Beyond Scores: Understanding LLM-as-a-Judge Mechanisms in Summarization Evaluation

Anonymous ACL submission

Abstract

LLM-based evaluators of natural language generation (NLG) quality are widely deployed as scoring tools and as automated training signals, yet the internal procedure by which they assign a rating remains poorly understood. We investigate this procedure mechanistically through an eight-attack perturbation taxonomy across the Readability and Adequacy dimensions of NLG quality, a generation pipeline producing paired clean/corrupt summaries with controlled error intensity and explicit token-level modification maps, and a four-experiment battery of causal tracing, logit-lens vocabulary projection, and attention-head knockout applied to Themis (Llama-3-8B) and Prometheus (Mistral-7B). Both evaluators implement a structured, coherent evaluation pipeline operating in two stages: below layer 15, attention performs local error comparison and routes the result to the final input position; above it, the MLP cascade integrates the signal and writes the rating, with the decision crystallizing in the residual stream at a sharp late layer ($L = 26$ on Themis, $L = 25$ on Prometheus). The scaffold is preserved across both base architectures, while local-attention recruitment varies systematically by criterion and commitment dynamics at the top of the cascade diverge by base-model family. We release the perturbation taxonomy and generation pipeline as public resources.

1 Introduction

Automatic evaluation using large language models (LLMs) has become widely adopted in natural language generation (NLG): dedicated evaluator models such as Themis (Hu et al., 2024b) and Prometheus (Kim et al., 2024), alongside prompting-based approaches including G-Eval (Liu et al., 2023) and MT-Bench (Zheng et al., 2023), are now routinely applied to tasks ranging from summarization faithfulness to dialogue coherence assessment and increasingly serve as automated signals for preference labeling and reward modeling. Existing examination of these LLM-as-a-judge systems has focused largely on behavioral analysis: measuring agreement with human ratings (Fabbri et al., 2021; Liu et al., 2023), cataloging failure modes such as criterion confusion (Hu et al., 2024a), and quantifying positional and length biases (Zheng et al., 2023). This paradigm characterizes *what* LLM-judges score but leaves *how* the rating is computed inside the model largely unexamined, limiting both evaluator reliability and our ability to anticipate failure cases.

We address this gap through a mechanistic study, asking: when an LLM-judge assigns a Readability score of 3 to a perturbed summary, does the model identify the specific quality defect, or does it produce a rating consistent with training-distribution patterns without engaging the underlying error? We focus on text summarization as a case study because it is one of the most extensively studied NLG evaluation tasks, providing mature perturbation methods and well-established benchmark datasets. Moreover, its structured article-summary format naturally facilitates localized perturbation interventions.

One key challenge in directly applying causal tracing to LLM-judges is that these methods require paired clean and corrupted inputs with precisely tracked token-level modifications and corresponding rating shifts. Existing adversarial summarization benchmarks, however, typically introduce errors at the sentence or document level, making it difficult to trace token-level changes. To address this limitation, we develop a new perturbation taxonomy consisting of eight localized attack types spanning the Readability and Adequacy criteria of NLG quality, along with a generation pipeline that produces paired clean/corrupt summaries with controlled error intensity and explicit token-level modification maps. With this framework, we analyze the internal components of two state-of-the-art

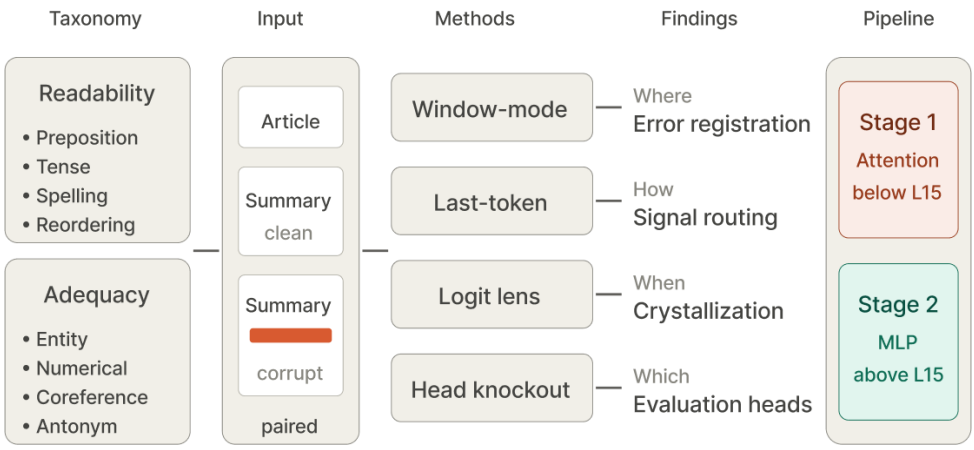


Figure 1: Overview of our analysis. We design an eight-attack **perturbation taxonomy** across the Readability and Adequacy dimensions of NLG quality, and generate paired clean/corrupt **summaries** of an article with controlled error intensity and explicit token-level modification maps. Four **mechanistic methods** - window-mode and last-token causal tracing, logit lens, and attention-head knockout, each reveal a different aspect of how the rating is computed: *where* errors are registered, *how* the signal routes to the output position, *when* the decision crystallizes in the residual stream, and *which* heads implement the verdict.

NLG LLM judges, fine-tuned from two different model families, to uncover how they internally process and assign scores for summary quality. Our key findings include:

1. LLM-based evaluator models exhibit structured, coherent information propagation during evaluation, rather than an ad hoc mapping from surface features to rating tokens. Specifically, the attention module in lower layers (< 15 layer) primarily performs error identification and signal routing, while the upper-layer MLP integrates these routed signals to produce the final rating, with the decision crystallizing in the residual stream at a sharp late layer ($\sim 25 - 26$ layer).

2. The process of error identification in attention varies systematically across error types: readability-related errors spread across adjacent syntactic context, while adequacy attacks remain concentrated on the perturbed tokens. This indicates that LLM-judges deploy distinct attention strategies depending on the criterion, suggesting that criterion-specific attention supervision could improve evaluator robustness.

3. The way the final rating is written into the residual stream differs by base-model family: Themis commits to its rating in a single dominant step at the top layer, while Prometheus spreads this write across mid-to-late layers and applies an opposite-direction correction at the final layer. This indicates that different base-model architectures may implement distinct output-assembly strategies even when arriving at the same final rating.

