiawei Liu, Songrun Xie, Junhao Wang, Yuxiang Wei, Yifeng Ding, and Lingming Zhang. 2024. [Evaluating language models for efficient code generation](#). In *First Conference on Language Modeling*.
- Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774.
- Haoyan Luo and Lucia Specia. 2024. [From understanding to utilization: A survey on explainability for large language models](#). *Preprint*, arXiv:2401.12874.
- Ming Ma, Jue Zhang, Fangkai Yang, Yu Kang, Qingwei Lin, Tianming Yang, Saravan Rajmohan, and Dongmei Zhang. 2025. [Dover: Intervention-driven auto debugging for llm multi-agent systems](#). *Preprint*, arXiv:2512.06749.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- 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.
- Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. [Mass-editing memory in a transformer](#). In *The Eleventh International Conference on Learning Representations*.
- Neel Nanda. 2023. [Attribution patching: Activation patching at industrial scale](#).
- Neel Nanda and Joseph Bloom. 2022. [Transformerlens](#). <https://github.com/TransformerLensOrg/TransformerLens>.
- Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, and 50 others. 2025. [Olmo 3](#). *Preprint*, arXiv:2512.13961.
- OpenAI. 2025a. [gpt-oss-120b & gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- OpenAI. 2025b. [Introducing gpt-5](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- J Rosser, José Luis Redondo García, Gustavo Penha, Konstantina Palla, and Hugues Bouchard. 2025. [Stream: Scaling up mechanistic interpretability to long context in llms via sparse attention](#). *Preprint*, arXiv:2510.19875.
- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. [Deep inside convolutional networks: Visualising image classification models and saliency maps](#). In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.
- Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. 2017. [Smoothgrad: removing noise by adding noise](#). *CoRR*, abs/1706.03825.
- Peiyang Song, Pengrui Han, and Noah Goodman. 2025. [A survey on large language model reasoning failures](#). In *2nd AI for Math Workshop @ ICML 2025*.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. [Axiomatic attribution for deep networks](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.
- Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. [Kimi k2: Open agentic intelligence](#). *Preprint*, arXiv:2507.20534.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. [Interpretability in the wild: a circuit for indirect object identification in GPT-2 small](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

- 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*.
- Zishan Xu, Shuyi Xie, Qingsong Lv, Shupeixiao, Linlin Song, Sui Wenjuan, and Fan Lin. 2025. Diagnosing failures in large language models’ answers: Integrating error attribution into evaluation framework. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21148–21165.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chuji Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Ning Yang, Hai Lin, Yibo Liu, Baoliang Tian, Guoqing Liu, and Haijun Zhang. 2025b. [Token-importance guided direct preference optimization](#). *Preprint*, arXiv:2505.19653.
- Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and HuaJun Chen. 2024. [Knowledge circuits in pretrained transformers](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 118571–118602. Curran Associates, Inc.
- Kayo Yin and Graham Neubig. 2022. [Interpreting language models with contrastive explanations](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 184–198. Association for Computational Linguistics.
- Fred Zhang and Neel Nanda. 2024. [Towards best practices of activation patching in language models: Metrics and methods](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Jue Zhang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2025a. [From reasoning to answer: Empirical, attention-based and mechanistic insights into distilled DeepSeek r1 models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3985–4002, Suzhou, China. Association for Computational Linguistics.
- Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. 2025b. [Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems](#). In *Forty-second International Conference on Machine Learning*.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025c. [Siren’s song in the ai ocean: A survey on hallucination in large language models](#). *Computational Linguistics*, 51(4):1373–1418.
- Zheng Zhao, Yeskendir Koishekenov, Xianjun Yang, Naila Murray, and Nicola Cancedda. 2025. [Verifying chain-of-thought reasoning via its computational graph](#). *Preprint*, arXiv:2510.09312.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

A Batch-Packed Multi-Target Backpropagation for Attribution Graph Construction

This appendix presents an efficient procedure for constructing attribution-graph edges under Layer-wise Relevance Propagation (LRP). The method leverages recent results showing that LRP-style relevance propagation in transformer models can be implemented via a *modified gradient* $\times$ *input* formulation, enabling direct use of automatic differentiation frameworks such as PyTorch (Achtibat et al., 2024; Arras et al., 2025). Building on this equivalence, we further exploit the batch dimension to propagate relevance from multiple attribution targets simultaneously, substantially reducing computational cost.

Why gradients represent relevance. LRP aims to decompose a scalar attribution objective, such as a contrastive logit difference $\Delta\ell$ as in this work, into contributions from intermediate activations. For non-linear architectures like transformers, recent work has shown that LRP rules can be implemented by *patching the backward pass* with customized local propagation rules. Crucially, these patched backward passes are equivalent to computing a modified gradient $\times$ input quantity, allowing relevance to be obtained using standard automatic differentiation (Achtibat et al., 2024; Arras et al., 2025).

Under this formulation, the relevance of a hidden state $\mathbf{h}_l^{(i)} \in \mathbb{R}^d$ is approximated by

$$\mathbf{R}_l^{(i)} \approx \mathbf{h}_l^{(i)} \odot \frac{\partial \Delta\ell}{\partial \mathbf{h}_l^{(i)}}, \quad (4)$$

where the gradient indicates how sensitive the attribution objective is to the hidden state. Thus, gradients act as the carriers of relevance, and no explicit relevance propagation rules need to be implemented beyond the modified backward pass.

Problem setup. We seek to construct an *attribution graph* whose nodes correspond to hidden-state relevances and whose edges describe how relevance propagates between layers. Fix a source layer s and a target layer t with $s < t$. Let

$$\mathbf{H}^{(s)} \in \mathbb{R}^{1 \times n \times d} \quad (5)$$

denote the source-layer hidden states for a sequence of length n , and let

$$\mathbf{H}^{(t)} = F_{s \rightarrow t}(\mathbf{H}^{(s)}) \in \mathbb{R}^{1 \times n \times d} \quad (6)$$

denote the corresponding target-layer hidden states, where $F_{s \rightarrow t}$ is the composition of transformer sub-modules between layers s and t .

A single forward and backward pass from $\Delta\ell$ yields gradients

$$\mathbf{g}_t^{(i)} = \frac{\partial \Delta\ell}{\partial \mathbf{h}_t^{(i)}} \quad (7)$$

at every hidden state. These gradients determine node relevances via $\text{gradient} \times \text{input}$ and are stored for later use. At this stage, we know *which* hidden states are relevant, but not *how* relevance flows between hidden states across layers.

Edge attribution via $\text{gradient} \times \text{input}$. To construct attribution-graph edges, we propagate relevance from a target-layer hidden state $\mathbf{h}_t^{(j)}$ to all source-layer hidden states $\{\mathbf{h}_s^{(i)}\}_{i=1}^n$. We treat the stored gradient $\mathbf{g}_t^{(j)} = \partial \Delta\ell / \partial \mathbf{h}_t^{(j)}$ as a fixed relevance signal. Because automatic differentiation requires a scalar objective, we define the attribution target as the inner product

$$\langle \mathbf{h}_t^{(j)}, \mathbf{g}_t^{(j)} \rangle, \quad (8)$$

which simply selects and weights the hidden dimensions that are relevant for the final decision. Backpropagating this scalar through $F_{s \rightarrow t}$ yields gradients with respect to $\mathbf{H}^{(s)}$.

Relevance propagated from target token j to source token i is then computed using the same $\text{gradient} \times \text{input}$ rule:

$$A_{j,i} \approx \sum_{k=1}^d \mathbf{h}_s^{(i,k)} \frac{\partial \langle \mathbf{h}_t^{(j)}, \mathbf{g}_t^{(j)} \rangle}{\partial \mathbf{h}_s^{(i,k)}}. \quad (9)$$

This quantity defines a directed edge from node (s, i) to node (t, j) in the attribution graph.

Batch-packed multi-target backpropagation.

Naively, computing $A_{j,i}$ for all $j \in \{1, \dots, n\}$ would require $O(n)$ separate backward passes, one per target token. To avoid this cost, we exploit the batch dimension to process multiple attribution targets simultaneously, following recent work on attribution graph construction with sparse features in transcoders (Hanna et al., 2025).

Let $\mathcal{J} = \{j_1, \dots, j_B\}$ be a set of target indices. We construct a batched source tensor

$$\tilde{\mathbf{H}}^{(s)} \in \mathbb{R}^{B \times n \times d}, \quad \tilde{\mathbf{H}}_{b,:}^{(s)} = \mathbf{H}_{1,:}^{(s)}, \quad (10)$$

so that each batch element contains an identical copy of the source-layer hidden states. Forward propagation yields

$$\tilde{\mathbf{H}}^{(t)} = F_{s \rightarrow t}(\tilde{\mathbf{H}}^{(s)}) \in \mathbb{R}^{B \times n \times d}. \quad (11)$$

We then define a batched output-gradient tensor $\tilde{\mathbf{G}}^{(t)} \in \mathbb{R}^{B \times n \times d}$ such that batch element b injects relevance only at its corresponding target index j_b :

$$\tilde{\mathbf{G}}_{b,j_b,:}^{(t)} = \mathbf{g}_t^{(j_b)}, \quad \tilde{\mathbf{G}}_{b,j,:}^{(t)} = \mathbf{0} \text{ for } j \neq j_b. \quad (12)$$

A single automatic differentiation call computes the corresponding gradients with respect to $\tilde{\mathbf{H}}^{(s)}$, yielding B independent relevance propagations in parallel. Applying $\text{gradient} \times \text{input}$ and writing each batch result into the appropriate row produces B rows of the token-to-token interaction matrix.

