Judge Circuits

Anonymous ACL submission

Abstract

LLM-as-a-judge has become the dominant paradigm for grading model outputs at scale, yet the same model assigns systematically different scores when its output format changes (e.g., a 1–5 rating vs. a True/False label). Existing diagnoses of these format-induced inconsistencies stop at the input-output level. Using Position-aware Edge Attribution Patching (PEAP), we causally investigate the internal mechanism in Gemma-3, Qwen2.5, and Llama-3. We find that judgments across structured understanding and open-ended preference tasks share a sparse, generalized *Latent Evaluator* sub-graph in the mid-to-late multi-layer perceptrons (MLPs); zero-ablating it collapses judgment while preserving world knowledge in architecturally modular models. By structurally decoupling abstract judging from output formatting, we provide a mechanistic account of format-induced inconsistency on the open-weight models we study: a continuous judgment signal computed in the shared trunk is mapped through fragile, format-specific terminal branches, enabling format-independent preference to be isolated downstream of the requested output format. Our findings imply that benchmark-level reliability comparisons across formats are partially measuring formatter geometry rather than evaluation quality.

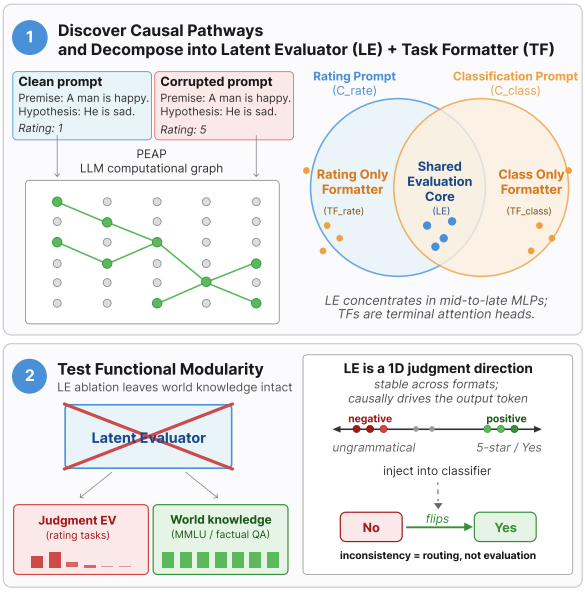


Figure 1: Overview of our pipeline on an MNLI minimal pair: (**1**) PEAP (Haklay et al., 2025) traces cross-token causal edges from the differential input tokens into a shared *Latent Evaluator* sub-circuit ($\mathcal{C}_{\text{LE}} := \mathcal{C}_{\text{rate}} \cap \mathcal{C}_{\text{class}}$). (**2**) We validate this circuit three ways: zero-ablation (red $\times$) isolates evaluation from world knowledge; BDAS (Wu et al., 2023) identifies a 1D judgment direction in the LE's activation space; Task Formatters ($\mathcal{C}_{\text{TF,rate}}, \mathcal{C}_{\text{TF,class}}$) in terminal layers map that judgment scalar to the concrete target token.

1 Introduction

The LLM-as-a-Judge (LaaJ) paradigm is now widespread across NLP for evaluation tasks such as benchmark scoring, reward modeling, and content moderation – automating quality assessment without a human in the loop (Calderon et al., 2025; Gao et al., 2025; Li et al., 2024). However, the reliability of LLMs as automated judges is heavily contested. Lee et al. (2025) document a contradictory dissociation – relative preferences are often consistent, but absolute ratings are not – and isolate two specific failure modes: self-consistency across repeated evaluations, and inter-scale consistency across different rating formats. Even large proprietary models fail on both dimensions, undermining the reproducibility of any LaaJ-driven leaderboard, reward, or safety judgment. Eshuijs et al. (2025) corroborate this from a different angle, showing that models frequently exploit shallow classification shortcuts – e.g., relying on lexical cues such as response length or sentiment polarity – rather than integrating the multiple aspects of input and target that holistic evaluation requires. Comparable inconsistency and calibration failures hold for judges of $<$ 70B parameters (Girrbach et al., 2025).

No prior work has investigated the internal com-

putational mechanisms underlying LLM judgment, a necessary step toward understanding and improving LaaJ reliability. We recast the diagnostic question from *"does the model judge consistently?"* to *"where in the computational pathway does format-induced inconsistency originate?"*, and hypothesize that LaaJ implements judgment via two architecturally separable sub-systems – a shared evaluation core and a format-specific output router – and that the consistency failures in Lee et al. (2025) localize to the latter.

To test this, we use Position-aware Edge Attribution Patching (PEAP) (Haklay et al., 2025), which handles the cross-token edges needed for judge circuits whose inputs span separated linguistic spans (e.g., premise vs. hypothesis) while remaining linear-in-edges to compute. Drawing on the intermediate-variable (Lepori et al., 2024) and formal/functional dissociation (Hanna et al., 2026) literature, we cross-validate every circuit with three independent causal probes – cumulative edge patching, subspace steering, and cross-format activation transfer – which converge on the same Latent Evaluator components and guard against non-identifiability (Miller et al., 2024; Méloux et al., 2025) (Figure 1).

Contributions:

(1) **We show that LLM judgment is computed by highly sparse, cross-task circuits** sharing a generalized *Latent Evaluator* in mid-to-late MLPs, recoverable at top-$k \leq 200$ edges.

(2) **We show that judgment modularity is architecture-dependent**: Qwen modular at 7B, Gemma only at 27B. On modular models, zero-ablating the Latent Evaluator preserves world knowledge while collapsing judgment; on Gemma-3-12B it degrades both, indicating tight entanglement with world-knowledge pathways.

(3) **We provide a mechanistic explanation of inter-format LLM evaluator inconsistency**, localizing it to format-specific output routing rather than to the underlying evaluation.

Together, these results suggest that LaaJ format inconsistency is a routing problem rather than an evaluation problem – and therefore that fixes can target the formatter without disturbing the model's judgment competence. [1]

[1] Code and data available at: https://anonymous.4open.science/r/JuCi/

2 Experimental Setup

Central Finding. *An LLM-as-a-judge implements judgment via two architecturally separable sub-systems – a shared evaluation core and a format-specific output router.*

We test this in three steps: §3 discovers the candidate sub-circuits; §4 probes whether the shared core is functionally isolated; §5 causally validates the split via cross-format activation transfer.

A *judgment task* in our setting asks the model to assign a quality, preference, or correctness score to a candidate text given the input it conditions on, producing a scalar rating or categorical verdict over the candidate rather than a free-form generation. Our pipeline operates on contrastive minimal-pair prompts (Figure 1); the rating-vs-classification decomposition into a *Latent Evaluator* and format-specific *Task Formatters* is introduced in §4.1.

Data We select five datasets that together span the three dimensions of evaluation that LaaJ is deployed for: (i) structured linguistic correctness (CoLA, MultiNLI, STS-B), (ii) preference / quality judgment (RewardBench), and (iii) subjective sentiment (Yelp).

- CoLA (linguistic acceptability) (Warstadt et al.): fluency and grammaticality as quality criteria.
- MultiNLI (natural language inference) (Williams et al., 2018): entailment / neutral / contradiction between a hypothesis and a premise.
- STS-B (sentence semantic similarity) (Cer et al., 2017): semantic equivalence between pairs.
- RewardBench (preference evaluation) (Lambert et al., 2025): the canonical testbed for open-ended LLM-as-a-judge capabilities.
- Yelp (sentiment, 1–5 star reviews) (Zhang et al., 2015): a subjective, user-written evaluation domain with a natural ordinal scale.

Models We evaluate five instruct-tuned models from three families: Gemma-3 (12B-it, 27B-it) (Team et al., 2025), Qwen2.5 (7B-Instruct, 14B-Instruct) (Qwen et al., 2025), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024), accessed via TransformerLens (Nanda and Bloom, 2022). We cap the minimal-pair subset at $|S| = 500$ for MNLI; CoLA, STS-B, RewardBench, and Yelp have 100 – 200 valid semantic pairs each. The split-half reliability check (App. M) confirms that within-task circuit IoU is comparable across these subset sizes. The computational geometry constraints behind the

cap and our backward-pass tracing budget are deferred to App. G.

Prompt design For each dataset we construct contrastive *minimal pairs*: a clean prompt (correct rating) and a corrupted prompt (incorrect rating) with matched token lengths for PEAP attribution[2]. Half the pairs assign the higher rating to the clean prompt and half to the corrupted prompt, so that per-edge attributions are symmetric by construction (§3.1). We format every input as a 1–5 rating prompt; to enable contrastive circuit analysis (§4.1), we additionally pair each dataset with a parallel classification-control prompt (categorical Yes/No, True/False, or Entailment/Contradiction labels) on the same instances. Exact templates and padding/alignment details are in Appendices F–G.

3 Discovering Judge Circuits in LLMs

We use *judge circuit* to refer to the sparse causal sub-circuit a model uses to compute a rating from a structured prompt; §4.1 decomposes it into a shared evaluation core ($\mathcal{C}_{\text{LE}}$) and a format-specific output branch ($\mathcal{C}_{\text{TF}}$). Our two-stage pipeline first applies PEAP to identify the causal pathways responsible for evaluation, then isolates task-specific formatting mechanisms from generic evaluation logic using contrastive control tasks.

3.1 Circuit Discovery via PEAP

Circuit discovery in decoder-only LLMs conceptualizes the forward pass as a computation graph $\mathcal{G}$ whose nodes are MLPs and attention heads and whose directed edges carry information flow, and seeks a sparse subgraph $\mathcal{C} \subset \mathcal{G}$ that causally accounts for a target behavior (Vig et al., 2020; Conmy et al., 2023; Wang et al., 2023). **Position-aware Edge Attribution Patching** (PEAP) (Haklay et al., 2025) extends Edge Attribution Patching (Hanna et al., 2024) to capture causal edges *across token positions* in addition to intra-token ones – a necessary property for judge circuits that must cross-reference separated linguistic spans (e.g., premise vs. hypothesis). Concretely, for each candidate edge from sender S to receiver R, PEAP estimates causal importance by the dot product of the receiver's gradient ∇R with the difference between the sender's activation on the clean and corrupted inputs ($S_{\text{clean}} - S_{\text{corr}}$). A single backward pass yields all receiver gradients simultaneously, so the entire ranked edge list over attention heads and MLPs is extracted in one forward–backward sweep per minimal pair. We extend PEAP with a symmetric polarity correction (full formulas in Appendix A) that handles our bidirectional minimal pairs (§2) without canceling genuine causal signal under naïve gradient summation. We separately verify that the extracted circuits are faithful to the full model (Appendix C) and stable under data resampling (Appendix M).

3.2 Structural Overlap: The Latent Evaluator

Cross-task structural overlap is established evidence of shared computation in transformer circuits (Tigges et al., 2024; Ferrando and Costa-jussà, 2024; Lan et al., 2024). Given two circuits $\mathcal{C}_A, \mathcal{C}_B$ traced on different tasks A and B and pruned to their top-k edges, we quantify similarity via Jaccard Intersection-over-Union on both the set of unique edges $\mathcal{E}$ and distinct components $\mathcal{N}$, abstracting away token positions:

$$\text{IoU}_{\text{edge}} = \frac{|\mathcal{E}_A \cap \mathcal{E}_B|}{|\mathcal{E}_A \cup \mathcal{E}_B|}, \quad \text{IoU}_{\text{node}} = \frac{|\mathcal{N}_A \cap \mathcal{N}_B|}{|\mathcal{N}_A \cup \mathcal{N}_B|}.$$

Edge IoU is the stricter metric; Node IoU measures architectural recruitment at a coarser grain.

Finding 1: Distinct judgment tasks share a dense computational trunk on every modular architecture.

On Gemma-3-12B at top-200 (Figure 4; Node IoU in Appendix B), we measure 61.0% Node IoU / 35.3% Edge IoU between CoLA and MNLI, 62.3% / 42.1% between MNLI and STS-B, and 48.8% Node IoU / 31.1% Edge IoU between RewardBench and CoLA. The same shared-trunk pattern holds across the modular models (Figure 5). Qwen2.5-7B in particular achieves a uniformly high Edge IoU (34.9–47.0%) on every task pair we test, including the open-ended RewardBench pairings. Qwen2.5-14B and Gemma-3-27B post lower raw Edge IoUs at the same k, but their Node IoUs remain substantial (28.1–55.9%), consistent with the scale-dependent redundancy effect documented in Appendix M: larger modular models route judgment through multiple computationally equivalent sub-pathways, so the same components are recruited but the specific top-200 edges differ across data splits. To rule out the possibility that this overlap reflects sample-size noise rather than

[2]For MNLI, minimal pairs are drawn from the entailment, contradiction subset; neutral instances are excluded so that clean and corrupted prompts have semantically opposed ground truth (App. G details the per-task selection rules).

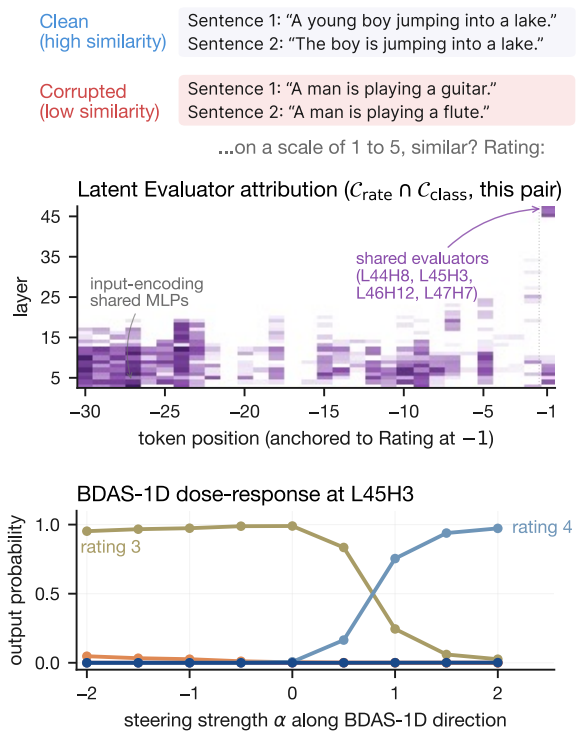


Figure 2: Anatomy of judgment on one STS-B pair (Gemma-3-12B, #149). **Top**: clean (high-sim.) and corrupted (low-sim.) prompts. **Middle**: PEAP attribution restricted to $\mathcal{C}_{\text{LE}} := \mathcal{C}_{\text{rate}} \cap \mathcal{C}_{\text{class}}$ (§4.1, top-1000 aggregate): early shared input-encoding MLPs plus late shared evaluators (App. J). **Bottom**: steering along BDAS-1D at L45H3 (App. D), α calibrated to one pair's worth of natural causal signal, flips the output from rating 3 to rating 4 with a smooth crossover at $\alpha \approx 0.5$–1.0.

genuine shared structure, we compute within-task split-half reliability on Gemma-3-12B at the same k: Node IoU is 76.3% on MNLI, 80.6% on STS-B, 61.6% on CoLA (Appendix M), meeting or exceeding the cross-task numbers. A layer-wise decomposition of Edge IoU (App. P) further localizes this cross-task overlap to the mid-to-late layers across the evaluated models, consistent with the LE/TF depth split developed in §4.1.

3.3 Sparse Circuit Faithfulness

To validate that the PEAP-discovered edges are causally *sufficient* for the model's judgment, we apply the per-instance MIB faithfulness metric (Mueller et al., 2025): starting from a fully corrupted forward pass, we progressively restore the top-k PEAP edges and measure the median fraction of the clean–corrupted EV gap (§3.1) that the patched sub-circuit recovers (full methodology and the magnitude-weighted sensitivity analysis are in Appendices C and N).

Finding 2: PEAP recovers highly sparse, faithful circuits across models and tasks.

Across the 25 (model, task) cells we trace, 21 reach median recovery ≥ 0.87 at some $k \leq 200$ (Figure 3); on Gemma-3-27B the open-ended RewardBench circuit saturates at median ≈ 1.0 with just $k = 5$ edges. The non-saturating cells are Gemma-3-12B on RewardBench and Yelp (median recovery ≈ 0 through $k = 200$, consistent with that model's functional entanglement of judgment with world-knowledge pathways; §4.2, Table 1), Llama-3.1-8B on MNLI (slower climb, ≈ 0.82 at $k = 200$), and Llama-3.1-8B on Yelp (peaks at ≈ 0.40 before drifting back down). A randomly-sampled-edge baseline (gray dashed line) hovers near 0% across every configuration, ruling out the possibility that any sparse subgraph would suffice.