The present work contributes (i) a perturbation taxonomy and generation pipeline, released as a public resource for further interpretability research on NLG evaluators, and (ii) a structured analysis of the internal mechanisms of LLM-based evaluation across two open-source models drawn from distinct base-model families, providing insights that can inform the design of more robust and interpretable LLM-based evaluators.

2 Related Work

LLM-based NLG evaluation. Dedicated evaluator models (Hu et al., 2024b; Kim et al., 2024) and prompting-based scoring frameworks (Liu et al., 2023; Zheng et al., 2023) have become standard tools for assessing NLG quality, and prior work has examined evaluator failure modes including criterion confusion and prompt sensitivity (Hu et al., 2024a; Zheng et al., 2023). Established factual-consistency and faithfulness benchmarks (Fabbri et al., 2021; Kryściński et al., 2020; Pagnoni et al., 2021; Honovich et al., 2022; Dziri et al., 2022) provide the empirical foundation for evaluator development but remain behavioral in scope, characterizing what LLM-judges score rather than how they arrive at those scores. The present work complements

this literature by examining the evaluator's internal procedure: rather than measuring agreement with human judgments, we analyze how the rating is computed inside the model.

Mechanistic interpretability of transformer language models. A growing body of work localizes computation inside transformer models through activation patching and causal mediation analysis (Vig et al., 2020; Meng et al., 2022), vocabulary-space projection of intermediate residual streams (nostalgebraist, 2020), and circuit-level identification of attention-head structure (Wang et al., 2023). These methods have been applied to factual recall (Meng et al., 2022), indirect-object identification (Wang et al., 2023), and other narrowly-scoped behaviors, but to our knowledge LLM-based NLG evaluation has not previously been the subject of mechanistic analysis. We extend these methods to the evaluator setting, applying them in combination across two evaluator models drawn from distinct base-model families.

Perturbation-based evaluation of NLG systems. Adversarial perturbation has long been used for diagnostic evaluation of NLG and natural-language inference systems, ranging from character-level noise (Belinkov and Bisk, 2017) and behavioral testing (Ribeiro et al., 2020) to factuality-focused perturbations of generated summaries (Kryściński et al., 2020; Pagnoni et al., 2021; Li et al., 2023; Min et al., 2023). These resources have established that perturbations can isolate specific failure modes, but they are designed for behavioral benchmarking and do not preserve the position-level metadata required for mechanistic analysis. Our perturbation taxonomy and generation pipeline build on this prior methodology while introducing the additional controls necessary for mechanistic study.

3 Data Perturbation

Existing adversarial datasets for NLG evaluation introduce errors in a coarse-grained manner such as clause-level insertions, deletions, or unconstrained reordering and typically do not record which tokens were modified or their locations. While sufficient for behavioral benchmarking, this design prevents mechanistic analysis, as causal tracing and attention-knockout interventions require paired clean/corrupt inputs with explicit token-level modification maps to attribute activation changes to specific perturbations.

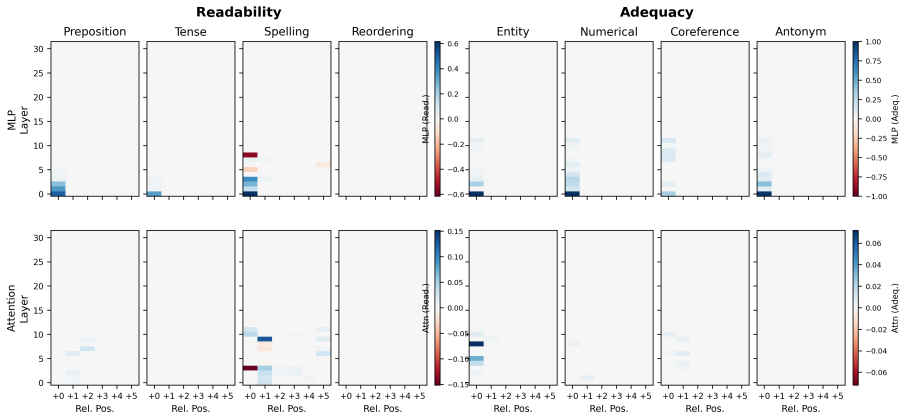
Final Node	Source Perturbations
Readability	
Preposition Mismatch	grammatical error (Pagnoni et al., 2021), subject-verb disagreement (Ribeiro et al., 2020)
Tense Mismatch	incorrect verb form (Pagnoni et al., 2021), grammatical error (Pagnoni et al., 2021)
Spelling	character-level noise (Belinkov and Bisk, 2017)
Sequential Reordering	jumbling word order (Ribeiro et al., 2020), sentence exchange (Kryściński et al., 2020)
Adequacy	
Entity Swap	entity swap (Kryściński et al., 2020), entity error (Pagnoni et al., 2021), perturb names/nouns (Ribeiro et al., 2020)
Numerical/Date Swap	number swap (Kryściński et al., 2020), change numbers (Ribeiro et al., 2020)
Coreference Mismatch	coreference error (Pagnoni et al., 2021), pronoun swap (Kryściński et al., 2020)
Antonym & Negation	sentence negation (Kryściński et al., 2020), negation/antonyms (Ribeiro et al., 2020)

Table 1: Consolidation mapping showing how perturbation types from prior NLG-evaluation benchmarks are unified into the eight nodes of our taxonomy across the Readability and Adequacy dimensions.

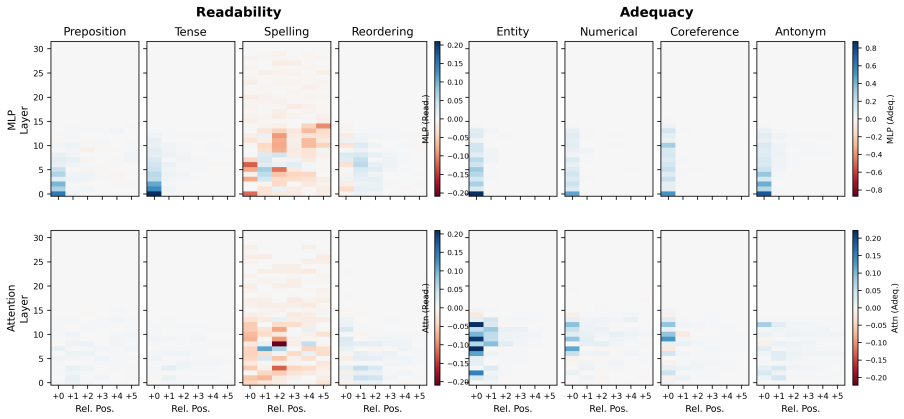
To address these limitations, we introduce two key controls in our perturbation framework. *Controlled intensity* specifies the number of altered tokens per sample via an intensity parameter k, which is explicitly recorded as metadata to enable sensitivity analysis with respect to error density. *Spatial localization* tracks the exact positional indices of all modified tokens, enabling causal tracing to anchor analysis on known perturbed regions rather than ambiguous spans.