Thus, by batching attribution targets, we reduce the cost of constructing attribution-graph edges from $O(n)$ backward passes to $O(\lceil n/B \rceil)$ passes, while remaining fully compatible with LRP-style relevance propagation through standard automatic differentiation.

B Computational Efficiency of Batch-Packed Backpropagation

This appendix provides empirical efficiency measurements for the batch-packed multi-target backpropagation method introduced in Appendix A. All experiments are conducted on a single NVIDIA A100 GPU using Qwen3-0.6B unless otherwise stated.

Scaling with batch size. Table 2 reports latency and peak GPU memory across different batch sizes B and input token lengths. Each cell shows “latency (seconds) / memory (MB)”.

Batching yields significant speedups across diverse input lengths (although the gain diminishes for longer sequences), with memory scaling linearly as expected.

Table 2: Latency and memory of batch-packed backpropagation across batch sizes (Qwen3-0.6B). Each cell shows “latency (seconds) / memory (MB)”.

Token length	$B=1$	$B=4$	$B=8$	$B=16$
63	0.26 / 3.5	0.08 / 14.0	0.05 / 27.9	0.02 / 55.8
252	1.26 / 13.7	0.25 / 53.7	0.14 / 107.1	0.09 / 215.3
754	2.63 / 42.4	1.02 / 164.6	0.77 / 327.0	0.63 / 643.6
1508	6.23 / 89.2	3.08 / 333.1	2.67 / 648.9	3.45 / 1290.0

Scaling with model size. Table 3 reports latency and memory across different Qwen3 model sizes at a fixed batch size of $B=8$.

Both latency and memory scale roughly proportionally to model size, indicating that the method remains efficient and scalable for larger models.

C Attribution Graph Pruning and Connected Subgraph Extraction

This appendix describes the pruning strategy used to construct a sparse and interpretable *attribution graph* from the dense token-to-token interaction matrices produced by batch-packed multi-target backpropagation (Appendix A). The objective of pruning is to remove low-relevance edges while preserving the dominant relevance flows between hidden states across layers.

Input representation. For each ordered pair of layers (s, t) with $s < t$, we assume access to:

- a token-to-token interaction matrix

$$\mathbf{A}^{(s \rightarrow t)} \in \mathbb{R}^{n \times n}, \quad (13)$$

where $\mathbf{A}_{j,i}^{(s \rightarrow t)}$ denotes the relevance propagated from source-layer hidden state $\mathbf{h}_s^{(i)}$ to target-layer hidden state $\mathbf{h}_t^{(j)}$, as defined by the gradient $\times$ input formulation in Appendix A;

- source-layer token relevances

$$R_i^{(s)} \equiv \sum_{k=1}^d \mathbf{R}_s^{(i,k)}; \quad (14)$$

- target-layer token relevances

$$R_j^{(t)} \equiv \sum_{k=1}^d \mathbf{R}_t^{(j,k)}. \quad (15)$$

Each nonzero entry of $\mathbf{A}^{(s \rightarrow t)}$ corresponds to a candidate directed edge from node (s, i) to node (t, j) in the attribution graph.

Graph construction. We construct a directed graph $\mathcal{G} = (V, E)$, where each node is indexed by a layer–token pair (l, i) . For each layer transition (s, t) , we:

1. add all source nodes (s, i) such that $|R_i^{(s)}| > 0$;
2. add all target nodes (t, j) such that $|R_j^{(t)}| > 0$;
3. add directed edges $(s, i) \rightarrow (t, j)$ for selected entries $\mathbf{A}_{j,i}^{(s \rightarrow t)}$ after pruning, with edge weight

$$w_{(s,i) \rightarrow (t,j)} = \mathbf{A}_{j,i}^{(s \rightarrow t)}. \quad (16)$$

Pruning objectives. The interaction matrices $\mathbf{A}^{(s \rightarrow t)}$ are generally dense, making direct graph construction impractical. We therefore prune edges using magnitude-based criteria applied to $|\mathbf{A}^{(s \rightarrow t)}|$. Pruning is performed *independently for each layer pair* to preserve local attribution structure and avoid conflating relevance scales across layers.

Let

$$E_{\text{nz}}^{(s \rightarrow t)} = \{(j, i) \mid |\mathbf{A}_{j,i}^{(s \rightarrow t)}| > 0\} \quad (17)$$

denote the set of nonzero candidate edges, and let

$$M^{(s \rightarrow t)} = \sum_{(j,i) \in E_{\text{nz}}^{(s \rightarrow t)}} |\mathbf{A}_{j,i}^{(s \rightarrow t)}| \quad (18)$$

denote the total absolute relevance mass for that layer transition.

Pruning modes. We support two pruning strategies:

- *Global threshold pruning.* In this mode, we retain all edges whose absolute relevance exceeds a fixed threshold $\tau > 0$:

$$E_{\text{keep}}^{(s \rightarrow t)} = \{(j, i) \mid |\mathbf{A}_{j,i}^{(s \rightarrow t)}| > \tau\}. \quad (19)$$

This mode enforces a uniform relevance scale across all layer transitions and is most appropriate when attribution magnitudes are comparable across layers.

Table 3: Latency and memory of batch-packed backpropagation across model sizes ($B=8$). Each cell shows “latency (seconds) / memory (MB)”.

Token length	Qwen3-0.6B	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
63	0.04/27.9	0.04/47.7	0.04/75.4	0.04/92.8
252	0.13/107.1	0.13/189.8	0.21/294.3	0.32/371.1
754	0.57/321.6	1.07/570.6	1.86/882.4	3.01/1117.6
1508	2.41/647.5	4.41/1147.0	9.99/1769.2	13.33/2228.1

- *Per-layer cumulative mass pruning.* In this mode, pruning is performed adaptively per layer transition by preserving a fixed fraction of the total relevance mass. Let

$$\{a_1 \geq a_2 \geq \dots \geq a_K\}$$

denote the sorted absolute values of all nonzero entries of $\mathbf{A}^{(s \rightarrow t)}$, where $K = |E_{\text{nz}}^{(s \rightarrow t)}|$. Given a percentile parameter $p \in (0, 1]$, we select the smallest index k^* such that

$$\sum_{k=1}^{k^*} a_k \geq p \cdot \sum_{k=1}^K a_k. \quad (20)$$

The resulting dynamic threshold is

$$\tau_{s \rightarrow t}^{(p)} = a_{k^*}, \quad (21)$$

and we retain all edges satisfying

$$|\mathbf{A}_{j,i}^{(s \rightarrow t)}| \geq \tau_{s \rightarrow t}^{(p)}. \quad (22)$$

This strategy guarantees that a fixed proportion of relevance mass is preserved for each layer transition, even when absolute relevance scales differ substantially across layers.

Connected attribution subgraph extraction.

After pruning, the attribution graph may be disconnected. Since our attribution objective is the contrastive logit difference at the final token position, we retain only the subgraph that contributes to this prediction. Specifically, we select the target node corresponding to the hidden state at the last layer and last token position, i.e., $(L-1, n)$, and extract the induced subgraph consisting of all nodes that can reach the target node via directed paths. All nodes and edges not connected to this target node are discarded.

Resulting graph properties. The final attribution graph $\mathcal{G}_{\text{conn}}$ is sparse, directed, and weighted. Each node (l, i) is annotated with its layer index, token position, and scalar node relevance $R_i^{(l)}$, while

each directed edge encodes signed relevance flow between hidden states across layers. By construction, every node and edge in $\mathcal{G}_{\text{conn}}$ lies on at least one relevance path terminating at the final prediction, enabling faithful path tracing, subgraph analysis, and visualization of dominant attribution circuits.

D More Details on Failure Case Collection

Here we provide additional details on failure case collection. For IFEval, MATH, and EvalPlus, we disable the models’ thinking mode to simplify reasoning traces and enable clearer error attribution. Prior work has shown that reflective reasoning can introduce non-monotonic trajectories that complicate analysis (Xu et al., 2025; Ashury-Tahan et al., 2026). In contrast, for the GAIA2 benchmark, we retain the default configuration with thinking mode enabled, using the official Meta-ARE system (Andrews et al., 2025), as GAIA2 involves complex multi-turn tasks where thinking mode supports long-horizon coherence.

To verify that our conclusions are robust to this design choice, we conduct a small controlled study. Specifically, we compare attribution distributions and failure mode classifications with and without thinking mode on 21 clean IFEval cases using Qwen3-0.6B. Of these, 16 cases yield identical annotations. Among the 5 remaining cases, 1 results in unbounded generation (and is excluded), while the other 4 exhibit shifts in the target–contrast pairs due to altered reasoning trajectories, reflecting changes in model behavior rather than instability in our method. Table 4 reports the distribution of failure patterns in the two settings. The overall attribution patterns are largely preserved: the majority of cases remain consistent, and URT remains the dominant failure pattern in both settings. This suggests that the overall patterns are largely robust to the thinking mode setting. The unbounded-generation case occurred only in the with-thinking setting, accounting for the difference

between $N=20$ and $N=19$ in Table 4.