Cross-method robustness. Faithfulness rules out the metric-fragility concern about sparse circuit extraction; the complementary non-identifiability concern flagged in §1 (Méloux et al., 2025), that different attribution algorithms may select different sparse subgraphs on the same model and task, we address by re-tracing every Qwen2.5-7B and Gemma-3-12B circuit with *LRPEAP*, an alternative attribution backbone we develop that keeps PEAP's position-aware edge formulation but replaces the gradient-based backward with an LRP-rule backward (Jafari et al., 2025) (Appendix O).

Finding 3: The judge circuit and its Latent Evaluator are stable across attribution backbones.

On the (Qwen2.5-7B, Gemma-3-12B) $\times$ 10-task panel, top-200 PEAP and LRPEAP edge sets share 34% mean Jaccard IoU on edges and 46% on components (permutation null $p_{99} = 1.9\%$); the Latent Evaluator subgraph $\mathcal{C}_{\text{LE}} = \mathcal{C}_{\text{rate}} \cap \mathcal{C}_{\text{class}}$ computed under each method recovers at 0.47 mean component IoU, peaking at 0.61 on MNLI. The partial edge-overlap is consistent with *computational redundancy*, where multiple sparse subgraphs implement the same judgment behavior; the LE/TF decomposition is the structural intersection both methods converge on (Appendix O).

4 Judge Circuit Modularity is Architecture-Dependent

Building on §3.2, we test whether the shared trunk is a functionally modular sub-system and not a

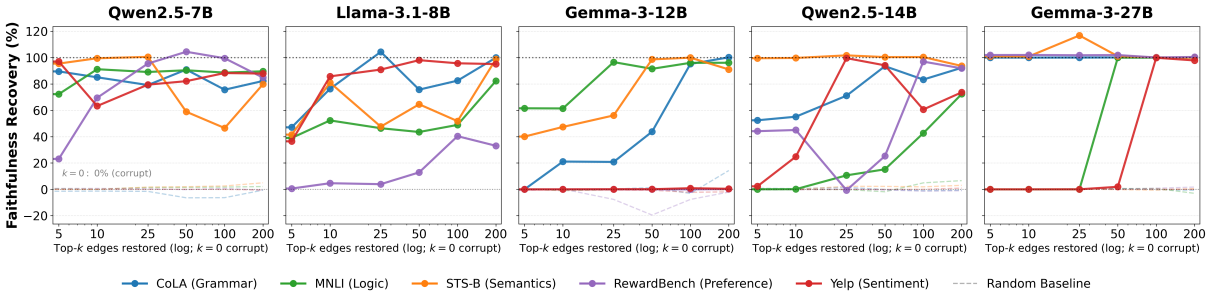


Figure 3: Sparse circuit faithfulness across the five evaluated models and five rating tasks. Each curve traces median MIB recovery as we cumulatively patch the top-k PEAP edges from a fully corrupted forward pass back toward the clean activations. Solid colored lines are the discovered circuits; the gray dashed line is a random-edge baseline. Curves saturating at ≈ 1.0 at small k indicate that the sparse circuit fully captures the model's evaluation behavior; flat curves (Gemma-3-12B / RewardBench, Yelp; Llama-3.1-8B / Yelp) reflect architectural entanglement on those particular cells rather than an absence of mechanism.

generic capability bottleneck (Hanna et al., 2026): if zero-ablating the Latent Evaluator collapses judgment but spares world-knowledge benchmarks, the sub-graph is doing genuinely judgment-specific work – which in turn licenses the format-transfer experiments in §5 as targeted perturbations.

4.1 Isolating Judgment from Formatting via Contrastive Circuits

For each dataset we trace two circuits on the same data: one for the rating task ($\mathcal{C}_{\text{rate}}$, e.g., "On a scale of 1 to 5...") and one for a classification control task ($\mathcal{C}_{\text{class}}$, e.g., yes/no) with matched prompt structure. Their structural overlap decomposes the model's cognition into two functionally distinct components:

- **The Latent Evaluator** ($\mathcal{C}_{\text{LE}} := \mathcal{C}_{\text{rate}} \cap \mathcal{C}_{\text{class}}$): the shared computational trunk. Components in this intersection process the core semantic judgment of the prompt, agnostic to output format. $\mathcal{C}_{\text{LE}}$ is the formal definition of the $\mathcal{C}_{\text{shared}}$ sub-circuit highlighted in Figure 1.
- **The Task Formatters** ($\mathcal{C}_{\text{TF,rate}} := \mathcal{C}_{\text{rate}} \setminus \mathcal{C}_{\text{class}}$ and $\mathcal{C}_{\text{TF,class}} := \mathcal{C}_{\text{class}} \setminus \mathcal{C}_{\text{rate}}$): specialized terminal routing branches, typically late-layer attention heads, that translate the abstract judgment into format-specific target tokens.

The judge circuit (§3) is therefore $\mathcal{C}_{\text{rate}} = \mathcal{C}_{\text{LE}} \cup \mathcal{C}_{\text{TF,rate}}$. We abbreviate Latent Evaluator and Task Formatter as LE and TF.

Finding 4: Contrastive tracing yields a clean Latent Evaluator / Task Formatter decomposition.

On Gemma-3-12B (CoLA $\times$ CoLA_CLASS, top-200), 3 of 17 analyzed heads act as shared evaluators – most strongly L45H3, L46H12, L47H7 – while the remaining 14 split cleanly into rating-specific formatters (9) and classification-specific formatters (5); Figure 2 (middle) visualizes the same $\mathcal{C}_{\text{LE}}$ on one STS-B pair. Per-head PEAP attribution analysis (Appendix J) confirms the functional distinctiveness of these shared evaluator heads through direct visualization of information flow patterns, providing further evidence that the LE/TF decomposition reflects genuine architectural structure rather than an artifact of the circuit intersection criterion. A parallel MLP-level cross-check via transcoder features at the PEAP-identified MLP positions (Appendix K) shows a two-level output-preparation regime stabilizing at L25 on CoLA and after a transitional L22–L24 stage on STS-B, with a shared numerical-output cluster firing across all prompts and prompt-specific value-encoding features emerging by L47 – consistent with the LE/TF split observed at the head level. A complementary layer-wise Edge IoU decomposition (App. P) isolates the TF-attributable signal as the (sender, receiver) cells where same-format circuits overlap but the matched cross-format pair does not; this ΔIoU concentrates on the mid-to-late diagonal band and narrows with model scale.

4.2 Functional Modularity via Zero-Ablation

Identifying a shared causal circuit does not guarantee that the Latent Evaluator is *functionally* isolated from unrelated capabilities such as world-knowledge recall. We test this by zero-ablating the Latent Evaluator: for every component (attention head or MLP) that appears as a sender in at least one top-k Latent Evaluator edge, we clamp its forward-pass output to zero. We then evaluate the ablated model against MMLU (Hendrycks et al., 2021) world knowledge and two factual

Model size	Clinical DB	Abstract Alg.	Physics	StrategyQA	CREAK	Status
Qwen2.5-7B	76% → 76%	57% → 57%	57% → 59%	68.5% → 68.0%	64.5% → 63.0%	**Modular**
Llama-3.1-8B†	75% → 75%	35% → 35%	43% → 43%	66.5% → 66.5%	48.5% → 48.5%	**Modular**
Gemma-3-12B	81% → 19%	45% → 21%	48% → 23%	64.5% → 61.5%	39.0% → 23.5%	Entangled
Qwen2.5-14B	87% → 87%	59% → 59%	69% → 69%	72.0% → 72.0%	63.5% → 63.5%	**Modular**
Gemma-3-27B	79% → 79%	51% → 51%	60% → 60%	53.0% → 53.0%	57.0% → 57.0%	**Modular**

Table 1: Zero-ablation semantic domain control. Ablating the Latent Evaluator collapses world knowledge (MMLU: Clinical DB, Abstract Alg., Physics) and formal factual retrieval (StrategyQA, CREAK) in Gemma-3-12B, but preserves both across the four other models – indicating that modularity depends on architecture, not scale alone. The full circuit-topology panels in Appendix L demonstrate the corresponding two-stage Latent Evaluator / Task Formatter separation across models and tasks, supporting the same generalization. † Llama-3.1-8B's merged top-50 Latent Evaluator contains only MLPs (no shared attention heads); the StrategyQA/CREAK cells reflect the meaningful MLP-only ablation, while the MMLU cells are vacuously preserved because the head-targeted MMLU runner had no heads to ablate – consistent with Llama's MLP-dominant evaluator (Appendix L).

QA datasets, StrategyQA (Geva et al., 2021) and CREAK (Onoe et al., 2021). These probes natively emit "Yes/No" or "True/False" tokens – mirroring our Task Formatter setups – while relying on disjoint semantic phenomena (factual retrieval vs. abstract judgment).

Finding 5: On modular architectures, ablating the Latent Evaluator leaves world knowledge intact.

Table 1 illustrates that, on the four modular models, every meaningfully-tested probe shows ≤ 2 pp degradation under ablation.[3] By contrast, iteratively ablating Latent Evaluator edges in the same models triggers a phase-transition collapse in judgment EV on every rating task tested (Appendix H, Figures 27–28). The Latent Evaluator therefore operates as a specialized sub-system whose removal destroys judging while leaving the model's world knowledge stores largely intact.

Finding 6: Modularity emerges at family-specific scales.

Qwen achieves clean modularity already at the smallest size we study (7B); Llama-3.1-8B does so for its MLP-dominant evaluator; Gemma-3 only at 27B. In contrast, Gemma-3-12B tightly entangles the Latent Evaluator with world-knowledge pathways: zero-ablation roughly halves MMLU clinical, physics, and CREAK accuracy (Table 1). Only when we scale to Gemma-3-27B does this entanglement dissolve. Scale alone therefore does not predict modularity: comparable parameter counts produce qualitatively different internal structure across families.

[3]Caveat: Llama-3.1-8B's MMLU cells are vacuously preserved since its merged Latent Evaluator contains no shared attention heads, making the head-only MMLU runner inert; the meaningful Llama tests are the StrategyQA and CREAK cells, which ablate the LE MLPs and show 0 pp drop.

5 Inter-Format Inconsistencies Arise from a Modular Mismatch

Given that the Latent Evaluator is a real, functionally modular sub-system, the question is how its output is transformed into the format-specific target token. Our hypothesis is that the *Task Formatter* branches (§4.1) are the locus of inter-format inconsistency: the Latent Evaluator computes a stable continuous judgment signal, but this signal is mapped onto format-specific tokens by fragile, non-linear terminal routing. We test this hypothesis via a causal cross-format patching experiment.

Causal Analysis via Format Transfer Injection We design a minimal causal test: *Format Transfer Injection* (FTI) following Merullo et al. (2024). For a given instance we capture the activations of the Latent Evaluator components during a pristine 5-*star rating* prompt and force those exact activations – a *blanket* activation transfer that overwrites the entire LE pattern, in contrast to the targeted 1D subspace interventions of Appendix D – into the computational graph of the same model running on a corrupted *classification* prompt (whose natural output would be the negative token, e.g., "No"). If the Latent Evaluator is the primary causal anchor for the judgment, the downstream classification head should receive the injected positive judgment signal and flip its output token – from "No" to "Yes" or "Entailment". If instead the terminal branches are doing the actual judgment work, the injection should have no effect. This blanket-transfer protocol contrasts with the targeted 1D subspace injection at the same LE components

Model	Dataset	Base Prob.	Patched Prob.	Cat. Flip Rate
Qwen2.5-7B	CoLA ($N = 26$)	17.4% ± 14.9%	93.5% ± 3.9%	100.0%
	MNLI ($N = 324$)	3.4% ± 6.3%	88.0% ± 10.5%	99.1%
	STS-B ($N = 134$)	1.6% ± 2.2%	85.6% ± 9.6%	100.0%
	Yelp ($N = 437$)	0.1% ± 0.4%	59.5% ± 38.5%	66.4%
	RewardBench ($N = 84$)	1.6% ± 5.1%	96.9% ± 10.1%	**100.0%**
Llama-3.1-8B	CoLA ($N = 8$)	28.6% ± 7.6%	41.7% ± 13.9%	25.0%
	MNLI ($N = 120$)	10.4% ± 12.6%	13.1% ± 17.0%	4.2%
	Yelp ($N = 94$)	7.1% ± 7.0%	11.1% ± 5.4%	0.0%
	RewardBench ($N = 9$)	26.1% ± 18.3%	39.6% ± 20.8%	22.2%
Gemma-3-12B	CoLA ($N = 47$)	6.1% ± 11.8%	17.6% ± 26.8%	14.9%
	MNLI ($N = 402$)	6.3% ± 11.0%	11.8% ± 15.9%	3.7%
	STS-B ($N = 44$)	15.1% ± 13.6%	53.5% ± 33.1%	50.0%
	Yelp ($N = 92$)	0.0% ± 0.1%	0.0% ± 0.1%	0.0%
	RewardBench ($N = 26$)	7.2% ± 12.4%	10.7% ± 17.3%	7.7%
Qwen2.5-14B	CoLA ($N = 12$)	24.3% ± 14.1%	69.0% ± 18.7%	91.7%
	MNLI ($N = 125$)	4.2% ± 9.4%	4.9% ± 14.2%	4.0%
	STS-B ($N = 156$)	0.0% ± 0.5%	16.0% ± 16.0%	5.1%
	Yelp ($N = 189$)	0.4% ± 0.8%	41.0% ± 16.9%	31.2%
	RewardBench ($N = 51$)	1.5% ± 5.0%	43.4% ± 36.4%	41.2%
Gemma-3-27B	CoLA ($N = 54$)	8.6% ± 10.6%	9.9% ± 11.2%	0.0%
	MNLI ($N = 151$)	8.0% ± 11.9%	8.0% ± 11.9%	0.7%
	STS-B ($N = 31$)	30.5% ± 7.3%	78.1% ± 13.7%	**96.8%**
	Yelp ($N = 189$)	1.3% ± 3.2%	3.2% ± 5.9%	1.1%
	RewardBench ($N = 19$)	20.8% ± 18.1%	22.1% ± 20.7%	5.3%

Table 2: FTI probability shifts on all five tasks. Patching a 5-star Latent Evaluator into a corrupted categorical classification prompt shifts probability mass toward the positive target token (Yes/Entailment) when the Task Formatter is geometrically compatible. N is the post-filter pair count under the inclusion criteria (source rating EV > 4, corrupted base prediction $\notin$ {Yes, Entailment}); per-cell discussion in §5.1.

(App. D); §5.2 develops the *scalar-vs-blanket* distinction as a deployment-relevant design property of the formatter.

5.1 The Latent Evaluator is the Causal Anchor for Judgment

Finding 7: Injecting the Latent Evaluator causally shifts downstream classifier outputs; inter-format inconsistency therefore localizes to the classification Task Formatter, not the Latent Evaluator.

The clearest causal demonstration is Qwen2.5-7B: blanket FTI flips the argmax in $\geq 99\%$ of CoLA, STS-B, MNLI, and RewardBench pairs (Table 2), with mean target-class probability rising from $\leq 17\%$ at baseline to $\geq 85\%$ post-injection; the same near-total flips hold on Gemma-3-27B / STS-B and Qwen2.5-14B / CoLA. The injected continuous judgment scalar is decoded by the classification formatter onto the discrete target token "Yes"/"Entailment" without breaking the output format space (no pair emits "5"): the judgment representation is stable, 1D, and shared across semantic domains (§4.1, App. D); the bottleneck is the terminal mapping, which varies sharply across tasks (3-way MNLI vs. binary classification) and across models (geometrically *insulated*, e.g., Gemma-3-27B $\times$ MNLI, vs. *exposed*, e.g., Qwen2.5-7B).

Finding 8: FTI fails when the formatter is geometrically insulated – either by scale (open-ended tasks) or by multi-attractor label structure (MNLI).