Overall perturbation dimensions. Based on a survey of existing studies on NLG quality assessment (Fabbri et al., 2021; Honovich et al., 2022; Liu et al., 2023), we focus in this work on data perturbations along two dimensions: (1) *Readability*, which captures text fluency, grammatical correctness, and structural coherence, and (2) *Adequacy*, which concerns factual grounding and logical entailment of the generated text.

Perturbation Taxonomy. To construct a perturbation taxonomy suitable for mechanistic interpretability, where each node isolates a single failure mode with minimal confounding, we begin by reviewing existing error frameworks for generated text quality (Kryściński et al., 2020; Pagnoni et al., 2021; Ribeiro et al., 2020), yielding an initial set of over 35 error types. We then refine this set through a two-stage reduction process. First, we merge perturbations targeting the same underlying semantic or syntactic failure mode into unified categories (Table 1). For example, *entity swap*, *entity error*, and *perturb names/nouns* are consolidated into *Entity Swap*. Second, we exclude perturbations that



(a) Themis (Llama-3-8B)



(b) Prometheus (Mistral-7B)

Figure 2: Window-mode causal tracing across all eight attacks on both evaluators: (a) Themis, (b) Prometheus. Top row of each panel: MLP causal effects. Bottom row: attention o_{proj} causal effects. Effects are normalized so that 1.0 recovers clean expected-rating behavior and 0.0 matches the unmodified corrupt baseline; per-cell aggregates use a 20% trimmed mean over 3 seeds $\times$ 200 samples per attack. Four independent color scales per panel preserve within-criterion visibility because effect magnitudes differ by roughly an order of magnitude between Readability and Adequacy.

are incompatible with causal interpretability analysis, such as *stylistic alterations*, which entangle style with semantics and modify multiple non-local tokens; and *coarse perturbations* (e.g., noise injection or nonsensical text), which excessively disrupt global structure and obscure intermediate representations, making them unsuitable for causal tracing.

After refinement, the resulting taxonomy consists of eight perturbation types spanning two evaluation dimensions. **Readability Attacks** introduce structural perturbations that degrade fluency while preserving factual content. These include *Preposition Mismatch*, *Tense Mismatch*, and *Spelling Mistakes*, which induce localized syntactic or orthographic disruptions known to affect model sensitivity (Ribeiro et al., 2020; Belinkov and Bisk, 2017; Hu et al., 2024a), as well as clause- and sentence-level shuffling to disrupt sequential coherence in longer contexts (Zheng et al., 2023). **Adequacy Attacks** target factual grounding with respect to the source context through three mechanisms. First, we corrupt atomic factual units by replacing named entities, dates, and numerical quantities with plausible but incorrect alternatives (Kryściński et al., 2020; Li et al., 2023; Min et al., 2023). Second, we introduce referential errors by altering pronouns or noun phrases to refer to incorrect antecedents (Dziri et al., 2022). Third, we inject logical contradictions by substituting adjectives or verbs with their antonyms, following standard natural language inference and factual consistency paradigms (Bowman et al., 2015; Williams et al., 2018; Kryściński et al., 2020).

Automated Perturbation Generation. We operationalize the taxonomy at scale via an automated

generation pipeline. Clean summaries are drawn from CNN/DailyMail (Hermann et al., 2015). For each perturbation category and intensity k, GPT-4o is prompted with category-specific few-shot exemplars and instructed to inject exactly k instances of the targeted error. To eliminate the need for post-hoc text alignment, the generator returns not only the perturbed text but also explicit `original_tokens` and `new_tokens` arrays, providing a token-level map of the attack. Outputs are constrained to a predefined JSON schema via API-level response formatting.

A two-stage filter retains only samples suitable for mechanistic analysis. Structurally, samples that fail JSON decoding or violate the requested intensity bounds are dropped. Behaviorally, both clean and perturbed texts are scored by the target evaluator; samples where the rating is unchanged are discarded, as the perturbation was either invisible or below the evaluator's sensitivity threshold. This yields a behaviorally verified corpus in which every clean/corrupt pair demonstrably induces a rating change.

4 LLM-judge Interpretation

Activation Patching. We localize where in the network a perturbation's effect is processed by substituting clean activations into a corrupt forward pass at specific (layer, position) sites and measuring how much of the clean rating behavior is recovered (Vig et al., 2020; Meng et al., 2022). We trace both MLP sublayers and attention output projections (o_{proj}), and quantify each intervention with a normalized causal effect

$$\text{effect} = \frac{X_{\text{patched}} - X_{\text{corrupt}}}{X_{\text{clean}} - X_{\text{corrupt}}}, \qquad (1)$$

where X is an output statistic at the final position, scaled so that 1.0 recovers clean rating behavior and 0.0 matches the unmodified corrupt baseline. We use the standard logit difference $z[t_{\text{clean}}] - z[t_{\text{corrupt}}]$ for last-token-mode tracing and head knockout; for window-mode tracing, where per-token BPE alignment reduces per-sample yield, we use the expected rating $X = \sum_{r=1}^{5} r \cdot P(r)$ over the renormalized rating-token distribution (Liu et al., 2023). *Window-mode* tracing restores activations in a six-token window starting at the perturbed token (relative positions $+0$ through $+5$), localizing where in the input the perturbation is processed; filtering, alignment, and aggregation details are in Appendix B. *Last-token-mode* tracing restores activations only at the final input position at each layer, localizing the depth at which the rating decision is assembled.

Logit Lens. To trace at what depth the model's decision forms, we apply the logit lens (nostalgebraist, 2020) to the residual stream at the final input position: at each layer ℓ, we project the residual through the final layer norm and unembedding to obtain a layer-ℓ distribution over the five rating tokens $\{t_1, \ldots, t_5\}$. Sweeping ℓ across all layers produces a depth-resolved trajectory for each rating, from which we read both the crystallization depth and the rating ultimately committed to.