Table 4: Failure pattern distribution on IFEval with and without thinking mode (Qwen3-0.6B).

Metric	w/o thinking	w/ thinking
# Cases	20	19
URT	70.0%	58.0%
OIT	5.0%	10.5%
URT+OIT	25.0%	31.5%

E Contrast Token Pair Identification

This appendix consolidates the procedures for (i) identifying the first error token (target token) in a model-generated completion, and (ii) selecting an appropriate contrast token for contrastive attribution. The details on target token agreement statistics and the discussions on the impact of our existing contrast token selection strategy are also presented.

E.1 Target Token Identification

For each failure case, we identify the earliest externally observable divergence from a correct trajectory and refer to the corresponding output token as the target token. This token is an analysis anchor and may be a downstream symptom of an earlier internal error rather than its mechanistic origin. We adopt a semi-automated annotation procedure to identify the target token. Specifically, we prompt multiple powerful LLMs including GPT-5.2-20251211 (OpenAI, 2025b), GPT-OSS-120B (OpenAI, 2025a), DeepSeek-V3.2 (DeepSeek-AI et al., 2025) and Kimi-K2-Thinking (Team et al., 2025), to independently predict the most likely target token for each failure case. The token receiving the majority of votes is selected as the automated result. Human annotators then verify this prediction and correct it when necessary, balancing efficiency with precision. The prompt template used in the above semi-automated annotation procedure is unified across all LLMs to ensure consistent and reproducible identification of the target token. The full template is shown below (using GAIA2 as an example). In the prompt template, “causes the error” is annotation shorthand rather than a claim of mechanistic causation.

```
version: v1
system: |-
- You are an expert in localizing error
  tokens in model-generated text
  completions.
```

- You will be provided with a prompt and a model-generated completion, and your task is to localize and output the FIRST token in the completion that is erroneous or causes the error.
- As the prompt and completion are taken from a particular dataset, refer to Section "Dataset Description" for a brief overview of the dataset.
- Refer to Section "On the Input Format" for details on how the prompt and completion will be presented to you.
- Follow the "General Guidance \ (Dataset Agnostic)" section for general instructions on how to approach the task of error token localization.
- Refer to the "Dataset Specific Instructions" section for any additional instructions that are specific to this dataset.
- Finally, adhere to the "On the Output Format" section to ensure your response is structured correctly.
- Note that the dataset specific sections \ (i.e., "Dataset Description" and "Dataset Specific Instructions") can be missing for some datasets. In such cases, simply skip those sections and proceed with the rest of the instructions.

Dataset Description

Gaia2 is a benchmark dataset for evaluating AI agents inside simulated, dynamic environments provided by the Meta Agents Research Environments \ (ARE\). It contains time-evolving scenarios with apps/tools \ (e.g., messaging, calendar, filesystem, shopping, city, contacts\) and multi-agent interactions. Agents must plan, search, and act via tool/API calls under temporal constraints, dynamic events, and noise. The dataset evaluates dimensions such as Execution, Search, Adaptability, Time, Ambiguity, and multi-agent collaboration. For details, see the dataset card and ARE documentation on Gaia2.

On the Input Format

- The input will be provided in the following JSON format:
- ```
{
 "prompt": "<The prompt text here>",
 "completion": "<The model-generated completion text here>",
 "indexed_completion": "<The model-generated completion text with each token indexed, i.e., `token_1[index_1]token_2[index_2]`",
 "ground_truth": "<The ground truth
```

```

 answer text here\>" \ (This field
 may be absent in some cases\))
}
- An example of the input format is as
 follows:
{
 "prompt": "<|im_start|>user
 What is the capital of France?<|
 im_end|>
 <|im_start|>assistant
 ",
 "completion": "<think> okay, let me
 see. The capital of France is
 Lyon. </think>
 \Lyon.",
 "indexed_completion": "<think>[0]
 okay,[1] let[2] me[3] see.[4]
 The[5] capital[6] of[7] France
 [8] is[9] Lyon.[10] </think>[11]
 [12] Lyon[13] .[14]",
 "ground_truth": "Paris, the capital
 of France."
}
- The "indexed_completion" field
 provides a tokenized version of the
 completion, where each token is
 followed by its index in square
 brackets. This will help you
 localize the position of tokens in
 the completion.
- Both the "prompt" and "completion"
 fields may contain special tokens,
 e.g., "<|im_start|>", "<|im_end|>",
 "<think>", "</think>", etc. These
 tokens are part of the model's
 output format, specifically,
- "<|im_start|>" and "<|im_end|>"
 denote the start and end of a
 message in a multi-turn
 conversation, e.g., "<|im_start|>
 user
 xxx <|im_end|>" indicates the start
 of a user message, while "<|
 im_start|>assistant
 xxx <|im_end|>" indicates the start
 of an assistant message.
- "<think>" and "</think>" denote the
 start and end of the model's
 internal reasoning process.
- More on the internal thought process
 of the model:
- The internal thought process of the
 model is represented by the
 reasoning tokens between the "<
 think>" and "</think>" tokens.
- This part reflects the model's
 reasoning steps before arriving at
 the final answer.
- The final answer to the prompt is the
 text that comes AFTER the "</think
 >" token.
General Guidance (Dataset Agnostic)
- Refer to the following steps for
 localizing an erroneous token in
 the model-generated completion:
- Step 1: Carefully read dataset
 description and the provided
 prompt to understand the context
 and requirements.

```

- Step 2: If "completion" and "indexed\_completion" fields contain reasoning tokens (i.e., tokens between "<think>" and "</think>"), separate the reasoning part from the final answer part.
- Step 3: Examine the final answer part of the completion first to identify any errors. If no errors are found in this part, skip the rest of the steps and output "<CORRECT>" as the final answer.
- Step 4: No matter whether the final answer part contains errors, proceed to examine the reasoning part for potential errors.
- Step 5: Once an erroneous token is identified, use the "indexed\_completion" field to find its index.
- Additional strict requirements for localizing the erroneous token:
  - Always report the FIRST erroneous token in the "completion" or "indexed\_completion" field.
  - The localized erroneous token MUST NOT be the special tokens used for formatting (e.g., "<|im\_start|>", "<|im\_end|>", "<think>", "</think>").
  - NEVER treat a token as erroneous just because it is a subword or punctuation mark, as the indexed token str does NOT have to be a single word; it can be a subword or punctuation mark as per the tokenization used by the model.
- On the difference between examining the final answer part and the reasoning part:
  - For the final answer part (i.e., the part AFTER "</think>"):
    - ALWAYS examine whether the tokens in this part fulfill the format/style requirements specified in the prompt. If you identify a token in this part that violates the format/style requirements, you SHOULD consider it as a potential erroneous token.
    - ALSO examine the factual correctness and logical consistency of the tokens in this part. If you identify a token in this part that is factually incorrect or logically inconsistent with the prompt, you SHOULD consider it as a potential erroneous token.
  - For the reasoning part (i.e., the part BETWEEN "<think>" and "</think>"):
    - NEVER examine whether the reasoning tokens have fulfilled the format/style requirements specified in the prompt, as the reasoning tokens between "<think>" and "</think>" are often in a free-form text format. For example, if the prompt requires

the model to "respond in a poem format", you MUST NOT consider a token in the reasoning part as erroneous just because the reasoning tokens are not in a poem format.

- Focus on the factual correctness and logical consistency of the reasoning tokens. If you identify a token in this part that is factually incorrect or logically inconsistent with the prompt, you SHOULD consider it as a potential erroneous token.

# Dataset Specific Instructions

- This dataset uses the ARE simulation where agent messages may trigger tool/API calls and receive environment feedback.
- Messages labeled as tool outputs (e.g., segments marked as 'role=tool-response' with content being tool results) represent environment feedback and MUST NOT be selected as erroneous tokens. Only tokens authored by the assistant are eligible.
- Your goal is to localize the FIRST token in the assistant completion that commits the agent to a course of action that makes the user request unachievable in the scenario context.
- "Unachievable" includes, but is not limited to:
  - Invoking a wrong or unavailable tool/app (e.g., writing to filesystem via 'create\_\_fs' when exploration to locate a folder is required first).
  - Assuming nonexistent resources or paths (e.g., hallucinated file/folder IDs or directories).
  - Attempting UI interactions that the environment does not support (e.g., instructing to "open a user interface" when no such UI exists or is permitted).
  - Skipping required discovery steps (e.g., failing to locate a folder before listing its contents) and immediately issuing a write/irreversible action.
- When such an action spans multiple indexed tokens, localize the very first token that begins the erroneous action or argument (e.g., the first token that starts the wrong tool name or the hallucinated path string).
- Tie-break rules:
  - If both planning text and an ensuing call appear, choose the first token that commits execution (e.g., the token that starts the tool name or the imperative "call"), not earlier speculative text.
  - If multiple independent errors appear, always select the earliest by

index.