(a) Multi-attractor label structure (MNLI). We apply Logit Lens[4] – which decodes intermediate activations into vocabulary space via the model's unembedding matrix W_U – to the late-layer Task Formatter components of Gemma-3-27B (App. I, Figure 11). The projection shows MNLI's classification formatter spreading its projected mass across three competing target tokens (*contradiction*, *entailment*, *neutral*; max/min ratio ≈ 2.7), while STS-B's formatter concentrates on a single token (ratio ≈ 19). This geometric difference *predicts* the FTI behavior: with the exception of Qwen2.5-7B (which flips MNLI near-perfectly), MNLI flip rates collapse to single digits on every other model (Table 2). The 1D judgment direction has no unambiguous target in a three-attractor basin, so the injected mass fragments and no single label reaches argmax.

(b) Within-family scale decrease (open-ended tasks). On both RewardBench and Yelp the FTI flip rate falls as Qwen scales from 7B to 14B and as Gemma-3 scales from 12B to 27B (Table 2). The trend is not an LE absence at scale, since sparse circuits still saturate MIB faithfulness (App. C) and 1D steering still moves the output (App. D). Instead, the open-ended Task Formatter becomes *geometrically insulated*: the scalar judgment direction is steerable, but the full rating-prompt LE pattern is no longer a sufficient causal key for the classification-prompt formatter to accept.

5.2 The Format Split is the Inconsistency Bottleneck

The FTI results close the causal loop on our third contribution: the Latent Evaluator's output, a 1D direction tracking the scaled rating signal (App. D), is universally received by downstream Task Formatters, but only reaches the argmax token when the TF's attractor geometry is compatible with a scalar input (Table 2, App. I). On RewardBench/Yelp, cumulative patching (App. C) and targeted 1D subspace steering (App. D) move the output cleanly while blanket FTI flips only a minority of instances. We read this *scalar-vs-blanket* divergence as evidence that the formatter's basin becomes more

[4] https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru

selective with scale, so deploying the LE as a robust LaaJ signal on open-ended tasks favors *targeted* subspace interventions over *blanket* activation transfer.

Finding 9: The LE's 1D direction is a usable zero-shot judgment scalar in the small-N preference regime.

A zero-shot 1D readout (BDAS-1D) at the LE site – whose per-pair dose-response Figure 2 (bottom) illustrates – tracks a fully supervised residual probe (Girrbach et al., 2025) within a few points of Spearman ρ on most cells and matches or exceeds it on small-N preference data, while beating prompted argmax in nearly every cell; on scale-aligned vocabularies (Yelp 1–5), prob-weighted EV remains the stronger baseline (App. E).

6 Discussion

The Latent Evaluator / Task Formatter split reframes the ongoing debate about LLM-as-a-judge reliability (Lee et al., 2025; Bavaresco et al., 2025; Chehbouni et al., 2025). Behavioral inconsistency under format perturbations is, on the mechanism we identify, the expected signature of a stable internal judgment routed through a fragile terminal mapping and not a failure of the underlying evaluation. This shifts the diagnostic question from *"does the model judge consistently?"* to *"does the formatter for this output specification preserve the underlying judgment?"*, and it predicts that benchmark-level reliability comparisons across formats are partially measuring formatter geometry as opposed to evaluation quality.

A second implication: comparable parameter counts produce qualitatively different internal structure across families, pushing back on the assumption that clean internal abstractions emerge as a generic consequence of scale. Which factor (pretraining data, post-training, attention sparsity, normalization placement) drives the Qwen vs. Gemma scale contrast is a natural follow-on.

Mechanism connects to practice through a regime caveat: latent signals outperform prompted Likert outputs (Girrbach et al., 2025), and we causally identify the subspace from which those signals are recovered. The advantage concentrates on small-N preference data where the discrete output is poorly calibrated and supervised probes overfit; on scale-aligned vocabularies, prompted aggregation remains the stronger baseline.

A natural open question is whether the two-step pattern we identify – a stable internal computation routed through fragile terminal pathways – recurs in other behaviors where output formatting matters (e.g., chain-of-thought, structured generation, tool calling). If it does, the LaaJ inconsistency we mechanistically pin down here would be one instance of a broader routing-vs-computation dissociation worth probing in those settings.

7 Related Work

Behavioral critiques of LaaJ validity. Beyond the inter-format inconsistencies established by Lee et al. (2025) and the shortcut-exploitation results of Eshuijs et al. (2025), Chehbouni et al. (2025) challenge the fundamental validity of LaaJ protocols, arguing that even strong models lack the robustness required to evaluate abstract concepts reliably. Bavaresco et al. (2025) corroborate this empirically in a large-scale comparison, finding that no single LLM consistently aligns with human judgment across tasks. Our mechanistic results refine these critiques: under the LE/TF split, much of the observed inconsistency localizes to the terminal formatting stage. Benchmark-level reliability comparisons across formats are partially measuring formatter geometry rather than evaluation quality.

Mechanistic precedents. Our work joins three lines of evidence: cross-task circuit overlap (Tigges et al., 2024; Ferrando and Costa-jussà, 2024; Lan et al., 2024), low-rank linear intermediate variables (Lepori et al., 2024; Mueller et al., 2026), and the formal/functional dissociation (Hanna et al., 2026) that the LE/TF split mirrors. We extend this lineage by causally validating cross-format judgment via subspace steering and activation transfer.

8 Conclusion

LLM judgment reliability depends not only on what models compute internally but on how that computation is routed to the output token. We identify a compact *Latent Evaluator* in mid-to-late MLPs that is functionally modular on most architectures we study but entangled with world-knowledge pathways on Gemma-3-12B, so modularity is architecture-dependent rather than a consequence of scale. The 1D causal direction underlying this sub-graph recovers a supervised linear-probe judgment signal zero-shot and exceeds it on small-N preference data, mechanistically locating the latent signal that practical reference-free rating methods rely on.

Limitations

A primary limitation of our mechanistic investigation stems fundamentally from the **computational geometry constraints** of tracing extensive architectures end-to-end. For context, natively mapping the CoLA judgment computational graph in Gemma-3-12B requires evaluating approximately 1.46 million candidate edges. While PEAP allows for tracing these evaluation circuits across temporal dimensions efficiently, performing such densely scaled edge patching computations – especially over the largest model variations like Gemma-3-27B (incorporating roughly 50,000 components) – strictly required us to constrain our analyzed dataset subset bounds to between 100 and 500 distinct samples representing minimal pairing. We partially mitigate this concern by reporting split-half circuit reliability (Appendix M): within-task circuits are substantially more stable than chance at every scale we tested, and comparable to or higher than the cross-task IoU numbers we report in §3.2.

Furthermore, while we show our principles across evaluations like grammar, logical entailment, sentiment and preference, **mapping exactly how models route highly subjective** or culturally biased evaluation metrics remains a compelling horizon for future research. The open-ended-task scope concern that RewardBench and Yelp circuits might require a denser subgraph than structured NLU is largely resolved by the present data: on Qwen2.5-7B, Qwen2.5-14B, and Gemma-3-27B the same sparse edge budget that recovers structured NLU also recovers open-ended judgment (Appendix C, Figure 3). The exceptions are Gemma-3-12B (where neither RewardBench nor Yelp saturates) and Llama-3.1-8B on Yelp alone (median recovery peaks at ≈ 0.40 before drifting back down); we attribute the Gemma-3-12B failure to its architectural entanglement (Table 1) and the Llama-3.1-8B Yelp shortfall to its weaker cross-task structural overlap and MLP-dominant Latent Evaluator (§3.2, Appendix L).

A more nuanced limitation concerns the **scalar-vs-blanket FTI decoupling** on open-ended tasks at scale, developed in §5.2. Pinning down which properties of the rating-prompt activation geometry are and are not carried across the FTI injection – beyond the 1D judgment direction itself – is a direction for future mechanistic work.

The **practical-judge result** (Appendix E) carries a regime caveat: when the prompted vocabulary is scale-aligned to the human label (Yelp 1–5 stars), prob-weighted EV is a strong baseline that the 1D BDAS readout does not exceed. Whether higher-rank extraction (e.g., k-D BDAS or multi-component aggregation) closes that gap is left to future work.

Our **cross-method confirmation** operates at two levels: an attribution-backbone comparison via LRPEAP (Appendix O), which substitutes RelP's LRP-rule backward (Jafari et al., 2025) into PEAP's position-aware edge formulation and recovers the same mid-to-late LE concentration with a $\sim$ 13–25$\times$ enrichment over the permutation null; and component-level cross-checks via the per-head PEAP attribution analysis (Appendix J) and the transcoder feature analysis at circuit-identified MLP positions (Appendix K). What remains absent is a direct comparison against a structurally distinct discovery algorithm such as ACDC, whose iterative edge-pruning objective differs more sharply from PEAP than LRPEAP's rule-level substitution; running ACDC on the same minimal-pair sets is the natural follow-on probe. The transcoder analysis (Appendix K) is limited to the 16k-feature Gemma-Scope-2 transcoders (McDougall et al., 2025) due to memory constraints; the 262k-feature variants would provide finer-grained feature decomposition but were not feasible to run at the scale of our circuit analysis.

The **cross-task IoU** values reported in §3.2 are bracketed by within-task split-half reliability (an upper bound) and a random-edge baseline (a lower bound), both on judgment circuits; we do not include an IoU comparison against a circuit traced on a non-judgment task (e.g., factual recall on MMLU) as a non-LaaJ external reference, which would further sharpen the interpretation of the LaaJ shared-trunk magnitude.

Ethical Statement

AI assistance (Claude Code) was used for coding and minor textual edits. All scientific claims, interpretations, and conclusions remain the responsibility of the authors.

References

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Bar-

bara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria. Association for Computational Linguistics.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Nitay Calderon, Roi Reichart, and Rotem Dror. 2025. The alternative annotator test for LLM-as-a-judge: How to statistically justify replacing human annotators with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16051–16081, Vienna, Austria. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Khaoula Chehbouni, Mohammed Haddou, Jackie CK Cheung, and Golnoosh Farnadi. 2025. Neither valid nor reliable? investigating the use of LLMs as judges. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems Position Paper Track*.

Dami Choi, Vincent Huang, Kevin Meng, Daniel D Johnson, Jacob Steinhardt, and Sarah Schwettmann. 2024. Scaling automatic neuron description. https://transluce.org/neuron-descriptions.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. 2024. Transcoders find interpretable LLM feature circuits. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Leon Eshuijs, Shihan Wang, and Antske Fokkens. 2025. Short-circuiting shortcuts: Mechanistic investigation of shortcuts in text classification. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 105–125, Vienna, Austria. Association for Computational Linguistics.

Javier Ferrando and Marta R. Costa-jussà. 2024. On the similarity of circuits across languages: a case study on the subject-verb agreement task. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10115–10125, Miami, Florida, USA. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Mingqi Gao, Xinyu Hu, Xunjian Yin, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2025. LLM-based NLG evaluation: Current status and challenges. *Computational Linguistics*, 51:661–687.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Leander Girrbach, Chi-Ping Su, Tankred Saanum, Richard Socher, Eric Schulz, and Zeynep Akata. 2025. Reference-free rating of llm responses via latent information. *arXiv*, abs/2509.24678.

Elena Golimblevskaia, Aakriti Jain, Bruno Puri, Ammar Ibrahim, Wojciech Samek, and Sebastian Lapuschkin. 2026. Circuit insights: Towards interpretability beyond activations. In *The Fourteenth International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv*, abs/2407.21783.

Tal Haklay, Hadas Orgad, David Bau, Aaron Mueller, and Yonatan Belinkov. 2025. Position-aware automatic circuit discovery. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2792–2817, Vienna, Austria. Association for Computational Linguistics.

Michael Hanna, Yonatan Belinkov, and Sandro Pezzelle. 2026. Are formal and functional linguistic mechanisms dissociated in language models? *Computational Linguistics*, pages 1–41.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Farnoush Rezaei Jafari, Oliver Eberle, Ashkan Khakzar, and Neel Nanda. 2025. Relp: Faithful and efficient circuit discovery in language models via relevance patching. *arXiv*, abs/2508.21258.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico. Association for Computational Linguistics.

Michael Lan, Philip Torr, and Fazl Barez. 2024. Towards interpretable sequence continuation: Analyzing shared circuits in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12576–12601, Miami, Florida, USA. Association for Computational Linguistics.

Noah Lee, Jiwoo Hong, and James Thorne. 2025. Evaluating the consistency of LLM evaluators. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10650–10659, Abu Dhabi, UAE. Association for Computational Linguistics.

Michael A. Lepori, Thomas Serre, and Ellie Pavlick. 2024. Uncovering intermediate variables in transformers using circuit probing. In *First Conference on Language Modeling*.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *Preprint*, arXiv:2412.05579.

Callum McDougall, Arthur Conmy, János Kramár, Tom Lieberum, Senthooran Rajamanoharan, and Neel Nanda. 2025. Gemma scope 2: Suite of SAEs and transcoders for Gemma 3.

Maxime Méloux, Silviu Maniu, François Portet, and Maxime Peyrard. 2025. Everything, everywhere, all at once: Is mechanistic interpretability identifiable? In *The Thirteenth International Conference on Learning Representations*.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Language models implement simple Word2Vec-style vector arithmetic. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5030–5047, Mexico City, Mexico. Association for Computational Linguistics.

Joseph Miller, Bilal Chughtai, and William Saunders. 2024. Transformer circuit evaluation metrics are not robust. In *First Conference on Language Modeling*.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. 2026. The quest for the right mediator: Surveying mechanistic interpretability for nlp through the lens of causal mediation analysis. *Computational Linguistics*, pages 1–48.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, and 4 others. 2025. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*.

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

Yasumasa Onoe, Michael Zhang, Eunsol Choi, and Greg Durrett. 2021. Creak: A dataset for commonsense reasoning over entity knowledge. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Bruno Puri, Aakriti Jain, Elena Golimblevskaia, Patrick Kahardipraja, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. 2025. FADE: Why bad descriptions happen to good features. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17138–17160, Vienna, Austria. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *arXiv*, abs/2412.15115.

Andres Saurez, Neha Sengar, and Dongsoo Har. 2026. Circuit fingerprints: How answer tokens encode their geometrical path. *arXiv*, abs/2602.09784.

Aaquib Syed, Can Rager, and Arthur Conmy. 2024. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 407–416, Miami, Florida, US. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *arXiv*, abs/2503.19786.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. Llm circuit analyses are consistent across training and scale. In *Advances in Neural Information Processing Systems*, volume 37, pages 40699–40731. Curran Associates, Inc.

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. 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*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2024. Jump to conclusions: Short-cutting transformers with linear transformations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9615–9625, Torino, Italia. ELRA and ICCL.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

A PEAP Attribution Formulas

This appendix contains the exact attribution-score formulas referenced in §3.1. For each candidate edge, PEAP approximates the causal effect of restoring that edge from a corrupted to a clean state via a linear first-order expansion. Let $\mathrm{EV} = \sum_{r=1}^{s} r \cdot P(\text{rating} = r)$ denote the expected value of the predicted rating distribution (where $P(\text{rating} = r)$ is the softmax over the rating-token logits at the final sequence position and s is the upper bound of the rating scale), and let

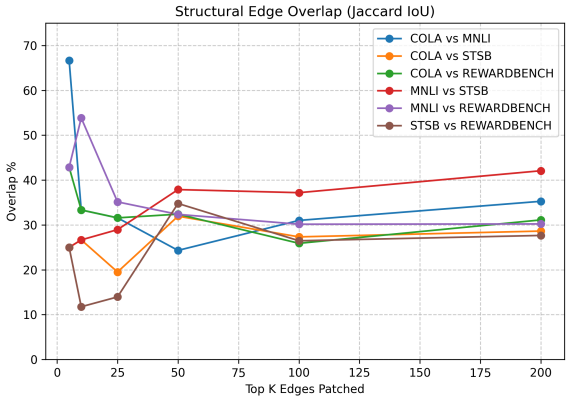


Figure 4: Cross-task Edge IoU on Gemma-3-12B across Top-K patching thresholds. Edges are PEAP-attributed connections between sub-components; higher curves indicate more shared structure at a given sparsity.

$m = \mathrm{sgn}(\mathrm{EV}_{\text{clean}} - \mathrm{EV}_{\text{corr}})$ be the per-pair polarity multiplier that keeps attributions directionally consistent across our symmetrically balanced minimal pairs (§2).