Attention Head Knockout. Attention head knockout identifies which individual heads implement the rating decision (Wang et al., 2023). For each of the $32 \times 32 = 1024$ heads in each model, we zero out that head's output during the forward pass on corrupt inputs and compute the resulting normalized causal effect (Equation 1) at the final position. Positive (red) effects identify *disruptive* heads that drive the corrupt rating - ablation releases the clean signal, while negative (blue) effects identify *suppressive* heads that defend the clean rating. We report per-attack heatmaps and inspect criterion-level patterns by comparing the four panels within each criterion group.

5 Experiments

5.1 Experimental Setup

We study two open-source LLM-judges from different base-model families: Themis (Hu et al., 2024b), built on Llama-3-8B, and Prometheus-7B-v2.0 (Kim et al., 2024), built on Mistral-7B. Both have 32 transformer layers and 32 attention heads per layer. For each of the eight perturbations from §3, we draw 200 clean/corrupt summary pairs from our CNN/DailyMail-derived corpus and score both texts with the target evaluator using a 1-5 NLG-evaluation prompt (Appendix A); Prometheus additionally uses a few-shot variant to place the rating token at the first generation position. Following the behavioral filter (§3), we retain only pairs whose rating changes after perturbation. All results are averaged over three seeds and 200 samples per attack.

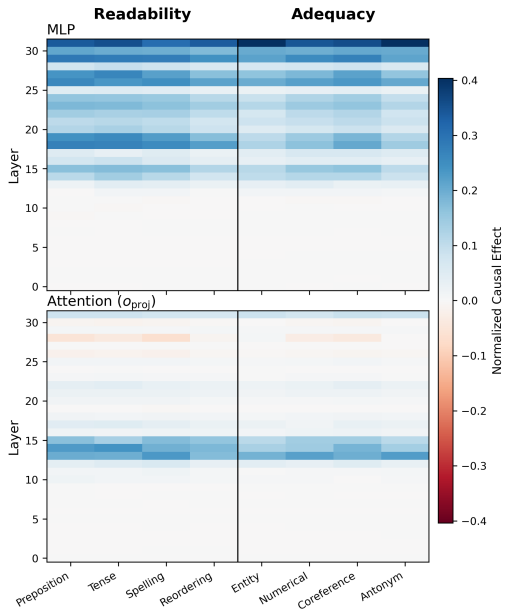


Figure 3: Last Token mode causal tracing on Themis across all eight attacks. Top row: MLP causal effects; bottom row: Attention o_{proj} causal effects.

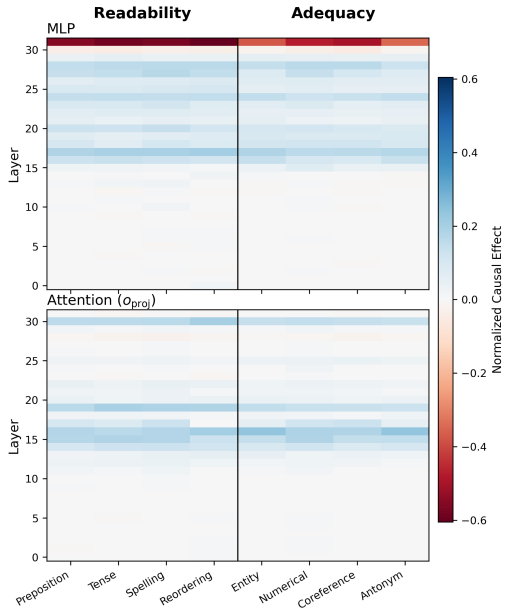


Figure 4: Last Token mode causal tracing on Prometheus across all eight attacks. Top row: MLP causal effects; bottom row: Attention o_{proj} causal effects.

5.2 Results and Analysis

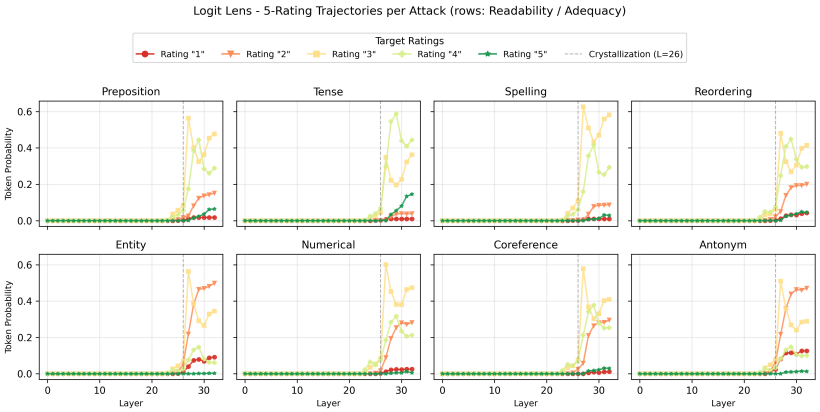
Errors register at early MLPs and route through mid-layer attention to the final position Early MLP layers locally register the perturbation at the perturbed-token position uniformly across attacks and across both evaluators. On Themis (Figure 2a, top) and Prometheus (Figure 2b, top), window-mode MLP causal effects concentrate at relative position $+0$ in layers 0-5 and are near zero elsewhere, with magnitudes strongest for the most token-localized attacks (Entity, Numerical, Antonym). The perturbed token's MLP output thus locally encodes the corrupted content irrespective of evaluation criterion or base-model family.

Window-mode attention diverges in attack-specific ways that mirror what each comparison requires, with three patterns visible in both models (Figure 2, bottom rows). Preposition spreads from $+0$ through $+2$-3 across L0-L10 (checking adjacent syntactic context); Entity and Numerical concentrate sharply at $+0$ in mid layers with no horizontal spread (retrieving from elsewhere in the article); and Coreference, Antonym, Tense, and Reordering produce weak window-mode attention overall, with their comparison happening through last-position attention to distant tokens outside our six-token window.