- IMPORTANT: Ignore input format issues for tools (e.g., minor JSON/schema/quoting/whitespace mistakes) unless they change semantics. Such surface-form errors should not be selected as the FIRST error if the underlying chosen action is already incorrect.
- Likewise, ignore any content in 'role=tool-response' segments; they are environment outputs, not assistant errors.
- If no erroneous behavior is found and the answer satisfies the prompt and scenario constraints, output "<CORRECT>".

# On the Output Format

- The output MUST be in the following JSON format:
 

```
{
 "error_token": "<The localized erroneous token>",
 "token_index": <The index of the erroneous token in the completion>,
 "explanation": "<A brief explanation of why this token is considered erroneous. Less than 50 words.>"
}
```
- An example of the output format is as follows:
 

```
{
 "error_token": "Lyon",
 "token_index": 10,
 "explanation": "The token 'Lyon' is erroneous because the correct capital of France is Paris, not Lyon. This indicates a factual error in the model's completion."
}
```
- Ensure that the "token\_index" corresponds to the index provided in the "indexed\_completion" field.
- The "explanation" should be concise yet informative, providing enough context to justify the localization of the error token.

## E.2 Target Token Agreement Statistics

Here we report quantitative evidence for the reliability of our target token identification procedure described in the main text. We employ a two-stage pipeline: (1) four independent LLMs each propose the target token, and (2) human annotators validate cases with majority ( $\geq 2$ ) agreement.

**Inter-agreement among LLM Proposers.** We use four independent validators: DeepSeek-V3.2 (DeepSeek-AI et al., 2025), Kimi-K2-Thinking (Team et al., 2025), GPT-5.2 (OpenAI, 2025b), and GPT-OSS-120b (OpenAI, 2025a). Table 5 reports the fraction of samples reaching  $\geq 2$ -validator consensus within  $\pm \delta$  tokens, alongside

the chance baseline for each benchmark. Table 5 shows that agreement rates are far above the chance baseline on IFEval, MATH, and EvalPlus (GAIA2 is excluded from this validation due to its long-context agent traces), indicating strong consistency among independent LLM proposers.

Table 5: LLM inter-validator agreement on target token identification. “Random” denotes the chance baseline; “Exact ( $\pm 0$ )” and “Relaxed ( $\pm 3$ )” denote the fraction of samples with  $\geq 2$ -validator consensus within the respective token window.

| Metric              | EvalPlus | IFEval | MATH  |
|---------------------|----------|--------|-------|
| Random              | 7.1%     | 7.7%   | 1.2%  |
| Exact ( $\pm 0$ )   | 57.4%    | 74.3%  | 56.0% |
| Relaxed ( $\pm 3$ ) | 86.7%    | 89.2%  | 88.0% |

**Human validation.** We further conduct human validation on the LLM majority-agreed cases. Table 6 reports the fraction of cases approved by human annotators under  $\pm 3$ -token matching. The high approval rates confirm that LLM consensus aligns well with human judgment, validating the reliability of our semi-automated target token identification pipeline.

Table 6: Human approval rates for LLM majority-agreed target token identifications.

| Benchmark | Exact-agreed | Relaxed-agreed |
|-----------|--------------|----------------|
| EvalPlus  | 80.0%        | 97.3%          |
| IFEval    | 96.4%        | 90.1%          |
| MATH      | 78.6%        | 86.4%          |

### E.3 Contrast Token Identification

To analyze the model’s preference mechanism, we pair the target token with a contrast token, a plausible alternative that would have led to a more correct continuation. The selection follows the following steps:

1. Assuming the target token is the model’s top-1 output token at the failure position (i.e., the token with the highest logit).
2. Examining the model’s top-10 predicted tokens at that position. If any of these tokens, when substituted, would plausibly lead to a more correct continuation, we select the highest-ranked such token as the contrast token.
3. If no suitable contrast token is found in the top-10 predictions, we query a stronger sibling model (e.g., Qwen3-1.7B for Qwen3-0.6B, or Qwen3-8B for Qwen3-4B) and inspect its top-10 predictions at the same position. If a valid contrast token is found, we again select the highest-ranked one.
4. If neither model yields a suitable candidate, we manually infer a semantically appropriate contrast token. In cases where multiple plausible candidates exist, we choose the one with the highest rank in the original model’s output logits.

### E.4 Impact of Contrast Token Selection and Plausible Alternatives

**Filtering breakdown.** The strict requirement on contrast tokens ensures high-confidence contrastive pairs for attribution, but it also constitutes the primary source of case filtering. Table 7 summarizes the filtering breakdown across processing stages. The dominant filtering factor is the inability to identify a valid contrast token under the strict recovery criterion, rather than ambiguity in target-token identification or failure to meet the logit threshold.

**Dataset extension.** The attribution-analysis collection summarized in Table 1 was assembled in two annotation passes: an initial pass yielding 71 clean cases across the four benchmarks, followed by an extension that produced the final clean collection reported in Table 1.

**Stability across the extension.** During the extension, we ran chi-squared tests of homogeneity comparing the initial batch with the extended annotations; all tests yielded  $p > 0.05$ , and the dominant attribution outcome (M-IA) and failure pattern (URT) remained stable across all benchmarks. This indicates that the reported population-level statistics did not show a significant shift in this sample.

To investigate whether we could relax such strict requirement on contrast token selection, we constructed three additional contrast-token pairs per case on a random subset of 12 failure cases; on average, **7.51 out of 10** top positively contributing tokens overlap with those from the baseline pair, indicating that qualitative conclusions are stable across plausible alternatives.

This suggests that the strict recovery requirement could, in principle, be relaxed to admit alternative contrast tokens, even when the model cannot fully

Table 7: Filtering breakdown by stage (% of original failure cases excluded at each stage).

| Stage                                | IFEval | MATH  | EvalPlus | GAIA2 |
|--------------------------------------|--------|-------|----------|-------|
| Unbounded generation                 | 7.0%   | 1.6%  | 0%       | 9%    |
| <b>Unable to identify token-pair</b> | 18.4%  | 24.2% | 27.8%    | 31%   |
| Logit-diff threshold                 | 13.9%  | 14.5% | 21.9%    | 20%   |

recover due to limited capability. However, we adopt the stricter criterion to ensure clean, high-confidence contrastive pairs, and leave the incorporation of more flexible alternatives to future work.

### E.5 Robustness to Contrast-Pair Recovery

We ablate the recovery mechanisms in the identification procedure above using three progressively more restrictive configurations. *Full* uses all four steps; *w/o Step 4* removes manual identification of a semantically appropriate contrast token; and *w/o Steps 3–4* additionally removes the stronger-sibling-model search, leaving only the target and base-candidate search in Steps 1–2. The filtering rate is the share of collected failures removed at token-pair identification. The rates in Table 8 refer to the Qwen3-0.6B original failure pool.

Because the coverage changes are largest for MATH and GAIA2, we report their attribution outcomes and failure patterns under all three configurations in Table 9 and Table 10. These distributions are computed on Qwen3-0.6B configuration-specific retained cases; they should not be read as paired comparisons across configurations.

The recovery mechanisms change coverage most for MATH and GAIA2. Within the tested variants, M-IA remains the largest attribution outcome and URT remains the largest failure pattern for both datasets.

## F Post-Training Recipe Analysis

We compare the SFT, DPO, and RL stages of the OLMo3-7B-Thinking and OLMo3-7B-Instruct paths on 24 IFEval failures shared across all six checkpoints. In Table 11, each table entry is a reported stage-level result for this fixed shared set. “Logit difference / Total” denotes the target-token logit minus the contrast-token logit, followed by the normalized relevance summed over the Instruction, Query, and Answer segments. More negative values indicate a stronger correction of the original failure decision.

In Figure 9, the Thinking path changes gradually across DPO and RL, whereas the Instruct path

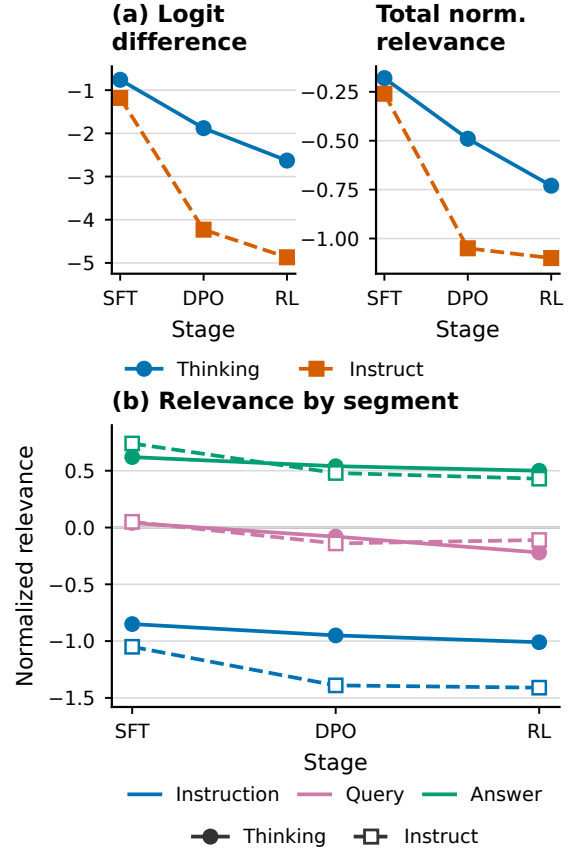


Figure 9: Post-training recipe summary for OLMo3-7B-Thinking and OLMo3-7B-Instruct on the fixed 24-case shared IFEval set. The plotted points are the reported stage-level results for this fixed shared set. The top panel shows the target-minus-contrast logit difference and total normalized relevance, where Total is the normalized relevance summed over Instruction, Query, and Answer segments. The bottom panel shows Instruction / Query / Answer normalized relevance.