For intra-token residual-stream communication between sender S_i and receiver R_j at the same token position ($i = j$), the attribution score is

$$\mathrm{Score}(S_i \to R_i) = m \cdot \left((S_i^{\text{clean}} - S_i^{\text{corr}}) \cdot \nabla R_i \right).$$

For cross-token edges ($i \neq j$) we capture the Attention mechanism's *crossing edges* in the PEAP formulation, treating the Value vector V at the source token as the sender, the Attention Output Z at the destination token as the receiver, and scaling by the Attention Pattern A:

$$\mathrm{Score}(V_i \to Z_j) = m \cdot A_{j,i} \cdot \left((V_i^{\text{clean}} - V_i^{\text{corr}}) \cdot \nabla Z_j \right).$$

The Value/Output decomposition follows the original PEAP formulation (Haklay et al., 2025); our contribution is the symmetric polarity correction m, which adapts PEAP to the bidirectional rating targets inherent to LLM-as-a-judge evaluation. A single backward pass on the corrupted prompt yields all ∇R and ∇Z terms simultaneously, so an entire circuit's attribution is extracted in one forward–backward sweep per minimal pair.

B Cross-task Node Overlap

Figure 6 reports the Node IoU complement to the Edge IoU view in Figure 4 (§3.2). Node IoU measures architectural recruitment at the granularity of attention heads and MLPs, ignoring the specific cross-token connections that Edge IoU restricts to.

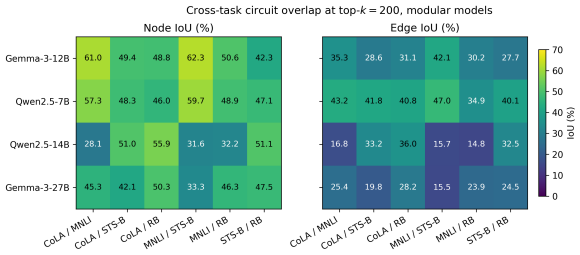


Figure 5: Cross-task circuit overlap at top-200 across all four architecturally modular models. The shared trunk is recoverable on every model, but the per-pair magnitude reflects each model's circuit redundancy (§3.2, App. M): smaller models route through fewer equivalent paths, so their top-200 edges are more conserved across tasks, while larger modular models distribute attribution across many equivalent sub-pathways, lowering raw Edge IoU even though Node IoU stays high.

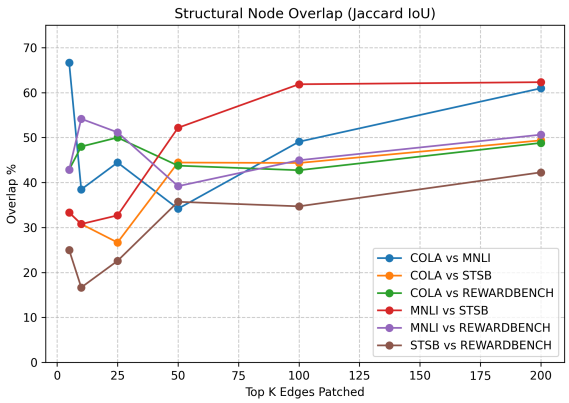


Figure 6: Cross-task Node IoU on Gemma-3-12B across Top-K patching thresholds. Companion to Figure 4.

Across all task pairs the Node IoU curve sits substantially above the corresponding Edge IoU curve at the same Top-K, reflecting that distinct tasks reuse the same physical sub-components while routing through partially distinct edge subsets.

C Circuit Faithfulness

C.1 Methodology

Circuit faithfulness – the degree to which a discovered subgraph causally accounts for the target behavior – is notoriously fragile and highly sensitive to seemingly insignificant changes in the ablation methodology (e.g., node vs. edge patching) (Miller et al., 2024). A parallel concern is non-identifiability: multiple incompatible circuits can artificially explain the same downstream behavior (Méloux et al., 2025). We therefore adopt the per-instance MIB formulation throughout the main body and report a sensitivity analysis against the legacy magnitude-weighted directional score in Appendix N, and we cross-validate the resulting circuits with two independent causal probes (BDAS, Appendix D; FTI, §5.1) to guard against accepting a circuit that is faithful under one metric but spurious under another.

To validate that the edges identified by PEAP are *sufficient* for eliciting the judge behavior, we evaluate the faithfulness of the extracted circuit via cumulative patching (Syed et al., 2024; Hanna et al., 2024). Starting from a fully corrupted forward pass, we progressively restore the activations of the top-k edges (ranked by absolute PEAP score) to their clean-state values. Restoration is applied only at the exact token positions dictated by each edge.

Following the MIB benchmark (Mueller et al., 2025), we define the faithfulness of a sparse circuit $\mathcal{C}_k$ (the sub-circuit containing the top-k attributed edges) as the mean per-instance fraction of the clean–corrupted EV gap that the patched circuit recovers:

$$\mathrm{Faith}(k) = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathrm{EV}^{(i)}(\mathcal{C}_k) - \mathrm{EV}^{(i)}_{\mathrm{corr}}}{\mathrm{EV}^{(i)}_{\mathrm{clean}} - \mathrm{EV}^{(i)}_{\mathrm{corr}}}.$$

A faithfulness score near 1.0 indicates that $\mathcal{C}_k$ fully encapsulates the model's rating behavior. Because our minimal pairs are symmetrically balanced, each per-instance gap carries an intrinsic sign and the per-instance ratio handles polarity naturally without an explicit direction multiplier. Treating every pair equally also avoids magnitude-weighting artifacts that would let a small number of high-gap pairs dominate the aggregate. We report the *median* across minimal pairs as our primary statistic, since the ratio distribution is heavy-tailed when a minority of pairs have near-equal clean/corrupt EVs; the mean, 95% bootstrap CI, and the count of low-gap pairs skipped via `mib_min_gap`$= 0.05$ are all reported in the supplementary CSVs. A legacy magnitude-weighted directional formulation is reported in Appendix N as a sensitivity analysis.

C.2 Results

Per-cell saturation points. The headline 21-of-25 finding is summarized in §3.3; here we report the representative per-cell saturation points behind it. On Gemma-3-12B, median faithfulness saturates at 0.96 on MNLI at $k = 25$, 1.00 on STS-B at $k = 50$, and 0.95 on CoLA at $k = 100$. On Gemma-3-27B, median recovery snaps to ≈ 1.00 at $k \geq 50$ on the four structured/open-ended tasks; the RewardBench circuit in particular saturates

at median 1.02 with just $k = 5$ edges – an extreme sparsity that we partly attribute to Gemma-3-27B's clean modularity (Table 1) concentrating the open-ended-task circuit into a very small number of highly-attributed edges.

Interpreting the shape of the curves. Two curve shapes in Figure 3 deserve explicit comment. First, a handful of cells – most prominently Gemma-3-27B × RewardBench (median 1.02 at $k = 5$) – reach full recovery essentially at the sparsest budget we probe. This is not a metric artifact: the per-instance ratio $(\mathrm{EV}(\mathcal{C}_k) - \mathrm{EV}_{\text{corr}})/(\mathrm{EV}_{\text{clean}} - \mathrm{EV}_{\text{corr}})$ with `mib_min_gap`$= 0.05$ is bounded below by the k=0 corruption floor and takes no shortcuts; the shape reflects the underlying attribution distribution. When a model is architecturally modular (§4.2) and the task is decoded through a shallow, terminal Task Formatter – as RewardBench is on Gemma-3-27B, where the binary preference-scoring token sits directly after a short helpful/aligned instruction – the causal work concentrates into a few deep-layer edges, and cumulative patching recovers the clean EV as soon as those edges are restored. Conversely, structured NLU tasks such as MNLI, which requires cross-referencing premise and hypothesis spans, distribute attribution across more edges and therefore climb more gradually through $k \in [10, 100]$ before saturating. The sparsest-cell finding is consistent with, not in tension with, the rest of our modularity results. Second, the Gemma-3-12B open-ended curves remain flat through $k = 200$. This is the entanglement regime documented in Table 1: PEAP still *localizes* stable open-ended edges on Gemma-3-12B (its split-half IoU on Yelp is 22.4% and on RewardBench is 25.6%, well above the 0.5–6.8% random baseline in Appendix M), but the top-200 subgraph is not sparse-recoverable because the circuit is densely interleaved with world-knowledge pathways. The flat shape therefore encodes an architectural property of Gemma-3-12B rather than an absence of mechanism; we treat it as a bounded scope condition on the sparse-circuit claim and say so explicitly in the Limitations (§8).

D Causal Subspace Steering of the Latent Evaluator

D.1 Methodology

Beyond isolating physical components, recent mechanistic work investigates how concepts are *encoded* inside identified circuits through linear vector arithmetic (Merullo et al., 2024) and through linearly-steerable conceptual variables that route latent states into specific geometric output fingerprints (Mueller et al., 2026; Saurez et al., 2026). Wu et al. (2023) formalized Interchange Intervention Training (IIT) and Distributed Alignment Search (DAS) grounded in causal abstraction, discovering alignments between interpretable abstract variables and distributed neural representations; complementarily, Girrbach et al. (2025) provide independent behavioral evidence that probability-weighted scores and linear probes on rating-position activations outperform prompted Likert outputs, indicating that judgment is encoded in a steerable latent representation.

To probe whether the Latent Evaluator's judgment signal lies along a single steerable direction, we apply a directional mean-difference steering protocol at the PEAP-discovered LE component positions, oriented toward positive judgment. For each source-task minimal pair $(x_{\text{clean}}, x_{\text{corr}})$ we cache activations at every hook position (ℓ, p, h) identified as a Latent Evaluator sender and compute the per-pair difference $\Delta = a_{\text{clean}} - a_{\text{corr}}$. We orient Δ toward the positive-judgment pole using the polarity multiplier $m = \mathrm{sgn}(\mathrm{EV}_{\text{clean}} - \mathrm{EV}_{\text{corr}})$ from §3.1 – the same multiplier that keeps PEAP attributions directionally consistent under our symmetric minimal-pair design – and average $m \cdot \Delta$ across all source-task pairs to obtain a per-hook *steering vector* $\bar{v}_{\ell,p,h}$. At inference on the target task, we add $\alpha \cdot \bar{v}_{\ell,p,h}$ to the corresponding hook activation during the forward pass and read off the resulting expected rating value. $\alpha = 0$ recovers the unsteered baseline; $\alpha = 1$ approximates a one-pair clean activation injection; $\alpha = 2$ extrapolates past it. This protocol probes a 1D *linear* characterization of the LE subspace: any single-direction encoding of the judgment signal predicts a smooth, monotonic dose-response in α.

D.2 Results

Finding: The Latent Evaluator's judgment is encoded in a 1D steerable subspace. Across the five models in Table 3, the directional mean-difference vector at the LE components steers the predicted rating from a neutral mid-scale value to a confident ≈ 5 at $\alpha = 2.0$ when the target domain is compatible (CoLA → MNLI, CoLA → STS-B). Qwen2.5-7B – the smallest model – matches Qwen2.5-14B and Gemma-3-27B in

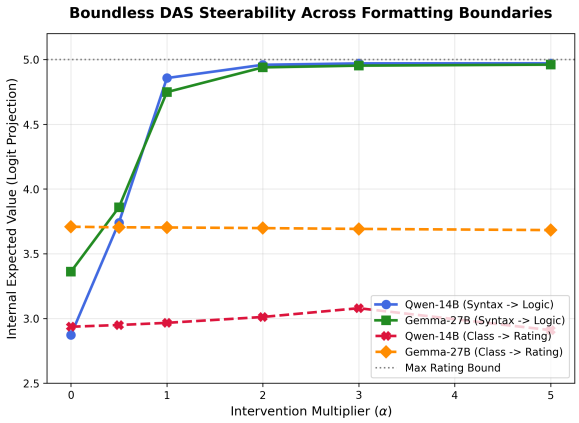


Figure 7: Cross-domain causal steering of the Latent Evaluator. By extracting a 1-dimensional directional steering vector at the LE components on a source domain (e.g., CoLA) and injecting it into the corresponding hooks of a distinct target domain (e.g., MNLI), we control the model's final output. The x-axis denotes the scalar multiplier (α) applied to the targeted subspace intervention, demonstrating bidirectional control over the model's judgment score independent of the underlying geometry.

steered EV precision (4.94 ± 0.05 on CoLA $\rightarrow$ MNLI at $\alpha = 2.0$); Llama-3.1-8B reaches the tightest steered EV in the panel (4.98 ± 0.01 on the same pair). The evidence for a shared *linear* judgment direction is twofold: (i) a steering vector computed on one domain (e.g., CoLA grammar) successfully steers judgment on a structurally unrelated domain (e.g., MNLI entailment), so the direction generalizes across tasks; and (ii) the steering response is monotonic and smooth in α (Figure 7, Figure 8), consistent with a 1D linear encoding rather than a nonlinear or multi-dimensional one.

Finding: DAS fails precisely at the cross-format boundary. Steering between distinct formatting modalities – from a binary classification task to a 5-bucket ordinal rating on STS-B – fails, shifting EV by a statistically negligible margin (Table 3, "Binary $\rightarrow$ Rating" rows). This deliberate negative control reinforces the Latent Evaluator / Task Formatter split: DAS handles the abstract judgment direction but cannot re-route a categorical output into an ordinal one, because that mapping lives in the non-linear terminal Task Formatter. This 1D-geometric finding is independently corroborated by the Format Transfer Injection experiment in §5.1, which uses direct activation patching (rather than a learned rotation) and reaches the same conclusion.

Finding: Subspace steering extends to open-ended tasks on the modular architectures. Supplementing Table 3 with cross-domain steering into RewardBench and Yelp circuits: on Qwen2.5-7B the steering vector from CoLA to RewardBench drives the target EV from a neutral baseline to 4.99 ± 0.02 ($\alpha = 2.0$), and STS-B $\rightarrow$ Yelp reaches 4.87 ± 0.04. Qwen2.5-14B and Gemma-3-27B show the same pattern (Qwen2.5-14B CoLA $\rightarrow$ RewardBench: 4.98 ± 0.01; Gemma-3-27B CoLA $\rightarrow$ RewardBench: 4.92 ± 0.19). Gemma-3-12B, consistent with its entanglement profile, fails all RewardBench-target steering (EV stays at baseline ≈ 0) and shows only partial recovery on Yelp targets. Steering vectors sourced from MNLI into either open-ended target are substantially weaker across all models (e.g., Qwen2.5-14B MNLI $\rightarrow$ RewardBench: 3.15 ± 0.88), mirroring

Model	Action (h $\rightarrow$ Rh)	Orig. EV ($\pm$ SD)	Steered EV ($\pm$ SD)
Qwen2.5-7B	CoLA $\rightarrow$ MNLI	2.75 ± 1.53	**4.94** ± 0.05
	CoLA $\rightarrow$ STS-B	2.86 ± 1.40	**4.95** ± 0.03
	STS-B Binary $\rightarrow$ Rating	2.86 ± 1.40	2.08 ± 0.61
Gemma-3-12B	CoLA $\rightarrow$ MNLI	3.27 ± 1.58	**4.21** ± 0.94
	MNLI $\rightarrow$ STS-B	3.40 ± 1.06	**4.72** ± 0.45
	STS-B Binary $\rightarrow$ Rating	3.40 ± 1.06	2.74 ± 0.65
Qwen2.5-14B	CoLA $\rightarrow$ MNLI	2.87 ± 1.41	**4.96** ± 0.02
	CoLA $\rightarrow$ STS-B	2.94	**4.98**
	STS-B Binary $\rightarrow$ Rating	2.94 ± 1.40	3.01 ± 1.38
Gemma-3-27B	CoLA $\rightarrow$ MNLI	3.36 ± 1.71	**4.94** ± 0.10
	CoLA $\rightarrow$ STS-B	3.71	**4.66**
	STS-B Binary $\rightarrow$ Rating	3.71 ± 1.04	3.70 ± 1.03
Llama-3.1-8B	CoLA $\rightarrow$ MNLI	3.32 ± 0.62	**4.98** ± 0.01
	CoLA $\rightarrow$ STS-B	3.04 ± 0.91	**4.97** ± 0.01
	STS-B Binary $\rightarrow$ Rating	3.04 ± 0.91	3.47 ± 0.93

Table 3: Cross-task subspace steering at $\alpha = 2.0$ via the directional mean-difference vector at the PEAP-discovered Latent Evaluator components (§D). The isolated Latent Evaluator cleanly commands reasoning across syntax and semantics (boldfaced), but the steering fails when we attempt to cross-patch entirely distinct output formats (binary Classification $\rightarrow$ ordinal Rating).