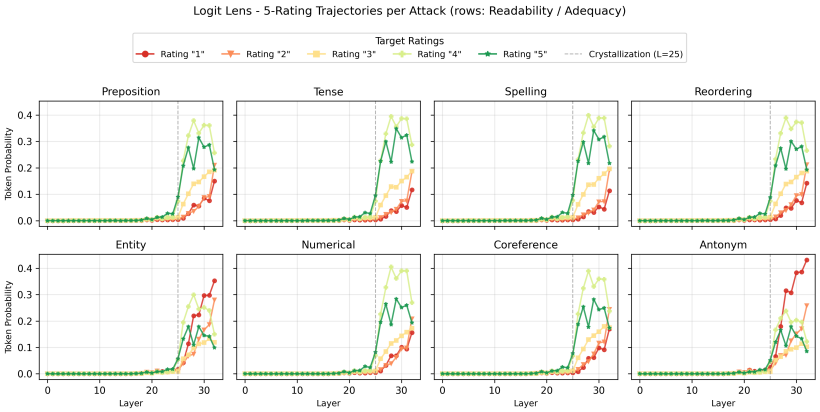
At the final position, all attacks share the same routing-and-integration architecture on Themis (Figure 3): last-position attention o_{proj} effects concentrate in a single band at L13-L15 where perturbation information is routed into the last token, after which the MLP effect rises in a stepped cascade with bands at L17-L18, L25-L27, and a peak at L31 that acts as the final write into the rating-token direction. Prometheus shares the same routing band but diverges at the top of the cascade (Figure 4): its last-position attention spans more depths (L14-L17, L19-L21, L29-L30), and its L31 MLP effect is strongly negative where Themis's is strongly positive. Themis thus commits the rating in one dominant top-layer step; Prometheus distributes the commit across mid-to-late layers and uses L31 for an opposite-direction correction.

The rating decision crystallizes at a sharp late layer (L25–L26) The rating decision crystallizes at a sharp, late layer that is shared across all attacks. On Themis (Figure 5a), the logit-lens probability mass on all five rating tokens is essentially zero through $L \sim 22$ and then rises steeply over a narrow band, with every attack transitioning at the same depth; we refer to $L = 26$ as the crystallization layer. Prometheus shows the same qualitative shape one layer earlier at $L = 25$ (Figure 5b), indicating that depth-aligned late commitment is a property of both base architectures.

Post-crystallization, the model commits to different ratings for different attacks, revealing an



(a) Themis (Llama-3-8B), crystallization at $L = 26$



(b) Prometheus (Mistral-7B), crystallization at $L = 25$

Figure 5: Logit-lens trajectories for each of the five rating tokens, per attack, on (a) Themis and (b) Prometheus. Top row of each panel: Readability attacks; bottom row: Adequacy attacks. Crystallization (dashed line) occurs at $L = 26$ on Themis and $L = 25$ on Prometheus.

implicit severity hierarchy. On Themis, Entity and Antonym crystallize to Rating 2 (harshest), Tense to Rating 4 (mildest), and the remaining attacks cluster at Rating 3. Prometheus reproduces this ordering more punitively (Entity and Antonym at Rating 1, most others at Rating 4) but with markedly less confident trajectories: where Themis's corrupt-rating probability climbs monotonically toward ~ 1.0 by the final layer, Prometheus's peaks at ~ 0.5 around $L = 30$ and then declines, with a sustained Rating-5 prior visible pre-crystallization that is never fully suppressed. The same crystallization depth therefore underlies very different commitment trajectories, consistent with the divergent top-of-cascade MLP dynamics shown in Figures 3 and 4.

A narrow band of attention heads at L10–L11 implements the rating verdict On Themis (Figure 6), every individual attack panel shows a horizontal band at L10-L11 containing both disruptive (red) and suppressive (blue) heads in an alternating pattern, with negligible individual-head effects elsewhere. The alternating sign indicates a small set of cooperating-and-competing heads that together implement the rating decision irrespective of attack.

Within this shared band, the two criteria differ in upstream-head recruitment: the Readability panels (Figure 6, top row) show additional head activity at L7-L9 contributing nearly as strongly as the L10-L11 band itself, while the Adequacy panels (bottom row) remain markedly sparser between L5 and L9. This matches the window-mode finding of Figure 2: Readability's broader early-layer recruitment aligns with its local-attention spreading, while Adequacy's tighter L10-L11 profile aligns with the mid-layer concentration of Entity- and Numerical-type attention.

Prometheus exhibits the same evaluation-band concept but distributed more broadly across L10-L15, with the per-criterion sparsity pattern inverted

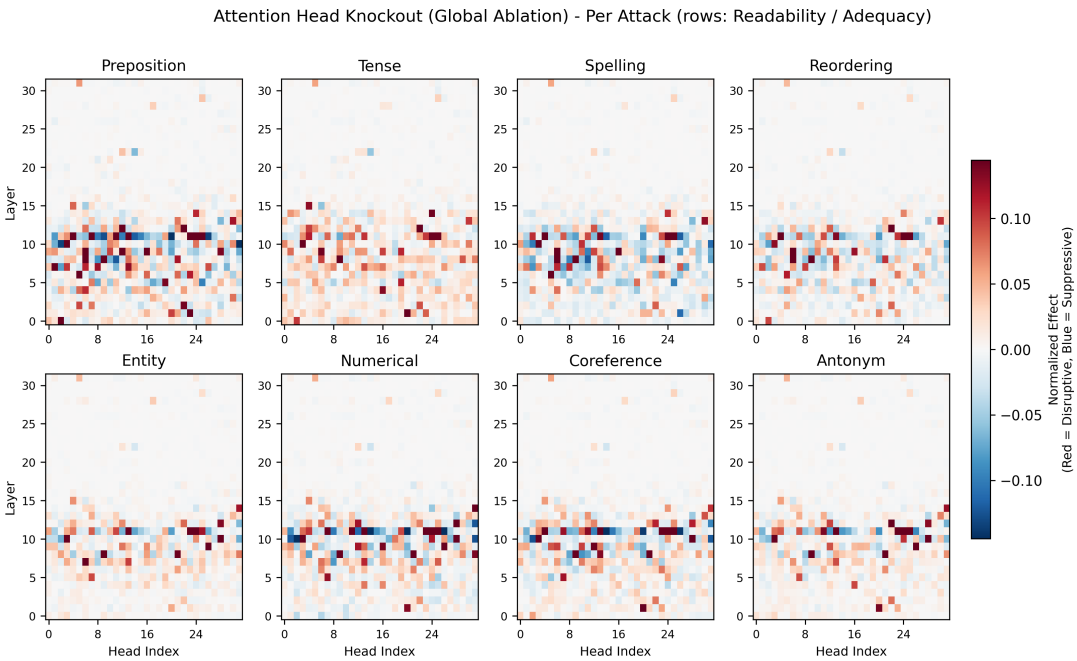


Figure 6: Per-attack attention-head knockout heatmaps on Themis. Top row: Readability attacks (Preposition, Tense, Spelling, Reordering). Bottom row: Adequacy attacks (Entity, Numerical, Coreference, Antonym). The L10-L11 evaluation band appears in every individual attack panel, with both disruptive (red) and suppressive (blue) heads in an alternating pattern; Readability attacks additionally recruit heads at L7-L9, while Adequacy attacks remain sparse between L5 and L9.

relative to Themis (Readability denser, Adequacy sparser; Appendix Figure 7). Both models show essentially no individual-head effects above L15, consistent with the division of labor identified across our experiments: attention does the comparison and routing work below L15, after which the upper-layer MLP cascade integrates the routed signal and writes the rating, with no individual head playing a critical role.