Table 8: Token-pair filtering rates under progressively more restrictive contrast-pair recovery configurations.

| Dataset  | Full  | w/o Step 4 | w/o Steps 3–4 |
|----------|-------|------------|---------------|
| IFEval   | 18.4% | 19.4%      | 20.4%         |
| MATH     | 24.2% | 27.4%      | 32.3%         |
| EvalPlus | 27.8% | 28.1%      | 28.1%         |
| GAIA2    | 31.0% | 43.0%      | 43.0%         |

Table 9: Attribution-outcome distributions for the contrast-pair recovery ablation.

| Dataset | Configuration | M-IA  | NC-IA + M-AG | NC-IA + AG |
|---------|---------------|-------|--------------|------------|
| MATH    | Full          | 43.2% | 21.6%        | 35.1%      |
|         | w/o Step 4    | 45.7% | 20.0%        | 34.3%      |
|         | w/o Steps 3–4 | 46.9% | 21.9%        | 31.2%      |
| GAIA2   | Full          | 80.0% | 15.0%        | 5.0%       |
|         | w/o Step 4    | 85.7% | 10.7%        | 3.6%       |
|         | w/o Steps 3–4 | 85.7% | 10.7%        | 3.6%       |

shows its largest shift at DPO and only a modest additional change at RL. Fixing both paths at 7B partially controls for model-size confounding, but it does not disentangle differences in training data, objective, or the broader post-training recipe.

Figure 8 and Figure 6 provide the denser Thinking-path trajectory that complements the aligned six-checkpoint comparison in Figure 9. They trace the fixed Thinking failure set across all intermediate checkpoints, rather than only the matched SFT, DPO, and RL stages summarized here.

## G Inter-Annotator Agreement Statistics on Attribution Analysis

We also report inter-annotator agreement (IAA) statistics for the two key annotation stages in our analysis: (1) the classification of attribution outcomes (M-IA, NC-IA + M-AG, NC-IA + AG) and (2) the classification of failure patterns (URT, OIT, URT + OIT). Table 12 reports agreement rates and Wilson 95% confidence intervals for attribution outcome classification, and Table 13 reports agreement rates for failure pattern classification (among cases categorized as M-IA or NC-IA + M-AG). The IAA measurement follows a three-phase protocol:

- **Pre-calibration (IFEval, MATH, EvalPlus).** Two annotators independently labeled a subset of cases per benchmark to establish baseline agreement and surface ambiguous category boundaries; the resulting rates appear in the first three rows of each table. GAIA2 is excluded from IAA measurement because its long agentic prompts (10k+ tokens) make double annotation prohibitively costly.

- **Calibration discussion.** The two annotators jointly reviewed disagreements, aligned category definitions (notably the M-IA / NC-IA + M-AG boundary and the URT / OIT distinction), and produced a refined annotation guideline.

- **Post-calibration spot check (EvalPlus only).** To verify that the refined protocol reduces ambiguity, both annotators independently re-labeled a fresh batch of EvalPlus cases (disjoint from the pre-calibration batch). EvalPlus was selected because it exhibited the lowest agreement during pre-calibration. Agreement rose to 88.2% for attribution outcomes and 83.3% for failure patterns.

The refined annotation protocol was then applied by a single annotator to all four benchmarks (including GAIA2) to produce the final labels used in the main text.

## H Comparison with Alternative Attribution Methods

This appendix provides a direct comparison between AttnLRP (the attribution method adopted in this work) and two representative baselines: an alternative LRP-based method (CP-LRP (Ali et al., 2022)) and a standard gradient-based method (Gradient (Simonyan et al., 2014)). Our goal is not to introduce a new attribution method, but to evaluate whether AttnLRP provides faithful attributions suitable for failure diagnosis and whether the diagnostic framework is compatible with alternative faithful attribution methods.

**Perturbation-based evaluation.** We evaluate all three methods using a perturbation-based protocol on 60 randomly selected failure cases across

Table 10: Failure-pattern distributions for the contrast-pair recovery ablation.

| Dataset | Configuration | URT   | OIT   | URT + OIT |
|---------|---------------|-------|-------|-----------|
| MATH    | Full          | 50.0% | 29.2% | 20.8%     |
|         | w/o Step 4    | 52.2% | 26.1% | 21.7%     |
|         | w/o Steps 3–4 | 50.0% | 27.3% | 22.7%     |
| GAIA2   | Full          | 78.9% | 13.2% | 7.9%      |
|         | w/o Step 4    | 81.5% | 11.1% | 7.4%      |
|         | w/o Steps 3–4 | 81.5% | 11.1% | 7.4%      |

Table 11: Post-training-stage metrics for OLMo3-7B on 24 shared IFEval failures.

| Path     | Metric                       | SFT             | DPO                | RL                 |
|----------|------------------------------|-----------------|--------------------|--------------------|
| Thinking | Logit difference / Total     | −0.76/ − 0.18   | −1.88/ − 0.49      | −2.63/ − 0.73      |
|          | Instruction / Query / Answer | −0.85/0.04/0.62 | −0.95/ − 0.08/0.54 | −1.01/ − 0.22/0.50 |
| Instruct | Logit difference / Total     | −1.18/ − 0.26   | −4.23/ − 1.05      | −4.87/ − 1.10      |
|          | Instruction / Query / Answer | −1.05/0.05/0.74 | −1.39/ − 0.14/0.48 | −1.41/ − 0.11/0.43 |

Table 12: Inter-annotator agreement on attribution outcome classification.

| Phase                   | Dataset         | <i>N</i>  | Agreement    | Wilson 95% CI         |
|-------------------------|-----------------|-----------|--------------|-----------------------|
| Pre-calibration         | IFEval          | 21        | 71.4%        | [50.0%, 86.2%]        |
| Pre-calibration         | MATH            | 16        | 50.0%        | [28.0%, 72.0%]        |
| Pre-calibration         | EvalPlus        | 19        | 47.4%        | [27.3%, 68.3%]        |
| <b>Post-calibration</b> | <b>EvalPlus</b> | <b>34</b> | <b>88.2%</b> | <b>[73.4%, 95.3%]</b> |

Table 13: Inter-annotator agreement on failure pattern classification.

| Phase                   | Dataset         | <i>N</i>  | Agreement    | Wilson 95% CI         |
|-------------------------|-----------------|-----------|--------------|-----------------------|
| Pre-calibration         | IFEval          | 18        | 38.9%        | [20.3%, 61.4%]        |
| Pre-calibration         | MATH            | 5         | 40.0%        | [11.8%, 76.9%]        |
| Pre-calibration         | EvalPlus        | 12        | 58.3%        | [32.0%, 80.7%]        |
| <b>Post-calibration</b> | <b>EvalPlus</b> | <b>24</b> | <b>83.3%</b> | <b>[64.1%, 93.3%]</b> |

all four datasets (IFEval, EvalPlus, MATH, and GAIA2). For each case, we rank input tokens by attribution score and measure how many top-ranked tokens must be masked to flip the model’s top-1 prediction away from the error token. Table 14 reports the average number of tokens required and the overall fix rate.

Table 14: Perturbation-based evaluation of attribution methods across 60 failure cases.

| Method   | Avg. tokens to fix | Fix rate |
|----------|--------------------|----------|
| AttnLRP  | 1.7                | 100%     |
| CP-LRP   | 2.0                | 100%     |
| Gradient | 3.7                | 96.7%    |

Both LRP-based methods achieve a 100% fix rate, whereas the gradient baseline still fails in some cases. Between the two LRP methods, the difference is marginal (1.7 vs. 2.0 tokens), consistent with the original AttnLRP paper where the faithfulness gap between AttnLRP and CP-LRP is also small.

**Attribution sharpness.** Beyond perturbation accuracy, we measure how concentrated each method’s attribution is on the most relevant tokens. We compute two metrics: (1) the *concentration ratio*, defined as the fraction of total attribution mass captured by the top-10 content tokens (excluding special tokens such as `<|im_start|>` and `<think>`), and (2) the *Gini coefficient*, measuring the overall inequality of the attribution distribution (1.0 = all mass in one token, 0.0 = uniform). Results are reported in Table 15 and Table 16, respectively.