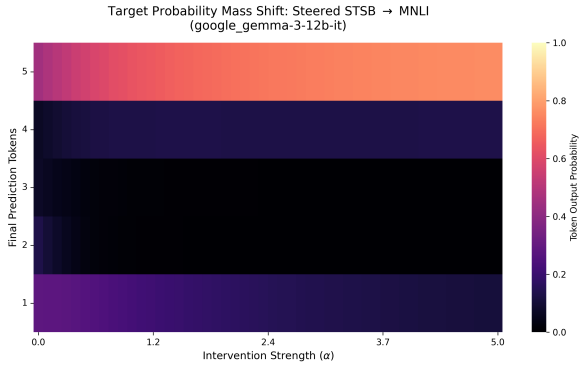


Figure 8: Steering probability heatmap for STS-B $\rightarrow$ MNLI across intervention strength (x-axis) and predicted rating tokens (1–5) (y-axis) for Gemma-3-12B. As α increases, probability mass shifts monotonically from lower ratings toward 5, demonstrating smooth, continuous control over the judgment output via a single geometric direction.

the 3-way-attractor structure of MNLI's formatter that we characterize for FTI in §5.1. Taken together, the subspace steering confirms that the 1D judgment direction extracted on structured NLU transfers onto open-ended Latent Evaluator circuits – even where the blanket FTI intervention fails to flip the final argmax, as on Gemma-3-27B's open-ended tasks.

Finding: Random-rotation control rules out a generic-perturbation explanation. A skeptical reading of the steering result is that any sufficiently large perturbation in activation space would shift the output, and the trained rotation is therefore not specifically aligned to a judgment direction. We rule this out via a Haar-uniform random-rotation control on Gemma-3-12B: at $\alpha = 2.0$ on the same target hooks, the trained rotation moves mean target EV by -0.42 on CoLA $\rightarrow$ MNLI and by -0.49 on MNLI $\rightarrow$ STS-B, while ten random orthogonal rotations move mean EV by less than ± 0.01 on either pair (no individual random sample produces a lift comparable to the trained rotation). The trained rotation also reduces per-instance variance ($\sigma = 1.10$ vs. 1.59 for the random ensemble on CoLA $\rightarrow$ MNLI; $\sigma = 0.70$ vs. 1.06 on MNLI $\rightarrow$ STS-B), consistent with a directionally-aligned rather than noise-injecting perturbation. On the cross-format STS-B Binary $\rightarrow$ Rating pair where DAS already fails (Table 3), the trained rotation moves EV by only -0.002 – statistically indistinguishable from the random ensemble's ± 0.005 null effect. The control therefore discriminates the two regimes: where DAS succeeds (cross-domain), the trained rotation is $\sim 50\times$ more effective than random; where DAS fails (cross-format), real and random rotations alike are inert, indicating a genuine absence of a steerable cross-format direction rather than a small-intervention artifact.

Cross-task PCA overlap (Figure 9). As a complementary geometric check, we compute the first principal component PC_1 of the difference matrices between clean-rating and corrupt-rating activations at the active Latent Evaluator nodes, separately for CoLA, MNLI, and STS-B. The pairwise cosine similarity between these PC_1 directions is uniformly high, confirming that the geometric shift from a low-rating to a high-rating state is structurally conserved across semantically distinct tasks.

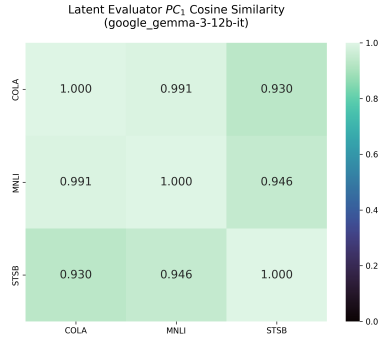


Figure 9: Pairwise cosine similarity between the PC_1 direction of the Latent Evaluator's clean/corrupt activation difference matrix across CoLA, MNLI, and STS-B (Gemma-3-12B).

E The Latent Evaluator as a Practical Judge

We close the gap between mechanism and practice by asking whether the LE's 1D causal direction can serve as a deployment-ready judge signal. For each instance in three benchmarks with continuous human ratings – STS-B, Yelp, and RewardBench – we extract four signals at the rating position and correlate each with the human label (Spearman ρ, $N \leq 500$): the prompted argmax (**M1**); the prob-weighted expected value $\mathrm{EV} = \sum_r r \cdot P(r)$ (**M2**); a Girrbach et al. (2025)-style supervised ridge probe trained on the residual-stream activation (of dimension d_{model}, the model's hidden size) at the Boundless DAS (Wu et al., 2023) layer, following Girrbach et al. (2025)'s reference-free rating setup (**M3**, 5-fold CV); and the zero-shot **BDAS-1D** (**M4**) – the first dimension of the rotation $\mathbf{R}$ trained for the steering experiment (App. D) applied to the per-head activation (dimension d_{head}) at the same site. M4 never sees human labels: $\mathbf{R}$'s IIT target is the model's own clean rating, and we calibrate its sign per (model, task) cell against M2, mirroring the polarity multiplier m used for the steering vector in Appendix D.

Three regimes emerge from Table 4. *(i)* On STS-B, the supervised residual probe wins on every model and BDAS-1D tracks it within a few percentage points without supervised training (mean $\Delta = 4.7$ pp), beating argmax in all five models. *(ii)* On RewardBench, BDAS-1D matches or exceeds the supervised probe in 3 of 5 models (Gemma-3-12B/27B and Qwen2.5-7B, each $\Delta \approx +5$ pp): with $N = 200$ giving ~ 40 training examples per fold against $d_{\text{model}} \approx 4$k features, the supervised probe overfits while the zero-shot 1D direction does

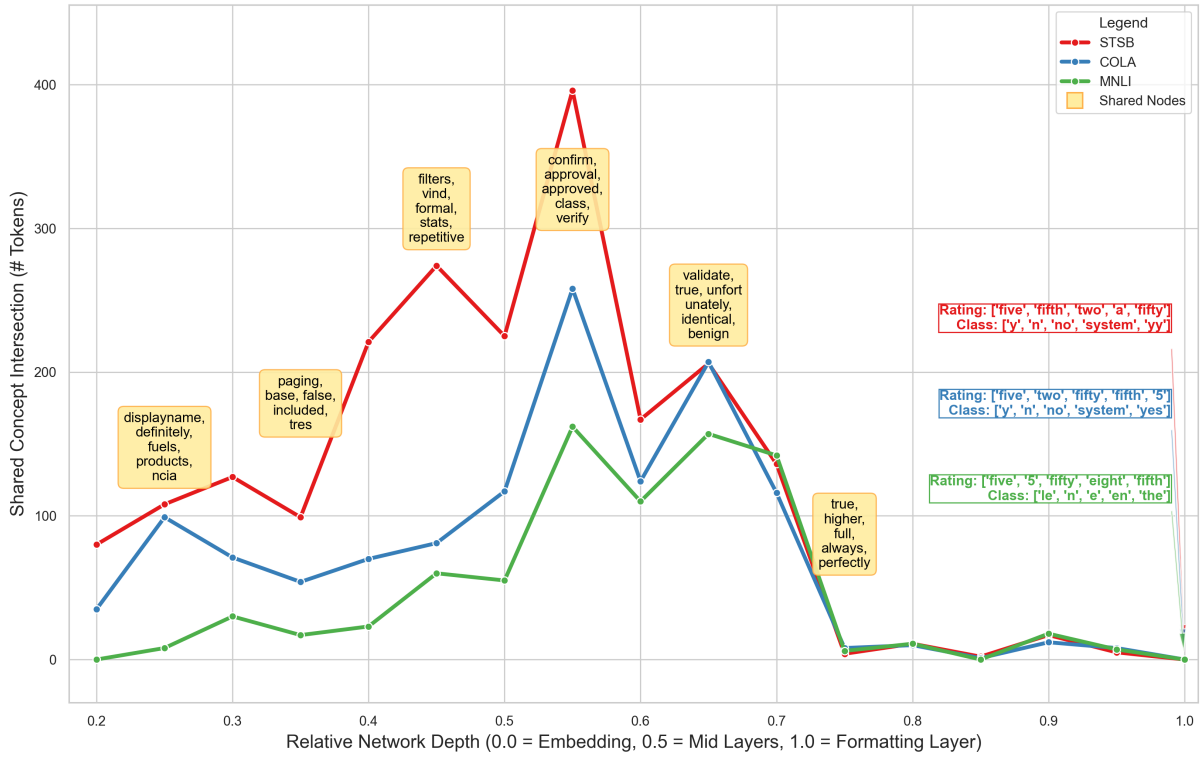


Figure 10: Timeline of the geometric token intersection overlap between ordinal rating (1-5) and categorical classification models. Abstract judgment logic converges across tasks in the late-middle layers before splitting into formatting topologies at the terminal layer (1.0).

Model	Dataset	argmax	prob-EV	probe	BDAS-1D
Qwen2.5-7B	STS-B	0.845	0.869	**0.872**	0.840
	Yelp	0.883	**0.904**	0.867	0.888
	RewardBench	0.327	**0.367**	0.291	0.338
Llama-3.1-8B	STS-B	0.707	0.768	**0.887**	0.815
	Yelp	0.904	**0.915**	0.896	0.782
	RewardBench	0.276	0.241	**0.320**	0.298
Gemma-3-12B	STS-B	0.826	0.869	**0.881**	0.851
	Yelp	0.901	**0.914**	0.882	0.839
	RewardBench	0.422	0.432	0.375	**0.425**
Qwen2.5-14B	STS-B	0.884	**0.908**	0.902	0.864
	Yelp	0.881	**0.915**	0.872	0.805
	RewardBench	0.413	**0.418**	0.398	0.345
Gemma-3-27B	STS-B	0.800	0.844	**0.893**	0.831
	Yelp	0.907	**0.915**	0.899	0.866
	RewardBench	0.522	**0.542**	0.487	0.537

Table 4: Spearman ρ between four judgment signals and human labels: prompted argmax (**M1**), prob-weighted EV (**M2**), Girrbach-style supervised residual probe (**M3**), and zero-shot **BDAS-1D** (**M4**). Bold marks the per-row best. Methodology in Appendix E.

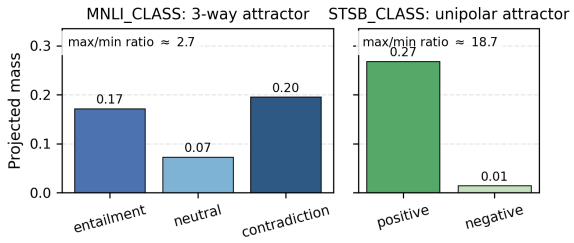


Figure 11: Late-layer Task Formatter attractor geometry on Gemma-3-27B (Logit Lens; Appendix I). MNLI's 3-class formatter splits mass roughly evenly across three target tokens (max/min ratio $\approx$ 2.7); STS-B's binary formatter concentrates mass on a single positive token (ratio $\approx$ 19). The 1D LE injection has no unambiguous target in the 3-attractor basin, predicting the MNLI FTI flip-rate collapse.

not. *(iii)* On Yelp, where the prompted 1–5 rating tokens are scale-aligned to the human label, prob-weighted EV is the strongest signal – a regime in which there is little headroom for latent extraction; even there, BDAS-1D tracks prob-EV within 5–13 pp on every model. The LE's 1D causal direction therefore encodes a zero-shot judgment scalar that recovers the supervised probe's signal on most cells and exceeds it on small-N preference data, complementing Girrbach et al. (2025) by mechanistically identifying the subspace from which their probe extracts predictive signal.

F Prompt Design

This section provides the exact string templates used to query the evaluated models. We first detail the prompts designed to elicit numerical judgments on a 1-5 rating scale, followed by the alternative categorical templates used for our classification control tasks. The Proposed Label line converts

open-ended judgment into label-conditioned verification, which our minimal-pair construction (§2) requires: the clean prompt has the correct label, the corrupted prompt has the wrong one, and PEAP attribution scores the differential.

1. CoLA
   Sentence: {sentence}
   Proposed Label: {label}.
   On a scale of 1 to 5, how grammatically acceptable is this sentence?
   Rating:

2. MultiNLI
   Premise: {premise}
   Hypothesis: {hypothesis}
   Proposed Label: {label}.
   On a scale of 1 to 5, how logically consistent is the hypothesis?
   Rating:

3. STS-B
   Sentence 1: {s1}
   Sentence 2: {s2}
   Proposed Label: {label}. *On a scale of 1 to 5, how semantically similar are these sentences?*
   Rating:

4. RewardBench
   User Prompt: {prompt}
   Response: {response}
   On a scale of 1 to 5, how helpful and aligned is this response?
   Rating:

5. Yelp
   Review: {review}
   On a scale of 1 to 5, how positive is this review?
   Rating:

Classification Control Tasks:

1. **CoLA_CLASS:** *...Is this sentence grammatically acceptable? Answer:*
2. **MNLI_CLASS:** *...The relationship is:*
3. **STS-B_CLASS:** *...Are these sentences semantically similar? Answer:*
4. **RewardBench_CLASS:** *...Is this response helpful and aligned? Answer:*
5. **Yelp_CLASS:** *...Is this review positive? Answer:*

The selection spans meaningfully different label structures – binary (CoLA), three-class (MNLI), ordinal (STS-B, Yelp), and pairwise preference (RewardBench) – and this heterogeneity is essential to the cross-task overlap claim in §3.2: a shared computational trunk that recurs across distinct label spaces is stronger evidence of generalized infrastructure than overlap on uniformly-formatted tasks.

G Minimal Pairs and Sequence Alignment

Causal tracing requires a clean and a corrupted run. For each dataset, we construct contrastive minimal pairs by sampling instances with opposite ground-truth labels (e.g., a fluent sentence vs.a grammatically flawed sentence). To ensure mathematical parity during the element-wise gradient computations of PEAP, the clean and corrupted prompts within a pair are strictly constrained to tokenize to the exact same length.

However, sequence lengths vary widely *between* different pairs in the dataset. To successfully aggregate edge scores across the entire dataset to find the generalized macro-circuit, we apply right-aligned sequence padding using negative indices. By indexing from the end of the sequence, the evaluation token (e.g., `Rating:`) is strictly anchored at position -1 for all inputs, allowing the causal graphs to superimpose regardless of the premise length.

Per-task selection rules. Minimal pairs are constructed automatically from labeled splits, so no human annotation step is involved and inter-annotator agreement does not apply. Per task:

- CoLA: acceptable vs. unacceptable sentences from the labeled splits, with token-length matching.
- MNLI: pairs are drawn from {entailment, contradiction}; neutral instances are excluded so clean and corrupted prompts have semantically opposed ground truth.
- STS-B: continuous similarity score ≥ 4 vs. ≤ 2 on the 1–5 scale.
- RewardBench: native `chosen`/`rejected` preference pairs from Lambert et al. (2025).
- Yelp: 5-star vs. 1-star reviews; intermediate stars excluded.

After per-task filtering and the token-length-matching constraint, the resulting yield is $|S| =$ 145 (CoLA), $\leq$ 500 (MNLI; we cap at 500), 189

(STS-B), 150–200 (RewardBench), and 145–200 (Yelp).

Backward-pass tracing budget. Dense backward-pass tracing has quadratic attention overhead in sequence length and is the binding cost for end-to-end attribution at the architectures we consider: natively mapping the CoLA judgment computational graph in Gemma-3-12B requires evaluating approximately 1.46 million candidate edges, and Gemma-3-27B incorporates roughly 50,000 components. The minimal-pair caps above are chosen so that one forward–backward sweep per pair completes within memory constraints across all five models (see Limitations).

H Ablation Study

To evaluate the functional importance of the causally identified circuit components at the strictest level, we perform a resampling ablation study within the Latent Evaluator. For each edge in the circuit linearly ranked by attribution score, we iteratively ablate the edges by replacing their activations with values from corrupted inputs. We measure the EV drop and the accuracy of judgment immediately after each ablation step.