An End-to-End Two-Stage Evaluation Pipeline. These findings synthesize into a two-stage pipeline shared across both evaluators: below layer 15, attention performs comparison and routing (early MLPs encode the corrupted content at the perturbed token, an L10–L11 band of heads implements the rating verdict, and L13–L15 attention routes the verdict to the final input position); above it, the MLP cascade integrates the routed signal and writes the rating, with the decision crystallizing at a sharp late layer ($L = 26$ on Themis, $L = 25$ on Prometheus). Prometheus shares this scaffold but diverges at the top of the cascade, using $L31$ for an opposite-direction correction where Themis writes in one dominant top-layer step—indicating that LLM-judges across different base-model families share a common evaluation pipeline but implement family-specific output-assembly strategies.

6 Conclusion

Across four mechanistic experiments on two LLM-based evaluators from different base-model families - Themis (Llama-3-8B) and Prometheus (Mistral-7B), we find that NLG judges implement a structured, coherent evaluation pipeline rather than an ad-hoc rating-generation shortcut: attention performs local error comparison and routes the result to the final position below L15, after which the upper-layer MLP cascade integrates the routed signal and writes the rating, with the decision crystallizing in the residual stream at a sharp late layer ($L = 26$ on Themis, $L = 25$ on Prometheus). This scaffold is preserved across both base architectures, while local-attention recruitment varies systematically by criterion and commitment dynamics at the top of the cascade diverge by base-model family. We release the eight-attack taxonomy, generation pipeline, and behaviorally verified clean/corrupt corpus as public resources to support both further interpretability research on NLG evaluators and downstream applications such as contrastive evaluator training, distillation of localized evaluators, and polarity-reversed fine-tuning for self-correction.

7 Limitations

Model and task scope. Our circuit-level findings are based on two open-source evaluators (Themis built on Llama-3-8B; Prometheus built on Mistral-7B) applied to English-language summaries drawn from a single source distribution (CNN/DailyMail), using a fixed evaluation prompt per evaluator (Appendix A). While our taxonomy is designed to be task-agnostic and the two base architectures are drawn from distinct model families, we have not directly verified the identified circuit on dedicated evaluators of dialogue response generation, story generation, or factuality assessment outside summarization, nor on larger or multilingual evaluator models. Generalization of the two-stage circuit structure to these settings, and its stability under prompt variation, remains an empirical question.

Behavioral-filter selection. As described in §3, our analysis is restricted to samples whose evaluator rating demonstrably changes under perturbation. This focuses our circuit characterization on cases where the evaluation procedure successfully detects the perturbation; samples on which the evaluator misses the perturbation entirely fall outside our analysis. Such failure-mode samples carry interpretable signal about evaluator robustness that the present work does not address.

Single intensity and single-attack samples. All experiments use single-attack samples at intensity $k=1$ (one perturbed token or token-pair per sample). Circuit behavior under higher intensities, or under mixed-attack samples in which two or more perturbation categories co-occur, is not characterized here; we flag this as a follow-up direction in §6.

Cross-family claims rest on two evaluators. Patterns we describe as "shared across base-model families" or "family-specific" are supported by $n = 2$ evaluators, one drawn from each of two architectures. These observations should be read as evidence from two independent data points rather than as generalizations across the full Llama or Mistral families, or across the broader space of LLM-based NLG evaluators.

Logit-lens approximation. Our depth-resolved decoding of rating commitments uses the logit lens, which approximates intermediate beliefs by projecting the residual stream through the final unembedding. Where the residual stream is rotated or scaled relative to its final-layer geometry, the logit lens can under-report probability mass on the eventual rating; tuned-lens variants address this but introduce per-layer training that is not directly comparable across base-model families. We chose the logit lens for cross-architecture consistency and treat absolute pre-crystallization probabilities as qualitative rather than calibrated.

Generator-model dependence. Perturbations are produced by GPT-4o under category-specific prompts (§3). Any systematic stylistic bias in the generator could in principle correlate with evaluator response patterns in ways the present work does not fully isolate. We mitigate this by retaining only behaviorally-verified clean/corrupt pairs and by tracking perturbed positions explicitly, but a fully generator-independent characterization is outside the present scope.

8 Ethics Statement

All experiments were conducted on publicly available datasets (CNN/DailyMail summarization corpus, Hermann et al., 2015) and publicly released evaluator models (Themis, Hu et al., 2024b; Prometheus, Kim et al., 2024). No personally identifiable information was collected or processed, and no human subjects were involved beyond the dataset's original construction. Perturbation generation uses GPT-4o via standard API access; generated perturbations are limited to controlled syntactic and semantic alterations of public news summaries and do not produce content beyond the source-text scope.

We release the perturbation taxonomy, generation pipeline, and behaviorally-verified clean/corrupt corpus as public resources to support further mechanistic study of NLG evaluators. The same artifacts could in principle be repurposed to construct adversarial inputs targeting evaluator failure modes. We view this risk as limited: the perturbation categories we study (entity swaps, antonym substitutions, grammatical and typographical errors, and clause reordering) are well-established in the prior NLG-evaluation literature, and our circuit-level findings do not by themselves provide novel attack vectors against deployed LLM-judges. We assess the net contribution enabling more careful evaluator development through mechanistic understanding of how current evaluators operate as positive.

Use of AI Assistants. We used AI assistants

(Claude and ChatGPT) during the preparation of this work for two purposes: (i) supporting code implementation and debugging of our causal tracing and attention-head-knockout infrastructure, and (ii) language polishing, proofreading, and improving the stylistic clarity of the manuscript. AI assistants were not used to generate research questions, design experiments, or interpret results. All research design, experimental decisions, scientific interpretations, and final text were developed and verified by the human authors, who hold full responsibility for the integrity and accuracy of this work.

References

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 632–642.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar R Zaïane, Mo Yu, Edoardo M Ponti, and Siva Reddy. 2022. Faithdial: A faithful benchmark for information-seeking dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920.