Table 15: Concentration ratio (top-10 content tokens).

| Method   | Mean   | Median | Std    |
|----------|--------|--------|--------|
| AttnLRP  | 0.5176 | 0.4889 | 0.1611 |
| CP-LRP   | 0.5111 | 0.4738 | 0.1916 |
| Gradient | 0.4428 | 0.4198 | 0.1669 |

Table 16: Gini coefficient of attribution distributions.

| Method   | Mean   | Median | Std    |
|----------|--------|--------|--------|
| AttnLRP  | 0.7326 | 0.7398 | 0.0664 |
| CP-LRP   | 0.7061 | 0.7262 | 0.1046 |
| Gradient | 0.6817 | 0.6798 | 0.0629 |

As shown in Table 15 and Table 16, AttnLRP achieves the highest mean and median on both metrics, indicating a marginally sharper attribution

signal. It also exhibits the lowest Gini standard deviation (0.066 vs. 0.105 for CP-LRP), suggesting more stable attribution sharpness across cases. Both LRP methods substantially outperform Gradient, whose more diffuse attributions are consistent with its lower perturbation efficiency.

**Qualitative comparison.** To illustrate the qualitative differences, consider an IFEval case in which the model is asked to produce a Japan itinerary in Shakespearean style without using commas:

*“I am planning a trip to Japan, and I would like thee to write an itinerary for my journey in a Shakespearean style. You are not allowed to use any commas in your response.”*

The model produces an incorrect output by inserting a comma after “halls”:

*“In the land of thy golden halls,”*

We analyze input attributions for the decision between the target token “,” and an alternative contrast token “and”. Gradient assigns the strongest signals to largely irrelevant content/style tokens (e.g., “Japan”, “Shakespearean”). In contrast, AttnLRP and CP-LRP assign higher importance to constraint-related tokens (e.g., “not allowed”, “commas”), with AttnLRP exhibiting a sharper and more localized focus. This suggests that gradient-based attributions are less faithful in this setting, whereas LRP-based methods better capture the governing constraint, with AttnLRP providing a more precise signal for debugging and improvement.

Overall, we acknowledge that AttnLRP and CP-LRP perform comparably in this diagnostic setting, and either could serve as the underlying attribution method. Our choice of AttnLRP is motivated by its slightly sharper signal and stronger performance reported in the original paper. Our contribution lies in the diagnostic framework rather than the specific attribution method, and the framework is compatible with any faithful attribution method.

## I Sample Expanded Attribution Graphs

For clarity of exposition, the attribution graph shown in the main text is a reduced representation that includes only the dominant layers, nodes, and relevance pathways. Minor layers and low-magnitude edges are omitted to avoid visual clutter. The Figure 10 provides an expanded version of the graph that preserves a larger portion of the model’s internal relevance propagation. While the full graph still prunes extremely low-contribution edges for readability, it offers a more exhaustive

view of the attribution structure used in our analysis.

In addition to the representative cases discussed above, we also observe another class of attribution graph patterns in the MATH dataset, as shown in Figure 11. It further illuminates the model’s internal decision dynamics. Specifically, the full attribution graph for this example reveals that the model’s preference for the target token is driven overwhelmingly by internal biases, most notably the BOS token, while receiving only minimal contribution from the reasoning chain present in the prompt. The relevance mass rapidly concentrates along a small set of bias-dominated pathways, bypassing the majority of intermediate reasoning tokens. This structure indicates that the model’s erroneous decision does not arise from misinterpreting the logical steps in the prompt, but rather from an inherent predisposition encoded in deep layers and positional biases, which suppress the semantic evidence that should have been propagated along the reasoning chain.

## J Statistical Analysis of Attribution Graph Structure

This appendix presents an exploratory statistical analysis of the attribution graphs constructed for the failure cases studied in the main text. We examine the layer-wise relevance structure of the final (prediction) token, decompose its attribution into distinct source components, and investigate whether these structural features give rise to clearly separable failure categories. The analysis covers the initial non-agentic graph-analysis subset of 57 clean traces (21 from IFEval, 19 from EvalPlus, 17 from MATH) for input-level relevance profiling, of which 23 are categorized as NC-IA and undergo detailed attribution graph analysis (Sections K.1–K.4). Cross-model comparisons use Qwen3-0.6B, 1.7B, 4B, and 8B on the same failure cases (Section K.5). This subset corresponds to the initial annotation pass on non-agentic benchmarks; the stability analysis in subsection E.4 indicates that the structural patterns reported below extend to the full clean collection summarized in Table 1. As we show below, while certain aggregate-level regularities emerge robustly (e.g., layer-wise functional specialization and consistent critical-layer locations), the data do not exhibit cleanly separable clusters along any single structural dimension, highlighting the challenge of automated failure categorization from attribution

graph structure alone.

### J.1 Relevance Profile Analysis

For each failure case, we extract the scalar relevance  $R_i^{(l)}$  of the last token (i.e., the token position at which the next-token prediction is made) across all  $L$  transformer layers, yielding a *relevance profile* vector of dimension  $L + 1$  (including the embedding layer). For Qwen3-0.6B, the raw relevance at the embedding layer averages 0.66 and grows monotonically to 18.85 at the final layer (L27), confirming that the contrastive logit difference is progressively amplified across the transformer stack. To enable comparison across cases with different logit-difference magnitudes, we normalize each profile by subtracting the embedding-layer value and dividing by the absolute value of the final-layer relevance:

$$\hat{R}^{(l)} = \frac{R^{(l)} - R^{(-1)}}{|R^{(L-1)}|}, \quad (23)$$

where  $R^{(-1)}$  and  $R^{(L-1)}$  denote the embedding-layer and final-layer relevance, respectively. The resulting normalized profiles reveal how relevance accumulates across the transformer stack for different failure cases.

Figure 12 visualizes the normalized relevance profiles for all failure cases across the three benchmarks (IFEval, MATH, and EvalPlus). While all profiles share a monotonically increasing trend by construction, distinct trajectory shapes emerge: some cases exhibit rapid early growth followed by saturation, whereas others show a delayed, late-layer surge. To investigate whether these shape differences correspond to meaningful failure categories, we perform an exploratory clustering analysis.

#### Exploratory clustering of relevance profiles.

Since relevance profile analysis and attribution decomposition require attribution graph construction, which is not performed for cases already explained by input attribution (M-IA), we restrict the following analyses to the 23 failure traces categorized as NC-IA (with or without subsequent attribution graph explanation). We apply  $k$ -means clustering to the normalized relevance profile vectors (each 29-dimensional), selecting  $k=3$  via the elbow curvature and silhouette score. The resulting clusters (Figure 13) suggest three broad accumulation patterns: (i) gradual near-linear growth (Cluster 0,

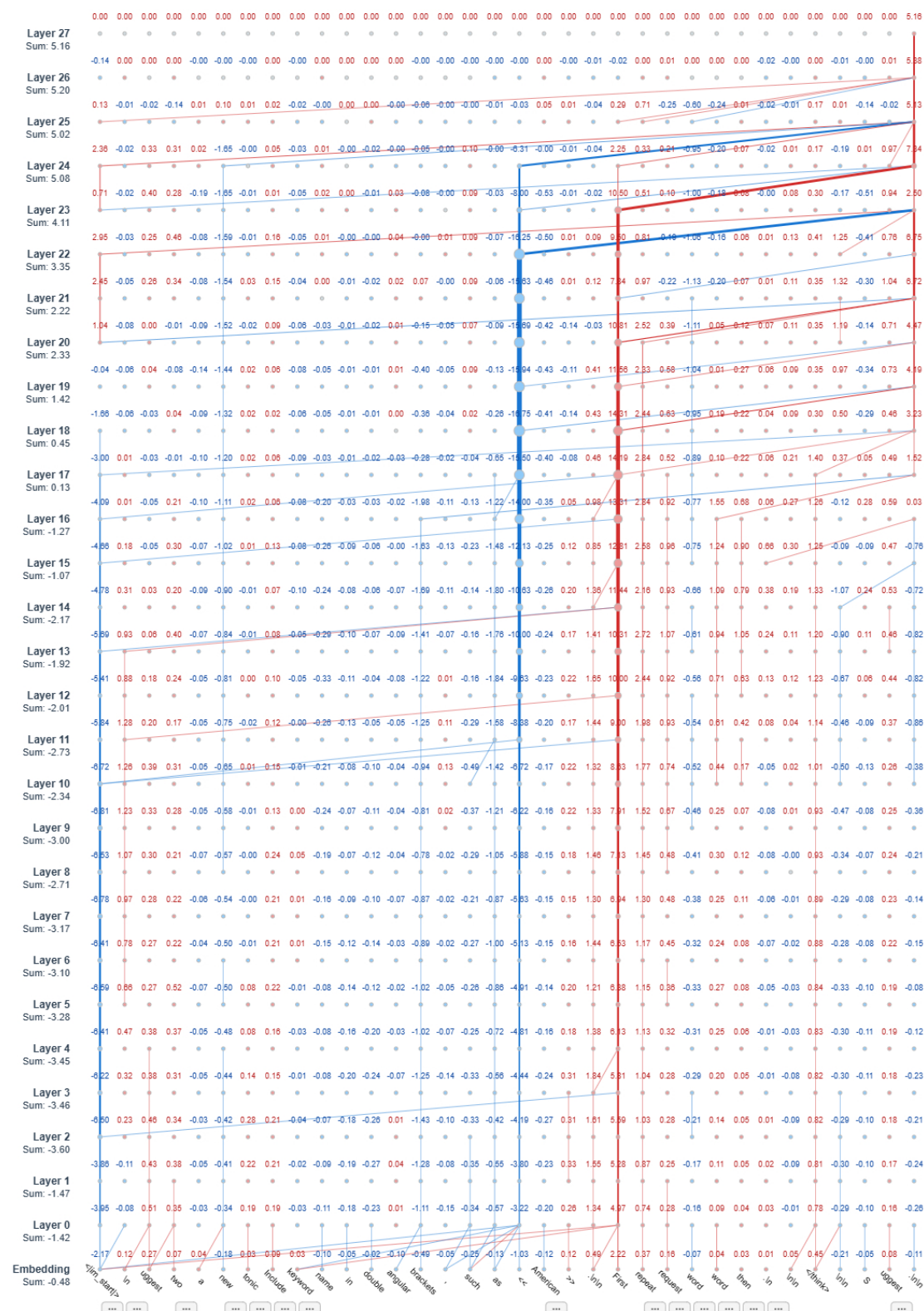


Figure 10: Expanded attribution graph for case in Figure 3.