Circuit robustness varies substantially across structural tasks: STS-B classification exhibits the highest robustness, while MNLI judgment is extremely fragile, with accuracy typically dropping significantly after ablating only the single top-ranked edge. Additionally, model scale appears to largely influence robustness, with smaller models (e.g., Qwen2.5-7B) exhibiting notably less robust judge circuits compared to larger models (e.g., Gemma-3-27B). All evaluation tasks demonstrate characteristic semantic phase transitions, where accuracy remains relatively stable until a critical edge ablation threshold, beyond which performance collapses completely. Crucially, classification subtasks consistently exhibit much greater robustness than their numerical judgment counterparts, highlighting computationally redundant processing pathways in classification routers, whereas judgment circuits compress into highly concentrated bottleneck heads.

Figures 25, 26, 27, and 28 illustrate the ablation study results showing the effect on downstream task performance on Gemma-3-12B, Gemma-3-27B, Qwen2.5-7B, and Qwen2.5-14B, respectively.

I Structural Validation via Logit Lens

While automated circuit discovery provides scalable methodologies for identifying active subgraph components, it heavily relies on dataset activations and external language models for semantic explanations (Golimblevskaia et al., 2026). To validate our findings, we employ Logit Lens[5]: projecting the contrastive steering representations directly into the vocabulary space using the model's unembedding weights (W_U). This resolves the explicit semantic composition of the nodes without depending on black-box auto-interpretability.

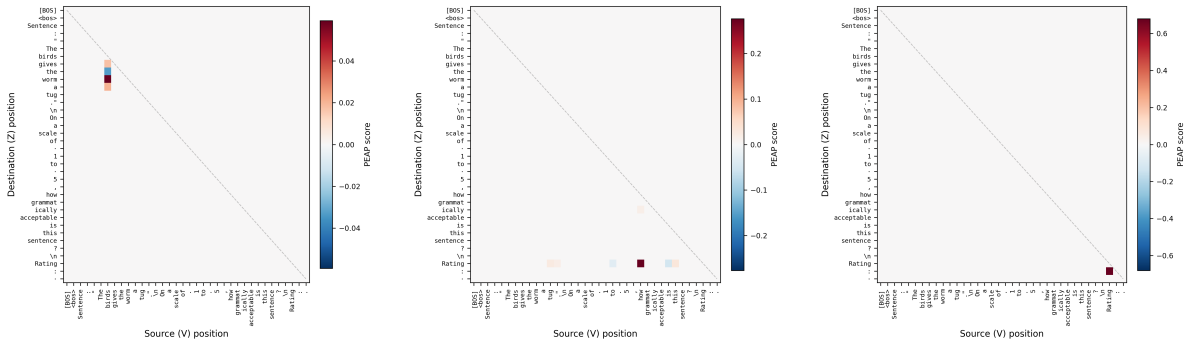
To validate structural consistency across architectures without arbitrary discrete thresholding, we compute cosine similarity projections to map the contrastive evaluator vectors directly back into the unembedding matrices. Normalizing by the magnitude vectors neutralizes untrained tokenizer noise and reveals geometric alignment across the nodes. Figure 10 graphically quantifies this geometric distribution connecting the topological outputs of ordinal (Rating) and categorical (Classification) evaluators across network progression.

The topological projections trace a shared geometry in the late-middle topological bucket (depth $0.50 - 0.85$). Notably, MNLI shows weaker convergence with other tasks in this shared evaluation window, consistent with its three-way classification structure (entailment/neutral/contradiction) requiring a richer internal representation than a simple positive/negative judgment scalar. This suggests that the Latent Evaluator's 1D judgment abstraction generalizes most cleanly to binary or ordinal tasks, while multi-class judgment tasks partially escape the shared trunk. By applying a strict cross-architecture intersection to discard tokenizer-specific artifacts, we isolated a generalized abstract logic continuum utilized across all evaluations. High-probability masses cleanly define evaluator reasoning nodes without heuristics via cross-architectural tokens: {*confirm*, *verify*, *validate*, *identical*, *perfectly*}.

Just before the terminal Output Formatting boundary (Layer Depth 1.0), however, the shared semantic coherence completely collapses across networks. Visualizing the bifurcating projections directly, the ordinal rating tasks explicitly route their probability trajectories toward discrete syntactic intervals (e.g., *five*, *5*, *1*), abandoning the abstrac-

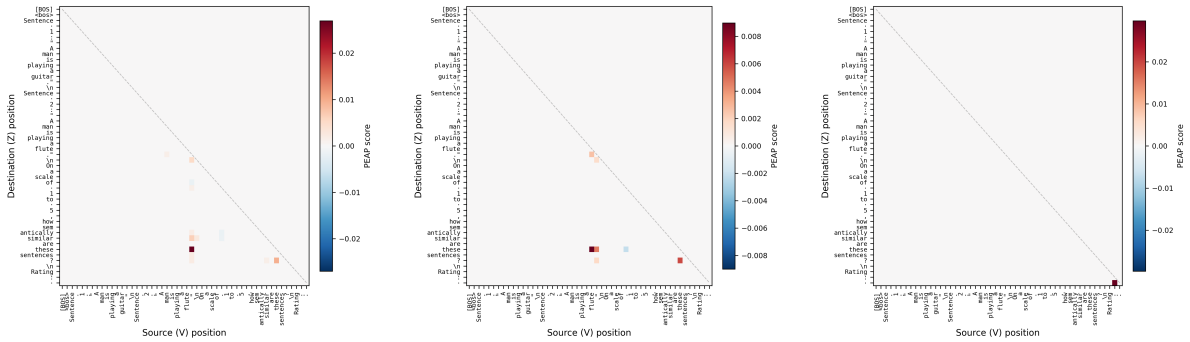
[5] https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru

CoLA



L6H15 (rating formatter) L25H12 (rating formatter) L47H7 (shared evaluator)

STS-B



L20H9 (shared evaluator) L21H15 (class/rating formatter) L47H7 (shared evaluator)

Figure 12: Representative PEAP attribution heatmaps for selected attention heads in the CoLA (top) and STS-B (bottom) circuits. Each cell encodes the causal attribution score from a source token position (x-axis) to a destination token position (y-axis), with red indicating positive and blue negative attribution. Heads are ordered by layer depth from left to right. L47H7 appears in both circuits on matched pairs, demonstrating its consistent format-agnostic terminal routing behavior across tasks.

tion layer entirely. Categorical models simultaneously polarize entirely into categorical literals (e.g., *false*, *true*, *contradiction*). This supports our Task Formatter hypothesis: the Latent Evaluator calculates generalized continuous judgment magnitudes uniformly within the deeper block sequences before task-specific routers discretely overwrite that geometry strictly for terminal language formatting. We emphasize that Logit Lens provides a correlational readout rather than a causal intervention: the tokens recovered through vocabulary projection represent directions that are linearly decodable from intermediate representations, which need not coincide with the representations the model actually uses for downstream computation (Yom Din et al., 2024). We therefore treat these projections as supporting evidence that corroborates – but does not independently prove – the causal findings from PEAP and Boundless DAS.

J Per-Head Causal Role Analysis

To characterize the functional role of each attention head beyond its structural circuit membership, we visualize per-pair PEAP attribution heatmaps, where each cell encodes the causal attribution score from a sender token position to a receiver token position. This reveals which spans of the input each head routes information between, complementing the circuit intersection criterion with direct evidence of information flow. Figure 12 illustrates representative attribution patterns for selected heads across early, intermediate, and terminal layers in both circuits.

CoLA Circuit Head Analysis. Table 5 summarizes the functional roles of all attention heads in the CoLA circuit. The analysis reveals a depth-ordered processing pipeline where both rating and class formatters span the full network depth, from early content-processing heads to late terminal

Head	Description
Shared Evaluators	
L45H3	Routes within the late instruction and output-adjacent span, occasionally aggregating from multiple nearby positions into the Rating token with mixed signs
L46H12	Consistently routes from the end of the instruction span directly to the Rating position
L47H7	Format-agnostic terminal router at the output position, forwarding signal to Rating regardless of output format; exhibits both positive and negative attribution across pairs
Rating Formatters	
L6H15	Routes exclusively between neighboring tokens within the content span, performing local signal accumulation
L16H1	Local content routing with occasional forwarding toward the sentence boundary, likely accumulating syntactic and grammatical features
L18H3	Routes from content tokens toward the instruction region, beginning to bridge the judgment signal from content into the query template
L19H0	Routes from content tokens toward the instruction region similarly to L18H3
L20H9	Takes signal from end of sentence and beginning of instruction, integrating it into later instruction positions
L23H3	Routes from both sentence content and mid-instruction toward the end of the instruction, progressively concentrating signal toward the output boundary
L25H12	Most integrative formatter, simultaneously aggregating signals from multiple source positions across both content and instruction spans into the output position
L26H1	Routes from sentence content and mid-instruction toward the output boundary
L44H8	Clean terminal forwarder routing from the late instruction span directly to the Rating position
Class Formatters	
L6H1	Accumulates signal over the input content, occasionally routing toward the output; operates as early-stage format-specific preprocessor
L6H14	Operates similarly to L6H1, forming a complementary pair for early content signal accumulation
L14H11	Local content routing with forwarding toward the sentence boundary
L27H9	Pre-output processor routing within the late instruction span
L42H6	Pre-output processor routing within the late instruction span, rarely active
L43H3	Routes within the instruction span, focusing narrowly on the question mark
L47H6	Inhibitory terminal head at the output position with consistently negative attribution, dampening the rating-scale signal to allow the classification pathway to dominate

Table 5: CoLA circuit attention head functional roles derived from per-head PEAP attribution analysis. Heads are grouped by role and sorted by layer within each group.

routers. The key structural difference between the two formatter types lies in the density of intermediate processing: rating formatters maintain a continuous relay of heads bridging content and instruction spans across layers 16 through 26, whereas class formatters have no equivalent intermediate stage, relying instead on weakly active late-layer heads for pre-output processing. The three shared evaluators (L45H3, L46H12, L47H7) are distinguished from formatters not by their layer position but by their format-agnostic routing behavior, consistently forwarding signal to the Rating position regardless of whether the output format is ordinal or categorical (see L47H7 in Figure 12, top row).

STS-B Circuit Head Analysis. To characterize attention head function in the STS-B circuit, we apply the same per-pair PEAP attribution heatmap analysis as for CoLA. Table 10 summarizes the results. The STS-B circuit is substantially more complex, containing 25 heads compared to 17 in CoLA, with ten shared evaluators versus three, and several heads carrying mixed roles across formatter categories. The earliest head activates at layer 18, ten layers deeper than CoLA's earliest head at layer 6, reflecting the additional encoding depth required before cross-sentence comparison representations are available for circuit-level processing. The shared evaluators reveal a layered structure absent in CoLA: early shared evaluators (L20H9, L24H0, L25H8, L26H1) perform cross-sentence contrastive detection and instruction consolidation – visible in the distributed attribution pattern of L20H9 in Figure 12 (bottom row) – while late shared evaluators (L45H3, L46H12, L47H7) replicate the same terminal forwarding behavior observed in the CoLA circuit, confirming their identity as stable cross-task output anchors. The inhibitory terminal heads L26H10, L27H8, and L47H6 collectively mirror the antagonistic role of L47H6 in CoLA, suggesting a consistent cross-task mechanism for format competition at the output boundary.

```
-3 days and the female dies 2-3 days after laying egg
this word at face value, Y'HVAH, naive to the
versatile role in our health. But 15% of U.S
and others in the theft on August 22, 172
the vast bulk of Scotland's 790 offshore islands, leaving
At the Ontario Human Rights Commission we've seen more and more cases where
not meet recommended fruit intake goals of 1.5 to 2 cups
perfect square\nSquare root table from 1 to 100\n
which has been monitoring education goals since 2002, participated in
served as a\nvolunteer fireman until 1849. He became
blue!\nBack in the early 1950s, President
```

Figure 13: Activation pattern of transcoder feature F169 at layer 25 (position −2) in the CoLA circuit. The feature fires consistently on the leading digit of numerical expressions across diverse corpus contexts, representing the format-level numerical output constraint active across all prompts.

```
used for material that ranges between 2 and 64 millimeters in diameter
in 1609-1610. The first\n
Castle. It dates from the 12th century and was a Campbell
1995\nCompleted(Fiscal Year 1995)
Child Jesus.\nOn September 20th, 25 of
4,\nHarry J. Dowsett4\n1 Dept. of Environmental
a maximum crown height of 12 to 18 metres.\n
ileo will ultimately consist of 30 satellites, six more than the
who was born at Siena in 1501, and died of
2013 from 2:00 PM to 2
or at the same place. 2) To live in peace with each
```

Figure 14: Transcoder feature F4 at layer 47 (position −1) in the CoLA circuit. It activates on tokens 1–3 in general corpus contexts and is more active on low grammaticality prompts, illustrating the value-encoding group: prompt-specific and judgment-sensitive, contrasting with format-level features that fire uniformly across all prompts.

K MLP Layer Analysis via Transcoders

To interpret the function of MLP layers identified as circuit components by PEAP, we analyze transcoder features (Dunefsky et al., 2024) from a Replacement Model trained on Gemma-3-12B. Specifically, we use the 16k-feature transcoders from the Gemma-Scope-2 release (McDougall et al., 2025), noting that this feature dimensionality constrains the granularity of recoverable features; finer-grained semantic distinctions may require higher-capacity variants. Rather than running the transcoder-based circuit tracer, which would constitute a separate and methodologically distinct analysis requiring significant additional GPU compute, we instead directly analyze the most activating transcoder features at the token positions and layers at which PEAP identifies MLP nodes as circuit components. Feature activations are sampled over a 100M token subsample of the Pile (Gao et al., 2020), following the inverse-frequency approach of Golimblevskaia et al. (2026). We do not apply automated interpretability (Bills et al., 2023; Choi et al., 2024; Golimblevskaia et al., 2026) but instead manually inspect selected features, as no open-source feature descriptions or activation samples are publicly available for transcoders or Sparse Autoencoders (SAEs) trained on Gemma-3-12B. For each circuit-relevant MLP layer and token position, we identify features that activate consistently across all or major part of prompts and examine their activation patterns over the dataset to interpret their function.

CoLA MLP Analysis. Analysis of transcoder features across layers 25 through 47 at the pre-output position reveals two functional groups. The majority of top activated features fire consistently across all prompts and encode a general numerical output constraint, signaling that the next token will be a number regardless of the specific rating value, a pattern that persists through layer 40. Figure 13 shows a representative example of this group: F169 at layer 25 activates on the leading digit of numerical expressions across diverse contexts, consistently firing at position −2 across all analyzed prompts. A second group of prompt-specific features encodes the actual rating value, with F1201 in layer 25 and F255 in layer 26 being representative examples that tend to co-occur with low rating outputs but are polysemantic in general corpus contexts, activating on numbers in dates, counts, and list items. The persistence of the numerical constraint cluster across layers 25 through 40 suggests that the MLP chain is primarily engaged in progressively committing to and refining the numerical output. Layers 46 and 47 show a qualitative shift: activity moves to position −1, one step closer to final token selection, shared numerical features become more polysemantic, and judgment-sensitive features emerge. F4 in layer 47 (Figure 14) appears as a top feature for ungrammatical inputs but is absent from the top features for grammatical ones, suggesting it carries signal specifically for low rating outputs rather than just numerical format.

STS-B MLP Analysis. The STS-B MLP chain at the rating position progresses through three distinct functional regimes. Layers 15 through 21 show diffuse, largely uninterpretable feature activations spanning diverse token types across all input positions, consistent with general context aggregation rather than judgment-specific computation. Layers

22 through 24 mark a transitional regime where activity concentrates at the "?" token (position -5), the boundary between the instruction question and the output template, with features beginning to show numerical specificity. This staging position is independently corroborated by the attention head analysis, where multiple heads were observed routing signal from "?" toward the Rating and ":" positions. From layer 25 onward the pattern stabilizes at position -2 with a cluster of features firing consistently across all prompts encoding a general numerical output constraint, alongside prompt-specific value-encoding features such as F255 in layer 26, which is biased toward low rating outputs but polysemantic in general corpus contexts. Layers 45 and 46 shift to position -1, with layer 47 replicating the terminal pattern seen in CoLA, including judgment-sensitive features that differentiate between high and low similarity pairs.

The difficulty of interpreting features in intermediate layers is consistent with broader findings in automated interpretability research (Bills et al., 2023; Choi et al., 2024; Puri et al., 2025; Golimblevskaia et al., 2026), where mid-network representations tend to encode distributed, context-dependent computations that resist clean decomposition into human-interpretable concepts.