Xinyu Hu, Mingqi Gao, Sen Hu, Yang Zhang, Yicheng Chen, Teng Xu, and Xiaojun Wan. 2024a. Are llm-based evaluators confusing nlg quality criteria? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9530–9570.

Xinyu Hu, Li Lin, Mingqi Gao, Xunjian Yin, and Xiaojun Wan. 2024b. Themis: A reference-free nlg evaluation language model with flexibility and interpretability. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15924–15951.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 9332–9346.

Junyi Li, Xiaoxue Cheng, Wayne 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.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 2511–2522.

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.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.

nostalgebraist. 2020. interpreting GPT: the logit lens. LessWrong.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4902–4912.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33.

Kevin 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)*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Evaluation Prompts

This appendix lists the full prompt templates used to elicit ratings from each evaluator. Braced fields (`{article}`, `{summary}`, `{prompt_criteria}`, `{examples_block}`) are Python format-string placeholders substituted at runtime. Line wrapping within the templates below is a display artifact of the two-column layout; the rendered prompts use single logical lines per paragraph.

A.1 Themis Prompt

```
###Instruction###
Please act as an impartial and helpful evaluator for natural
language generation (NLG).
Your task is to evaluate the quality of the Summarization
strictly based on the given evaluation criterion.
You MUST keep to the strict boundaries of the evaluation
criterion.
**Scoring Rules:**
1. Use a Likert scale from 1 to 5.
2. 5 is the highest quality, 1 is the lowest.
3. Output ONLY the number.
###Evaluation Criterion###
{prompt_criteria}
###Task###
Article:
{article}
Summary:
{summary}
###Rating###
The rating is
```

A.2 Prometheus Prompt

Prometheus requires a few-shot variant of the prompt to reliably emit the rating token at the first generation position. The prompt body is:

```
###Task Description:
You are an impartial evaluator. Read the article, the response,
and the score rubric. Output ONLY a single integer between 1
and 5 representing your rating. Do not write any feedback,
explanation, or any other text -- just the digit. The examples
below illustrate the expected output format.
{examples_block}
###Now evaluate this:
###Article:
{article}
###Response to evaluate:
{summary}
###Score Rubric:
{prompt_criteria}
```

The `{examples_block}` placeholder is filled with the five article/summary/rubric/rating exemplars below, which span the full 1–5 rating range. All five share the same scoring rubric: "Evaluate the overall quality of the summary based on how accurately and completely it captures the main facts of the article. Rate from 1 (highly inaccurate or off-topic) to 5 (fully accurate and comprehensive)."

Example 1 (Rating 4). *Article:* The Federal Reserve announced a 0.25 percentage point interest rate cut today, the third such reduction this year, citing concerns about slowing economic growth and persistent global trade tensions affecting U.S. manufacturing output. *Summary:* The Fed cut interest rates by 0.25 points today due to economic growth concerns.

Example 2 (Rating 2). *Article:* The European Union announced a new trade agreement with Vietnam yesterday, eliminating tariffs on 99 percent of goods traded between the two regions over the next seven years and establishing new standards for labor and environmental protections. *Summary:* Vietnam and the EU had a meeting about trade and economic cooperation last week.

Example 3 (Rating 5). *Article:* Scientists at Stanford University have developed a new lithium-metal battery technology that can charge a smartphone in 30 seconds and last three times longer than current lithium-ion batteries, with potential applications in electric vehicles and grid storage. *Summary:* Stanford researchers developed a lithium-metal battery that charges phones in 30 seconds and lasts three times longer than lithium-ion, with potential uses in electric vehicles and grid storage.

Example 4 (Rating 1). *Article:* NASA's Perseverance rover has identified organic compounds in rock samples from Mars's Jezero crater, providing the strongest evidence yet of conditions that could have supported ancient microbial life on the planet, according to a study published in *Nature*. *Summary:* Astronauts from China landed on the moon last week to collect lunar samples for future research.

Example 5 (Rating 3). *Article:* A magnitude 6.2 earthquake struck off the coast of northeastern Japan early Tuesday morning, causing minor structural damage to buildings in three coastal cities and triggering brief tsunami advisories that were lifted within two hours. No casualties have been reported. *Summary:* An earthquake hit Japan and caused some damage. There were no injuries.

B Window-Mode Causal Tracing: Methodology Details

This appendix documents the filtering, alignment, and aggregation procedures used for window-mode causal tracing (§5). Last-token-mode tracing and head knockout follow the standard logit-difference formulation and do not require these additional controls, as their per-sample yield is not constrained by per-token alignment.

Sample-level filters. Three filters are applied before per-cell aggregation. (i) *Non-rating output filter:* samples whose clean forward pass does not assign substantial probability mass to the rating-token set are discarded; this is rare but catches degenerate prompts. (ii) *Score-difference filter:* we require $|E_{\text{clean}} - E_{\text{corrupt}}| \geq 0.05$, corresponding to approximately 5% probability mass shifting between adjacent integer ratings. Samples below this threshold yield uninterpretable patching effects because the denominator of Equation 1 approaches zero. Pass rates under this filter are roughly 25–40% for Readability attacks and 50–70% for Adequacy attacks, themselves an indicator of evaluator sensitivity per attack. (iii) *Anchor-validity filter:* per-token edit operations between clean and corrupt sequences are identified using `difflib` sequence-matching opcodes (`replace`, `insert`, `delete`, `equal`), and cross-referenced against the perturbation pipeline's recorded modification positions to confirm the anchor is at the intended site. For Reordering, where the pipeline's recorded swap can be inaccurate relative to the emitted text, we use difflib directly.

Token-length mismatch handling. Several perturbation categories (notably Spelling, Numerical, and Reordering) produce corrupt sequences of different length than clean: BPE tokenizers split misspellings and numerical substrings inconsistently. Naïvely indexing the clean cache at the corrupt-side position is therefore incorrect. We compute a per-token clean-source-position mapping from the difflib opcodes: `equal` opcodes yield a one-to-one mapping (`clean_pos = corrupt_pos + offset`); `replace` opcodes are clamped to the last valid clean position within the replaced span; and `insert` opcodes have no clean correspondence and are recorded as `NaN`. When patching at (layer, target position) in the corrupt run, we read the source activation from (layer, mapped clean-source position) in the clean run; cells with no valid alignment remain `NaN` and are excluded from the per-cell aggregate.