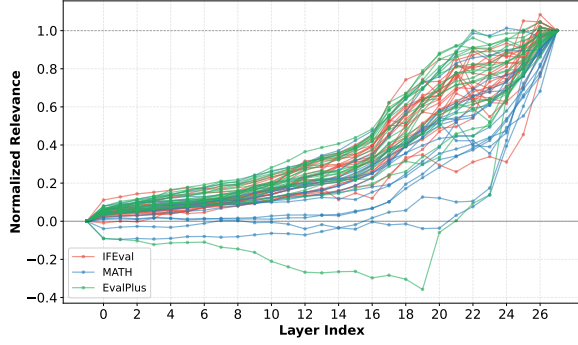


Figure 12: Normalized relevance profiles of the prediction token across all failure cases, colored by dataset. Each curve shows how relevance accumulates from the embedding layer (L-1) to the final layer (L27).

$n=13$ ), (ii) early plateau followed by late-layer acceleration (Cluster 1,  $n=3$ ), and (iii) rapid early growth with gentle late-layer progression (Cluster 2,  $n=7$ ). A PCA projection shows reasonable separation in the first two principal components (Figure 14), though as we show in the next subsection, this separation does not extend to other structural dimensions. We also caution that Cluster 1 contains only 3 traces, limiting the statistical reliability of any conclusions drawn from it; the clustering should therefore be interpreted as descriptive rather than as evidence of discrete failure types.

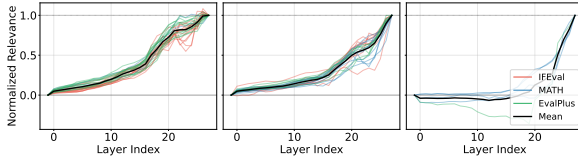


Figure 13: Clustered relevance profiles ( $k=3$ ). Each panel shows individual traces (thin lines) and the cluster mean (thick line).

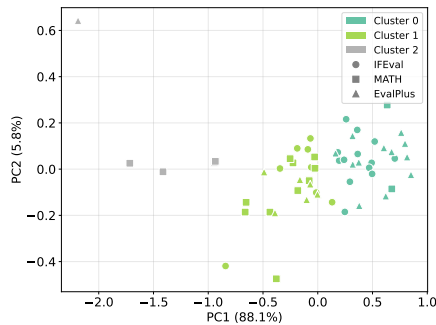


Figure 14: PCA 2D projection of the 29-dimensional normalized relevance profiles, colored by  $k$ -means cluster assignment.

## J.2 Attribution Decomposition

To understand the sources of relevance at the prediction token, we decompose the attribution at each layer into three components using the edge structure of the attribution graph:

- **Self Bias (SB):** The residual relevance at the prediction token not accounted for by any incoming edges, i.e., the difference between the node relevance and the total incoming edge weight. This captures the model’s internal bias at that position.
- **BOS Contribution (BOS):** The total edge weight from the beginning-of-sequence token (position 0) to the prediction token. This captures the influence of the attention sink (Xiao et al., 2024).
- **Other-Token Contribution (OC):** The total edge weight from all other tokens (excluding BOS and self-loops) to the prediction token. This captures the influence of context tokens.

Formally, for the prediction token at position  $n$  and layer  $l$ :

$$R_n^{(l)} = \underbrace{\text{SB}^{(l)}}_{\text{self bias}} + \underbrace{\text{BOS}^{(l)}}_{\text{BOS contribution}} + \underbrace{\text{OC}^{(l)}}_{\text{other contribution}}, \quad (24)$$

where  $\text{BOS}^{(l)} = \sum_{e \in \mathcal{E}_{\text{BOS} \rightarrow n}^{(l)}} w_e$ ,  $\text{OC}^{(l)} = \sum_{e \in \mathcal{E}_{\text{other} \rightarrow n}^{(l)}} w_e$ , and  $\text{SB}^{(l)} = R_n^{(l)} - \text{BOS}^{(l)} - \text{OC}^{(l)}$ .

To obtain scale-invariant composition features, we normalize each component by the mean absolute total relevance  $\overline{|R|} = \frac{1}{L+1} \sum_l |R_n^{(l)}|$  and define the composition proportions:

$$\begin{aligned} \text{SB fraction} &= \frac{\overline{\text{SB}}}{\overline{\text{SB}} + \overline{\text{OC}}} \\ \text{BOS fraction} &= \frac{|\overline{\text{BOS}}|}{\overline{\text{SB}} + \overline{\text{OC}}} \end{aligned} \quad (25)$$

where  $\overline{(\cdot)}$  denotes the layer-wise mean of the normalized values. The SB fraction measures the relative dominance of self bias over context integration, while the BOS fraction measures the relative strength of the attention-sink effect.

Figure 15 shows the composition space as a scatter plot of SB fraction versus total magnitude, colored by profile cluster. The per-cluster mean composition statistics are shown in Table 17.

Table 17: Per-cluster mean composition statistics (mean  $\pm$  std).

| Cluster              | SB/(SB+OC)        | BOS /(SB+OC)      | Magnitude         |
|----------------------|-------------------|-------------------|-------------------|
| Cluster 0 ( $n=13$ ) | $0.516 \pm 0.119$ | $0.239 \pm 0.112$ | $0.164 \pm 0.036$ |
| Cluster 1 ( $n=3$ )  | $0.332 \pm 0.009$ | $0.339 \pm 0.055$ | $0.259 \pm 0.121$ |
| Cluster 2 ( $n=7$ )  | $0.528 \pm 0.108$ | $0.258 \pm 0.098$ | $0.104 \pm 0.017$ |

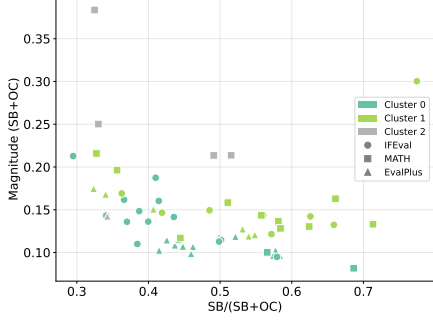


Figure 15: Composition space: SB fraction vs. total magnitude (SB+OC), colored by relevance-profile cluster. Marker shapes indicate datasets.

### Profile shape and composition are orthogonal.

A key finding is that the relevance-profile clusters do *not* align with the composition structure. The intra-cluster variance is  $2.61\times$  and  $8.80\times$  larger than the inter-cluster variance for SB fraction and BOS fraction, respectively, as shown in Table 18. Moreover, direct  $k$ -means clustering on the composition space yields near-zero agreement with profile-based clusters (Adjusted Rand Index =  $-0.004$ ). This orthogonality demonstrates that the attribution graph structure is inherently multi-dimensional: *how* relevance accumulates across layers (profile shape) is largely independent of *where* the relevance originates (SB vs. BOS vs. context). As a consequence, no single clustering scheme can adequately capture the structural diversity of failure cases, underscoring the difficulty of building automated failure taxonomies from attribution graphs at the current scale.

Table 18: Intra-cluster vs. inter-cluster variance ratio for composition features.

| Feature           | Intra- $\sigma^2$ | Inter- $\sigma^2$ | Ratio        |
|-------------------|-------------------|-------------------|--------------|
| SB/(SB+OC)        | 0.0105            | 0.0040            | $2.61\times$ |
| BOS /(SB+OC)      | 0.0093            | 0.0011            | $8.80\times$ |
| Magnitude (SB+OC) | 0.0020            | 0.0022            | $0.90\times$ |

### J.3 Layer-Wise Functional Specialization

In contrast to the clustering analyses above, which reveal high structural heterogeneity across indi-

vidual traces, aggregating composition features by *layer segment* exposes a robust and consistent pattern. We partition the transformer layers into three segments: *Early* (layers  $-1$  to  $8$ ), *Mid* (layers  $9$  to  $18$ ), and *Late* (layers  $19$  to  $27$ ).<sup>3</sup> We then compute the mean composition features within each segment (Figure 16).