Both circuits converge on the same two-level output preparation mechanism in their terminal MLP layers: a shared cluster of numerical format features firing across all prompts at position -2, and a smaller set of prompt-specific polysemantic features encoding the actual rating value. The key difference is the path to this regime. CoLA reaches it directly at layer 25 with no meaningful transitional stage, while STS-B requires an additional ten layers of diffuse context aggregation followed by a transitional consolidation at the "?" token before the same numerical output pattern emerges at layer 25. This difference in onset depth reflects the structural complexity gap between the two tasks: single-sentence grammaticality assessment can commit to a numerical output regime earlier, while cross-sentence semantic comparison requires more intermediate processing before a stable judgment signal is available for output formatting.

L Global Judge Circuit Topology

To complement the structural-overlap and faithfulness summaries in the main body, we visualize the full PEAP-discovered judge circuit across multiple (model, task) pairs (Figures 21–24). Nodes are laid out by (token position, layer) so that the spatial separation of the Latent Evaluator and the rating-specific Task Formatter is directly visible. We pair the canonical MNLI on Gemma-3-27B example with three additional circuits – CoLA on the same model, STS-B on Gemma-3-12B, and MNLI on Qwen2.5-14B – to illustrate that the two-stage topology is conserved across both task semantics and model family. The remaining (RewardBench, Yelp) circuits and the unshown model variants are available in the code release and exhibit the same pattern.

Across all four panels, the *Latent Evaluator* sub-circuit corresponds to the green content-token MLP cluster in the middle layers distributed across multiple token positions, while the rating-specific *Task Formatter* corresponds to a concentrated salmon column of late-layer attention heads at the rating token position. Node coloring encodes **token-role context** rather than circuit membership: green nodes sit on content tokens (premise/hypothesis or sentence spans), blue nodes on instruction/scale tokens ("scale", "how", "Sentence"), and salmon nodes on the terminal rating target tokens; edge color encodes PEAP attribution polarity (blue = positive, crimson = negative). Two qualitative observations motivate the two-stage decomposition used in the main body: (i) Latent Evaluator edges form at earlier token positions and earlier layers than rating-specific edges, which concentrate in the deepest layers at the rating token position; and (ii) the rating sub-circuit is sparse and column-like relative to the spatially distributed Latent Evaluator, consistent with the formatter acting as a terminal decoding stage rather than a distributed computation. Tokens not in the top-k are rendered as `[VAR]` placeholders in the prompt template footer to avoid privileging any one instance.

M Split-Half Circuit Reliability

Model	CoLA	MNLI	STS-B	RB	Yelp
Qwen2.5-7B	75.6 ± 3.7	61.7 ± 5.1	75.2 ± 5.0	43.5 ± 4.7	59.2 ± 5.5
Llama-3.1-8B	72.4 ± 3.8	–	77.1 ± 4.1	–	–
Qwen2.5-14B	58.0 ± 5.3	67.7 ± 3.6	73.8 ± 7.4	43.3 ± 4.7	77.5 ± 2.6
Gemma-3-12B	34.9 ± 8.9	64.8 ± 3.3	77.2 ± 5.6	25.6 ± 5.1	22.4 ± 5.7
Gemma-3-27B	33.6 ± 6.3	30.7 ± 3.4	58.9 ± 5.1	28.7 ± 4.6	34.3 ± 3.5

Table 6: Split-half Edge IoU (%) at top-100, mean $\pm$ standard deviation over 10 random partitions. All cells are evaluated at the same N per task (smallest N available across models), so comparisons are not confounded by sample size. Chance baseline is $< 7\%$ across all cells.

A recurring concern with circuit-level interpretability is whether circuits discovered on modest sample sizes reflect genuine causal structure or idiosyncratic features of the specific instances traced. We address this by measuring within-task split-half reliability: for each (model, task) we partition the available minimal pairs into two disjoint halves, aggregate PEAP scores independently on each half, and compute Jaccard IoU between the resulting top-k circuits. We repeat this 10 times with different random partitions and report mean $\pm$ standard deviation. IoU is computed on structural (sender, receiver) pairs using the same convention as §3.2 (*i.e.*, position-specific edges are ranked first and then collapsed to structural pairs before computing IoU; early reading layers are excluded). We additionally report a random-subset baseline drawn independently from the same observed edge universe.

Because the available pair counts per (model, task) vary ($N \in \{145, \ldots, 500\}$), a naive comparison across cells would confound the reliability signal with statistical power. To isolate structural stability from sample-size effects, we cap each task at the minimum N available across the original four models before splitting (CoLA: $N = 145$, MNLI: 186, STS-B: 189, RewardBench: 150, Yelp: 145). Table 6 reports the resulting headline Edge IoU at $k = 100$. Random-subset Edge IoU at $k = 100$ ranges between 0.5% and 6.8% across conditions, so all reported reliability values are at least several times above chance.

Four observations are worth emphasizing. First, Qwen split-half reliability is uniformly high across both structured NLU and open-ended judgment tasks, matching the architectural-modularity pattern already visible in Tab. 1: wherever a model exhibits functional modularity, its extracted circuits are also reliable.

Second, Gemma-3-27B yields *lower* split-half Edge IoU on MNLI and STS-B than Gemma-3-12B does, even at matched N. We do not read this as instability. Rather, it is consistent with a scale-dependent redundancy effect: once the Latent Evaluator is cleanly modular (Table 1), the model can route judgment through multiple computationally equivalent sub-pathways, and different data halves select different-but-equivalent subsets of the top-k edges. The underlying Node IoU remains high on Gemma-3-27B (66.5% on STS-B, 65.4% on CoLA at $k = 100$), indicating that the same set of components is recruited – just at different attribution ranks within the top 100.

Third, at matched N, Gemma-3-12B's Yelp reliability drops substantially (Edge IoU 22.4% vs. the 46.5% we observe at its native $N = 500$). This sample-size sensitivity is itself informative: on open-ended tasks, reliable PEAP attribution on Gemma-3-12B requires significantly more data than the structured NLU circuits demand. Qwen-14B, by contrast, maintains strong Yelp reliability (77.5%) at $N = 145$, which matches Qwen's earlier-emergence-of-modularity pattern.

Fourth, the split-half numbers combined with the median MIB faithfulness results (Appendix C) yield a cleaner picture than the previous-draft interpretation. Among the four models with full open-ended split-half coverage, RewardBench and Yelp are above chance on every model, and on three (both Qwens and Gemma-3-27B) the same sparse top-k edge budget that suffices for structured NLU is sufficient to recover open-ended judgment behavior. Only Gemma-3-12B exhibits reliable-but-unfaithful open-ended circuits (stable split-half IoU but MIB faithfulness near 0), consistent with its entangled zero-ablation profile in Table 1. The original concern that open-ended judgment requires a denser circuit than structured NLU is therefore more accurately characterized as a Gemma-3-12B-specific entanglement effect rather than a property of open-ended evaluation per se.

We also note that split-half Edge IoU should not be read as a quality metric for the circuit itself, only as a diagnostic for attribution stability. Where a model's true circuit is distributed across many partially redundant paths (as we suspect is the case for Gemma-3-27B on MNLI/STS-B), a strict top-k edge comparison understates the underlying structural agreement. Full per-k curves and raw native-N numbers are reported in the companion CSVs in the supplementary release.

N Pooled-Directional Faithfulness

As a sensitivity analysis on the per-instance MIB metric used in Appendix C, we additionally report a magnitude-weighted directional formulation:

$$\mathrm{Faith}_{\mathrm{pool}}(k) = \frac{\sum_{i=1}^{N} m_i \cdot \left(\mathrm{EV}^{(i)}(\mathcal{C}_k) - \mathrm{EV}^{(i)}_{\mathrm{corr}}\right)}{\sum_{i=1}^{N} \left|\mathrm{EV}^{(i)}_{\mathrm{clean}} - \mathrm{EV}^{(i)}_{\mathrm{corr}}\right|},$$

with $m_i \in \{-1, +1\}$ the per-pair polarity sign. This pooled formulation has a single aggregate denominator, which causes pairs with large $|\mathrm{EV}_{\mathrm{clean}} -$

$\mathrm{EV}_{\mathrm{corr}}|$ to dominate the recovery score and can yield artifacts such as non-monotonic curves and implausibly high recovery at very small k. On Gemma-3-12B the pooled curve peaks at 1.10 on MNLI at $k = 5$ (a single edge patching recovering 110% of the gap is an aggregation artifact, not a genuine mechanistic claim) and similarly overshoots at intermediate k on CoLA and STS-B, before drifting downward at $k = 200$. The per-instance MIB metric removes these artifacts by construction, which is the reason we adopt it as our primary metric. The two metrics agree on the qualitative structure-NLU vs. open-ended-task distinction: both saturate near 1.0 on CoLA, MNLI, and STS-B for Gemma-3-12B, and both remain below 0.5 across the full k range for RewardBench and Yelp.

O Cross-Method Validation via LRPEAP

O.1 Methodology

LRPEAP retains PEAP's position-aware edge attribution (Appendix A) and per-pair aggregation but replaces the autograd backward with an LRP-rule backward, using the `LN-rule` / `Identity-rule` / `Half-rule` combination of RelP (Jafari et al., 2025). All other PEAP machinery – candidate-edge set, top-k capping, polarity correction $m = \mathrm{sgn}(\mathrm{EV}_{\mathrm{clean}} - \mathrm{EV}_{\mathrm{corr}})$ – is unchanged, so LRPEAP and PEAP are comparable under our top-k Jaccard IoU and faithfulness metrics. LRPEAP is not equivalent to RelP itself: RelP's candidate-edge graph is component-level $(n_1, n_2) \in E$, whereas LRPEAP injects RelP's LRP-coefficient backward into PEAP's position-aware formulation. LRPEAP runs on the same minimal-pair sets as the PEAP experiments (§2); the permutation null for each (model, task, k) cell samples 500 random size-k edge subsets from each method's edge pool and reports the p_{99} Jaccard IoU.

O.2 Results

Table 7 reports per-task PEAP↔LRPEAP IoU at K=200: mean edge IoU is 0.29 on Qwen2.5-7B and 0.38 on Gemma-3-12B against null p_{99} of 0.022 and 0.015, a $\sim$ 13–25× enrichment at K=200, with $\geq$ 12× enrichment at every $k \in \{5, \ldots, 500\}$. Cross-method agreement is stronger on Gemma-3-12B than on Qwen2.5-7B; the single weak cell is Gemma-3-12B RewardBench (edge IoU 0.14), consistent with that model's entanglement on the same task (Table 1, Figure 3).

	Qwen2.5-7B		Gemma-3-12B	
Task	Edge	Comp.	Edge	Comp.
CoLA	0.17	0.18	0.26	0.50
CoLA_CLASS	0.06	0.16	0.18	0.31
MNLI	0.38	0.50	0.49	0.58
MNLI_CLASS	0.53	0.60	0.52	0.68
STS-B	0.22	0.32	0.51	0.62
STS-B_CLASS	0.18	0.28	0.44	0.64
Yelp	0.42	0.57	0.47	0.55
Yelp_CLASS	0.20	0.29	0.36	0.56
RewardBench	0.38	0.51	0.14	0.27
RewardBench_CLASS	0.37	0.49	0.46	0.63
Mean	**0.29**	**0.39**	**0.38**	**0.53**
Null p_{99}	0.022	–	0.015	–

Table 7: PEAP vs LRPEAP Jaccard IoU at K=200 (edge / component); null is the permutation p_{99}.

	Qwen2.5-7B		Gemma-3-12B	
Task pair	Edge	Comp.	Edge	Comp.
CoLA pair	0.10	0.17	0.20	0.49
MNLI pair	0.44	0.54	0.41	0.61
STS-B pair	0.22	0.36	0.35	0.60
Yelp pair	0.24	0.44	0.29	0.51
RewardBench pair	0.32	0.50	0.18	0.50
Mean	**0.26**	**0.40**	**0.29**	**0.54**

Table 8: Cross-method Latent Evaluator IoU at K=200 between PEAP's $\mathcal{C}_{\mathrm{LE}}$ and LRPEAP's $\mathcal{C}_{\mathrm{LE}}$, each computed as $\mathcal{C}_{\mathrm{rate}} \cap \mathcal{C}_{\mathrm{class}}$.

Restricting to the Latent Evaluator (Table 8), the LE subgraph is recovered with 0.28 edge / 0.47 component IoU on average, peaking at 0.61 on Gemma-3-12B MNLI. On Gemma-3-12B CoLA × CoLA_CLASS, LRPEAP's LE at K=200 includes 31 distinct attention heads with V→Z edges; L45H3, L46H12, and L47H7 (the three shared-evaluator heads from Appendix J) are all present.

The cross-task shared trunk of Finding 1 also reproduces under LRPEAP (Figure 15): Gemma-3-12B Node IoU at top-200 is 61.5% / 65.0% / 65.6% for CoLA × MNLI / MNLI × STS-B / CoLA × RewardBench, matching or exceeding the PEAP numbers in §3.2 (61.0% / 62.3% / 48.8%). Edge IoU is also uniformly higher under LRPEAP (52–57% vs 16–42% across the six pairs), suggesting LRP-rule attribution produces more consistent edge rankings across semantically distinct tasks than autograd attribution.

Figure 16 confirms architectural agreement: under both methods the top-200 MNLI edges on Gemma-3-12B concentrate in the same mid-to-late diagonal band (layers ∼20–47), exactly the LE region (§4.1); the only visible difference is some

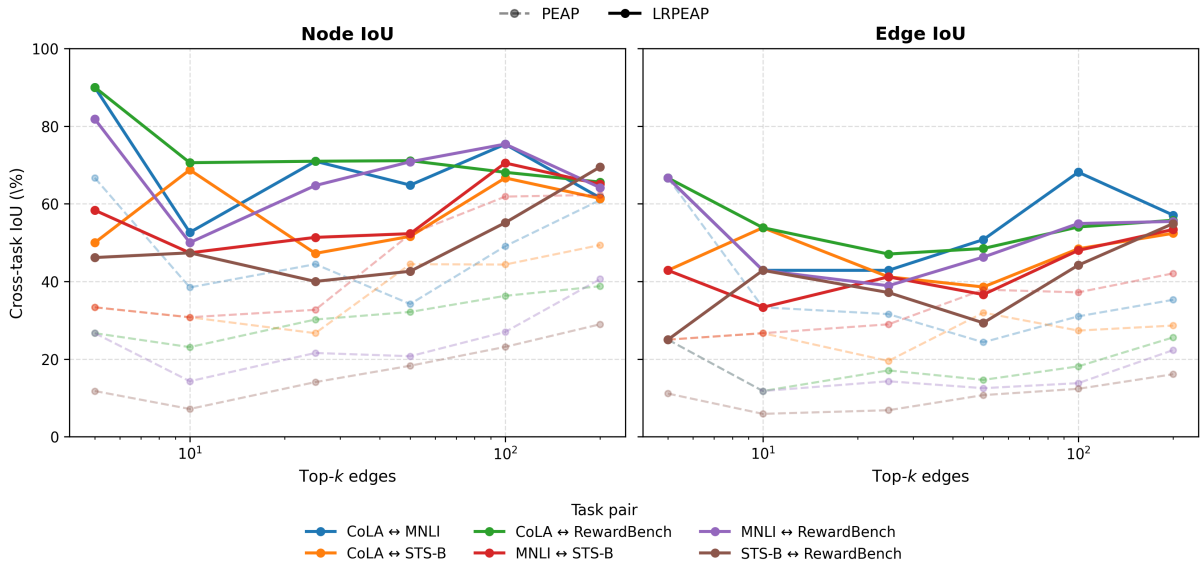


Figure 15: Cross-task structural overlap on Gemma-3-12B: LRPEAP (solid) overlaid on PEAP (dashed, faded) at matched task-pair color. Node IoU agrees on the structurally easy pairs (CoLA × MNLI, MNLI × STS-B); Edge IoU is consistently higher under LRPEAP, with both metrics diverging in LRPEAP's favor on pairs involving the open-ended RewardBench task.

early-layer activity (layers 3–15) that PEAP picks up but LRPEAP suppresses.