Per-cell aggregation. For each (layer, relative-position) cell, we accumulate effect values from all samples retained after filtering, pooled across three random seeds (23, 42, 17) each sampling a different 200-sample subset of the source corpus. The number of contributing samples varies per cell and per attack: well-behaved attacks have $N \approx 400$–500 per cell after seed pooling, while attacks with BPE-unstable perturbations retain as few as ~ 20 per cell (notably Themis on Spelling, where the Llama-3 tokenizer splits misspellings inconsistently). We aggregate per cell with a 20% trimmed mean: dropping the smallest and largest 10% of contributions and averaging the rest. This is the standard robust-statistic choice; plain mean is outlier-dominated at small N, while the median over-collapses the heavy-tailed distributions we observe (the median sits in a near-zero bulk and discards genuine signal). Cells with $N < 10$ are additionally masked as `NaN` and render in light gray.

Color scaling. Effect magnitudes differ by roughly an order of magnitude between Readability and Adequacy attacks (and between MLP and attention rows). To preserve within-criterion visibility while keeping the cross-criterion magnitude difference itself legible, each figure uses four independent color scales (one per row $\times$ criterion combination), displayed via four colorbars: two between the criterion groups and two on the right edge. The colormap is symmetric (`RdBu`) around zero.

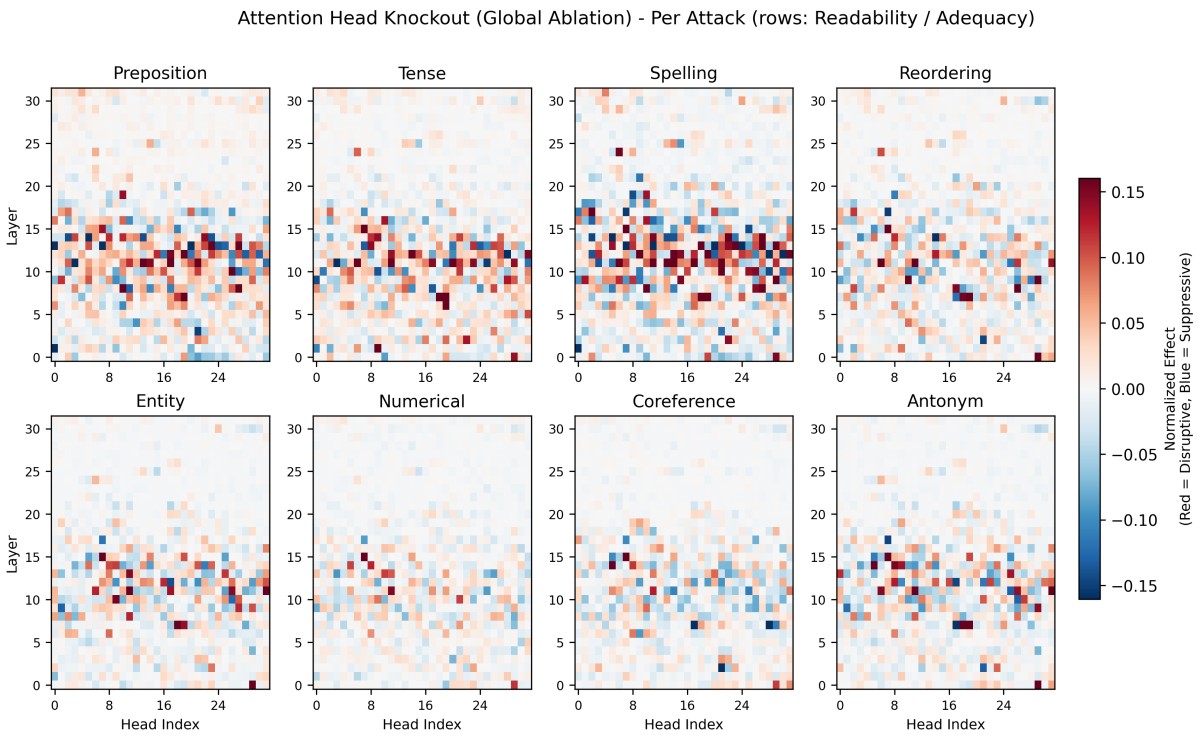


Figure 7: Per-attack attention-head knockout heatmaps on Prometheus. Top row: Readability attacks; bottom row: Adequacy attacks. The broader L10-L15 evaluation band is present in individual attack panels, with the Readability panels showing denser head activity than the Adequacy panels, the inverse of the per-criterion sparsity pattern observed on Themis (Figure 6).

C Model Size, Computational Budget, and Infrastructure

Model parameters. Our analysis targets two open-source NLG evaluator models: **Themis** (Hu et al., 2024b), built on Llama-3-8B with approximately 8 billion parameters, and **Prometheus-7B-v2.0** (Kim et al., 2024), built on Mistral-7B with approximately 7 billion parameters. Both models share a comparable transformer architecture: 32 layers, 32 attention heads per layer, and a hidden dimension of 4096. Perturbation generation uses **GPT-4o** via the OpenAI API; its parameter count is not publicly disclosed by the provider.

Computing infrastructure. The majority of mechanistic-analysis experiments (approximately 80% of GPU-hours) were conducted on a cluster of 16× **NVIDIA GeForce RTX 2080 Ti** GPUs (11 GB VRAM each), with the remaining 20% run on a single **NVIDIA H100** (80 GB) GPU. Models were loaded in **bfloat16** for inference and activation patching, with model-parallel sharding across the eight 2080 Ti devices to accommodate the 7–8 B parameter evaluators. Implementation used PyTorch and the HuggingFace `transformers` library.

Computational budget. The full experimental battery used approximately **800** GPU-hours, distributed as follows:

- **Perturbation generation** (GPT-4o API): approximately **5000** API calls across 8 attacks $\times$ 200 samples $\times$ 3 seeds, at an approximate cost of **$100**.
- **Window-mode and last-token-mode causal tracing**: **250** GPU-hours, dominated by per-layer/per-position activation patching across 32 layers $\times$ both MLP and attention sublayers.
- **Logit-lens analysis**: **150** GPU-hours.
- **Attention-head knockout** ($32 \times 32 = 1024$ heads per model, both evaluators): **400** GPU-hours.

We estimate the full experimental battery is reproducible with approximately 800 GPU-hours on equivalent consumer-grade hardware, or substantially fewer on high-memory data-center GPUs (e.g., A100/H100) that avoid the model-parallel sharding overhead.