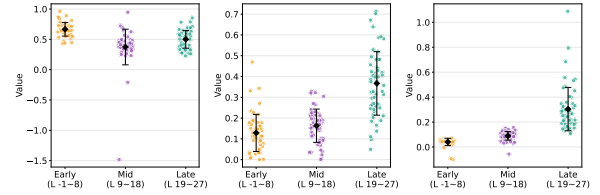


Figure 16: Distribution of composition features by layer segment (Early/Mid/Late). Diamonds denote segment means  $\pm 1$  standard deviation; dots show individual traces.

Unlike the case-level clustering, this aggregate analysis yields clear and consistent findings across all 23 traces:

- **Self bias dominates early layers.** The SB fraction is highest in early layers (mean  $\approx 0.73$ ) and decreases to  $\approx 0.49$ – $0.51$  in mid and late layers, indicating that early-layer computations are primarily driven by position-specific biases rather than contextual information.
- **Context integration peaks in mid layers.** The other-token contribution (OC fraction  $\approx 0.51$ ) is highest in mid layers, indicating that the model’s integration of contextual information is most active in the middle of the transformer stack. This finding is consistent with prior observations that mid-layer attention heads are responsible for semantic composition (Kobayashi et al., 2021).
- **BOS influence grows in later layers.** The BOS fraction increases from early ( $\approx 0.15$ ) to late layers ( $\approx 0.27$ – $0.36$ ), consistent with the attention-sink phenomenon becoming more pronounced in deeper layers.

<sup>3</sup>Layer indices are specific to Qwen3-0.6B (28 transformer layers, indexed L0–L27). For models with different depths, the segments are adjusted proportionally.

This three-phase functional specialization (*early bias establishment, mid-layer context integration, late-layer bias amplification*) is the most robust structural finding in our analysis: it holds across all traces regardless of profile cluster, dataset, or decomposition pattern. It also provides a mechanistic explanation for the main-text findings: failures amenable to input-level attribution (M-IA) correspond to cases where mid-layer context integration goes awry, while failures requiring attribution-graph analysis (NC-IA+M-AG) tend to involve late-layer bias accumulation that overrides contextual signals.

#### J.4 Critical Layer Identification

The second robust aggregate finding concerns the location of *critical transition layers*. We compute first-order differences ( $\Delta$ ) of each decomposition component across adjacent layers and locate the *peak transition layer* for each trace, defined as the layer transition at which the component changes most in absolute value (Figure 17).

Despite the heterogeneity observed in case-level clustering, the peak transition layers are remarkably consistent: median peaks concentrate in layers 20–26 across all clusters (Table 19). For BOS and OC in particular, the peak transitions cluster tightly around layers 23–26 ( $\text{std} \leq 4.0$ ), indicating that the model’s final “decision moment,” i.e., the layer at which the context-versus-bias tradeoff is resolved, is largely invariant to the specific failure case or profile shape.

This consistency has a direct practical implication: targeted interventions for correcting failure-inducing biases (e.g., activation editing or attention head pruning) should focus on the **final 6–8 layers** of the network, where the decisive attribution transitions reliably occur.

#### J.5 Cross-Model Comparison of Attribution Decomposition

Using attribution graphs from multiple Qwen3 model sizes (0.6B, 1.7B, 4B, 8B) computed on the same 23 NC-IA failure cases, we compare how the decomposition structure changes with model scale. For each failure case, we extract the same three components (SB, BOS, OC) and plot their layer-wise trajectories for all four model sizes.

Representative cross-model decomposition comparisons are shown in Figure 18. Taking the MATH case *Number Theory\_8* as an illustrative example,

the layer-averaged decomposition values across model sizes are:

Several consistent patterns emerge across failure cases:

- **Total relevance shifts toward negative values with scale.** For corrected cases, the total relevance shifts from positive (favoring the incorrect token) in the 0.6B model to negative (favoring the correct alternative) in larger models, reflecting the corrected logit difference. For the above example, the 0.6B model has a positive total relevance of 0.693, whereas the 1.7B model already reverses to  $-1.288$ .
- **BOS contribution diminishes with scale.** The absolute BOS contribution generally decreases from 0.6B to 8B ( $0.145 \rightarrow 0.041$  in the above example), suggesting that larger models are less reliant on the attention-sink shortcut.
- **Context integration sign flips for corrected cases.** When a larger model corrects the failure, the OC component typically flips sign (from positive to negative), indicating that the larger model assigns contextual relevance that now supports the correct token rather than the incorrect one.

These cross-model findings complement the input-level attribution scaling analysis in Figure 5 (main text) by revealing that the improvement in larger models is not merely a surface-level token reweighting, but reflects deeper structural changes in how relevance is composed and propagated across the transformer stack.

#### J.6 Summary

This analysis reveals a tension between case-level and aggregate-level structure in attribution graphs. At the case level, the data are inherently multi-dimensional: relevance profile shape and source composition are orthogonal ( $\text{ARI} \approx 0$ ), and no single clustering scheme captures the structural diversity of failure cases. This underscores the challenge of building automated failure taxonomies from attribution graphs at the current scale and with current decomposition granularity. At the aggregate level, however, two robust patterns emerge that are consistent across all traces: (1) a three-phase functional specialization (early bias  $\rightarrow$  mid-layer context integration  $\rightarrow$  late-layer bias amplification), and (2) critical attribution transitions concentrated in the final 6–8 layers. These aggregate regularities



Figure 17: Layer-wise  $|\Delta|$  heatmaps for SB, BOS, and OC across all traces (rows). Black boxes mark the peak transition layer for each trace. Darker colors indicate larger magnitude of change.

Table 19: Median peak transition layer per cluster (layer  $i \rightarrow i + 1$ ).

| Cluster              | $\Delta$ SB peak | $\Delta$ BOS peak | $\Delta$ OC peak |
|----------------------|------------------|-------------------|------------------|
| Cluster 0 ( $n=13$ ) | $26.0 \pm 6.8$   | $26.0 \pm 2.2$    | $24.0 \pm 2.1$   |
| Cluster 1 ( $n=3$ )  | $24.0 \pm 0.0$   | $24.0 \pm 0.0$    | $24.0 \pm 0.9$   |
| Cluster 2 ( $n=7$ )  | $20.0 \pm 10.0$  | $24.0 \pm 2.1$    | $23.0 \pm 4.0$   |

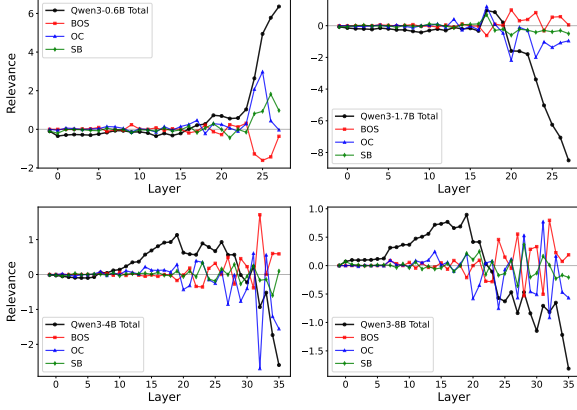


Figure 18: Cross-model attribution decomposition comparison for a representative failure case. Each panel shows the layer-wise decomposition (Total, SB, BOS, OC) for a different Qwen3 model size.

Table 20: Cross-model decomposition for a representative MATH failure case (*Number Theory\_8*). Values are layer-averaged.

| Component      | 0.6B   | 1.7B   | 4B     | 8B     |
|----------------|--------|--------|--------|--------|
| Total rel.     | 0.693  | -1.288 | 0.168  | -0.036 |
| BOS contrib.   | -0.145 | 0.100  | 0.086  | 0.041  |
| Other contrib. | 0.246  | -0.294 | -0.147 | -0.074 |
| Self bias      | 0.118  | -0.097 | -0.008 | -0.016 |

provide actionable guidance for model interventions and connect mechanistically to the behavioral failure patterns identified in the main text.

## K Implications: Monitoring and Alignment Training

We discuss two additional practical directions suggested by our findings, complementing the validated prompt-debugging workflow described in the main text (subsection 4.6, Appendix H).

### Monitoring model behavior during training.

Our checkpoint-wise and model-scaling experiments (Sections 4.3 and 4.4 of the main text) demonstrate that attribution patterns can track how failures evolve during training, serving as a diagnostic tool for model development. Such interpretability-based monitoring can reveal whether failure correction via model scaling or con-

tinued training is due to genuine changes in the underlying reasoning process (e.g., correctly weighting relevant tokens) rather than superficial changes (e.g., memorizing specific token patterns).

**Token-level signals for alignment training.** Attribution scores may provide a natural token-importance signal for alignment training. Instead of treating all tokens in a preferred or dispreferred response equally, one could use attribution to emphasize the tokens that actually drive the failure or preference gap. This suggests a natural integration with token-level preference optimization methods such as DPO (Rafailov et al., 2023). Recent work on Token-Importance Guided Direct Preference Optimization (Yang et al., 2025b) similarly shows that token-importance weighting can improve alignment robustness and stability, though their approach relies on gradient-based token importance, which can be noisy and less faithful. A promising direction is to use more faithful attribution methods, such as LRP-based scores, to derive token-importance signals for alignment training.