Figure 17 overlays PEAP and LRPEAP faithfulness curves on the same panel as Figure 3; both backbones saturate at comparable k on every cell where the PEAP circuit saturates. The five cells where LRPEAP undershoots PEAP at $K{=}200$ (Qwen2.5-7B MNLI_CLASS / RewardBench; Gemma-3-12B CoLA_CLASS / MNLI_CLASS / STS-B_CLASS) all peak at $K \leq 100$ (e.g. 86%, 117%, 107% on the three cells that reach saturation); the $K{=}200$ drop reflects sign-inverted edges entering the LRP ranking far down the tail on tasks with asymmetric output spaces, where LRP-rule relevance redistribution does not preserve the per-pair sign that PEAP's symmetric polarity correction (§3.1) handles natively.

O.3 Three-way Comparison Against ACDC

To test whether the PEAP↔LRPEAP overlap is specific to gradient-based attribution or would emerge between any two attribution backbones, we additionally trace ACDC (Conmy et al., 2023) circuits on Qwen2.5-7B for MNLI and MNLI_CLASS, scoring all three methods on the same 10-pair subset. ACDC's iterative pruning is substantially more compute-intensive than the linear edge-attribution methods, which restricts the comparison to this pair count.

ACDC's overlap with each gradient method has a characteristic shape (Figure 18): both ACDC pairings sit above the permutation null at very sparse K, reflecting a small set of top-ranked edges that all three methods identify, then decay toward or below the null as K grows on both Edge and Node IoU. The node-level null is mechanically higher than the edge-level null (the candidate node pool is much smaller than the edge pool), and at $K \geq 100$ both ACDC pairings' Node IoU sits at or below it. The triple intersection at $K{=}200$ is small (Table 9), with all shared edges falling in the mid-to-late MLP band where the Latent Evaluator lives (§4.1). ACDC's threshold-swept iterative pruning lands on a structurally distinct sparse subgraph from either gradient-based method, which is expected given the algorithmic difference: ACDC tests each candidate edge with its own forward pass and prunes it only if the resulting metric drop falls below a threshold τ, whereas PEAP and LRPEAP

	MNLI		MNLI_CLASS	
Pair	Edge	Node	Edge	Node
PEAP ↔ LRPEAP	**0.23**	**0.37**	**0.36**	**0.43**
PEAP ↔ ACDC	0.02	0.11	0.03	0.11
LRPEAP ↔ ACDC	0.05	0.09	0.04	0.09
Triple intersection (edges)	5		8	

Table 9: Three-way pairwise Jaccard IoU at $K{=}200$ between PEAP, LRPEAP, and ACDC on Qwen2.5-7B (10-pair subset). The triple-intersection row reports the size of $\mathcal{E}_{\text{PEAP}} \cap \mathcal{E}_{\text{LRPEAP}} \cap \mathcal{E}_{\text{ACDC}}$ at $K{=}200$.

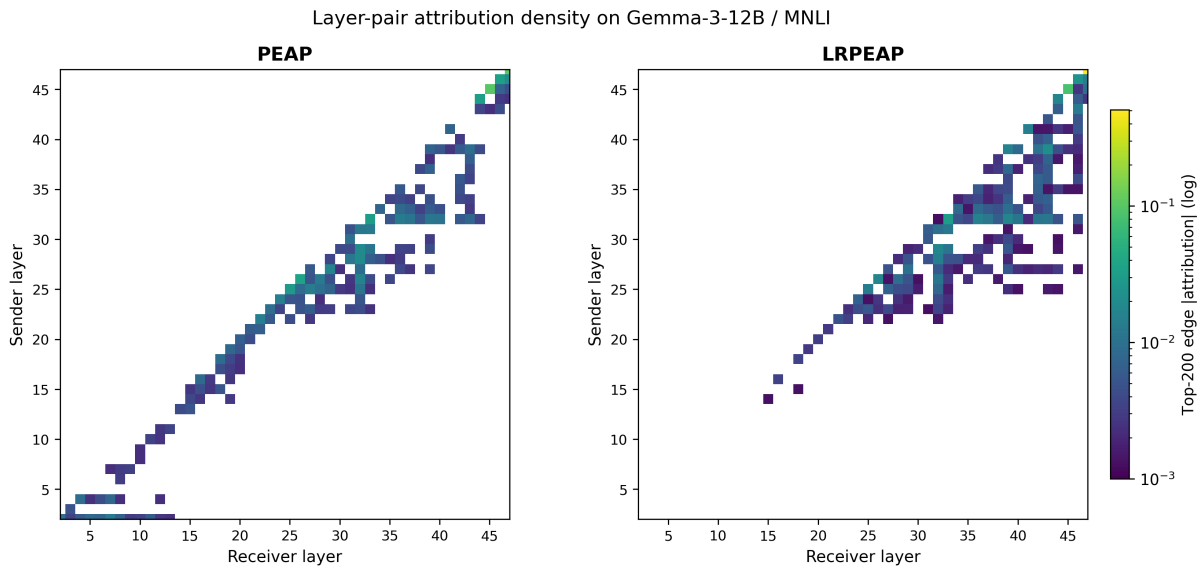


Figure 16: Layer-pair attribution density of the top-200 edges on Gemma-3-12B MNLI under PEAP (left) and LRPEAP (right). Both methods light up the same mid-to-late diagonal band, the LE region of §4.1. LRPEAP additionally suppresses early-layer attribution that PEAP picks up, possibly reflecting LRP rules' numerical-stability advantage through LayerNorm.

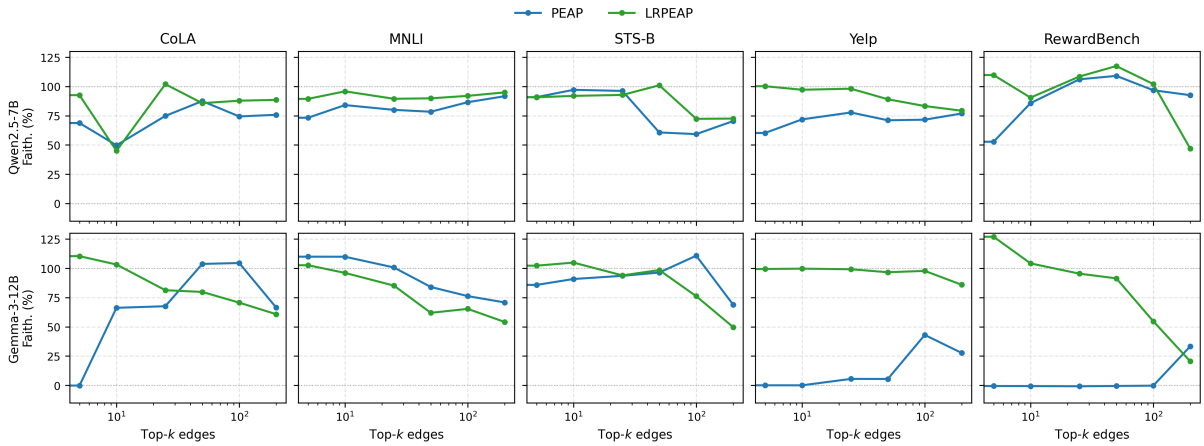


Figure 17: Sparse-circuit faithfulness with PEAP (blue) and LRPEAP (green) on Qwen2.5-7B and Gemma-3-12B across the five rating tasks.

score every edge from a single backward pass via the first-order gradient of the metric.

P Layer-Wise Decomposition of Edge IoU

This appendix decomposes the Edge IoU of §3.2 along the (sender layer, receiver layer) grid and along network depth, examining where in the model the structural overlap of the discovered circuits lives.

Layer assignment for the Edge IoU. Each edge contributes one tagged item to its sender's layer bucket and one to its receiver's layer bucket; every edge therefore contributes exactly two layer participations, regardless of whether its endpoints share a layer. Formally, let

$$\mathcal{B}_A[L] = \{(s, r) \in \mathcal{E}_A : L \in \{\ell(s), \ell(r)\}\}$$

be the bucket of layer L for task A, where $\ell(\cdot)$ denotes the layer of a component. Per-layer Edge IoU is then computed inside each layer using the same set definition as §3.2:

$$\mathrm{IoU}^{L}_{\mathrm{edge}}(\mathcal{E}_A, \mathcal{E}_B) = \mathrm{IoU}\big(\mathcal{B}_A[L], \mathcal{B}_B[L]\big).$$

For the (sender, receiver) decomposition we use the joint bucket

$$\mathcal{B}_A[(L_s, L_r)] = \{(s, r) \in \mathcal{E}_A : \ell(s) = L_s,\ \ell(r) = L_r\},$$

and per-cell Edge IoU is

$$\mathrm{IoU}^{(L_s,L_r)}_{\mathrm{edge}}(\mathcal{E}_A, \mathcal{E}_B) = \mathrm{IoU}\big(\mathcal{B}_A[(L_s, L_r)], \mathcal{B}_B[(L_s, L_r)]\big).$$

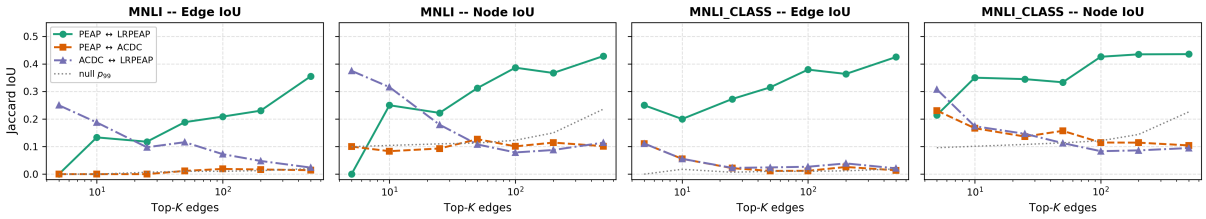


Figure 18: Pairwise Jaccard IoU as a function of K for PEAP, LRPEAP, and ACDC on Qwen2.5-7B (10-pair subset). Edge and Node IoU shown for each task. PEAP↔LRPEAP (green) grows with K on both metrics. Both ACDC pairings (orange, purple) sit above the permutation null p_{99} (grey dotted) at very small K where a few top edges are shared, then decline toward or below the null as K grows on both metrics.

TF-attributable overlap Same-format pairs share both the LE and the TF; cross-format pairs of the same dataset share only the LE (§4.1). The cell-wise difference

$$\Delta\mathrm{IoU}^{(L_s,L_r)} = \overline{\mathrm{IoU}^{(L_s,L_r)}_{\text{within-format}}} - \mathrm{IoU}^{(L_s,L_r)}_{\text{cross-format}}$$

isolates the contribution of the TF to the structural overlap. Within-format pairs are constructed pairwise from CoLA, STS-B, MNLI, and RewardBench and their classification variants; cross-format pairs match each dataset against its own counterpart. By construction, positive cells in the resulting heatmap (Figure 19) localize edges that two same-format circuits share but the matched cross-format pair does not, i.e., edges attributable to the TF.

Across the models, $\Delta\mathrm{IoU}^{(L_s,L_r)}$ concentrates on mid-to-late layers around the diagonal band, consistent with the LE/TF decomposition of §4.1. On the larger modular models (`Qwen2.5-14B`, `Gemma-3-27B`), the dispersion narrows and positive mass falls on a smaller set of (sender, receiver) pairs, consistent with the scale-dependent redundancy effect documented in Appendix M.

Cross-task layer-wise overlap. Figure 20 reports per-layer Edge IoU at $K = 200$ between STS-B and every other task (CoLA, MNLI, RewardBench, and its own cross-format variant STS-B_CLASS). Layers are binned into 10 relative-depth bins for cross-model comparison.

Two reference bands frame the cross-task lines. The lower band is the random-edge baseline (App. M): edges sampled uniformly from each task's full PEAP edge set, bucketed identically, with 95% CI over 200 trials. The upper band is the within-task split-half ceiling (App. M), extended per-layer with the same bucketing; it reports mean $\pm$ standard deviation across the 10 splits, Spearman–Brown corrected to project full-N reliability.

Across the models, the rating-format cross-task curves (STS-B vs. CoLA, MNLI, RewardBench) rise above the random floor by the mid layers and remain elevated through the final layers, locating the shared-trunk overlap of on the depth axis. The matched-dataset cross-format curve (STS-B vs. STS-B_CLASS) tracks the same trajectory through the mid layers but ranks last among the four cross-task lines in the final layers.

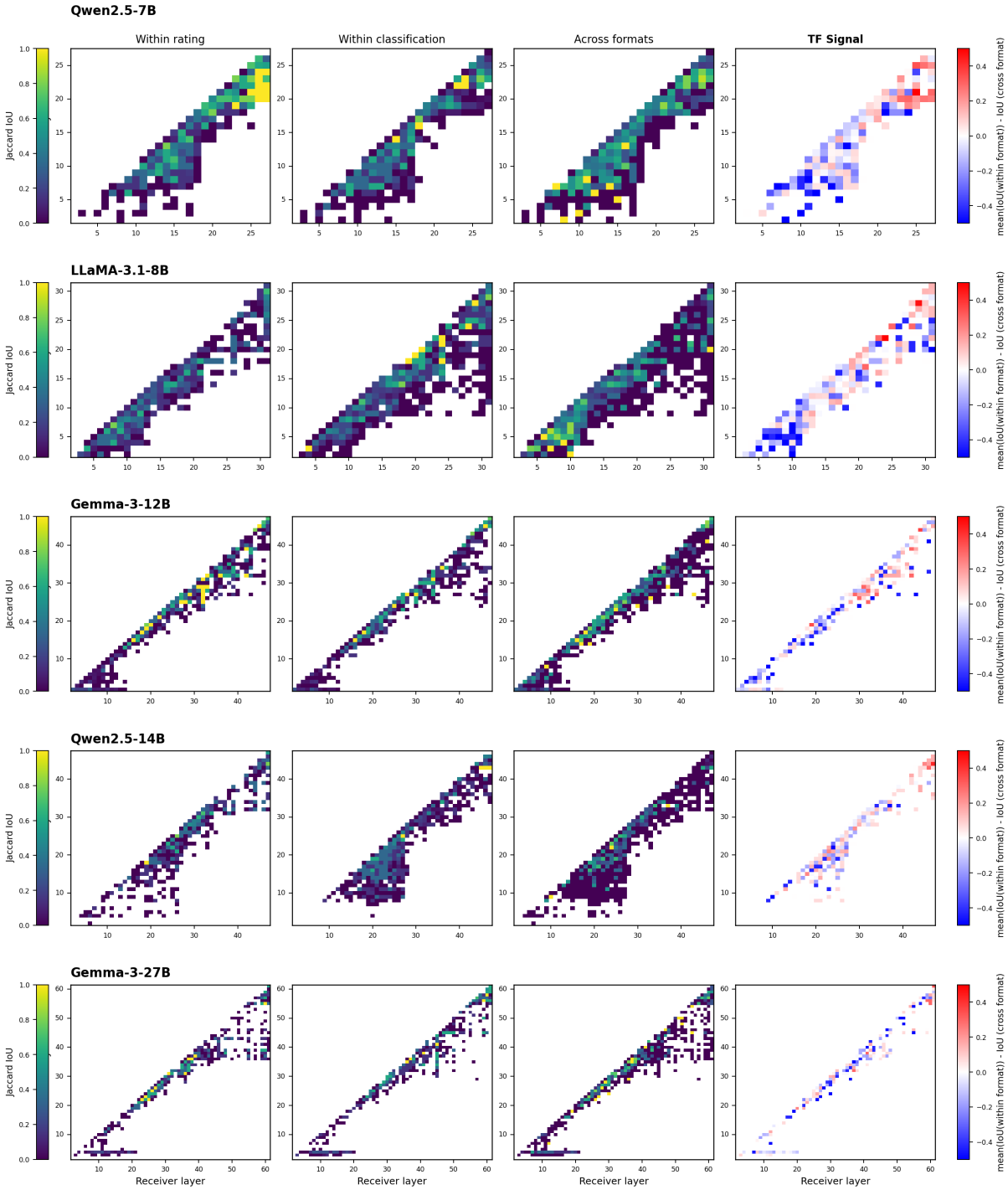


Figure 19: Per-model (sender, receiver) Edge IoU grids at top-200, one row per model. The first three columns show mean Jaccard IoU within rating pairs, within classification pairs, and across tasks; the fourth column shows the TF signal (App. P). Positive (red) cells in the TF signal panel localize edges shared by same-format pairs but absent from the matched cross-format pair, isolating the TF contribution.

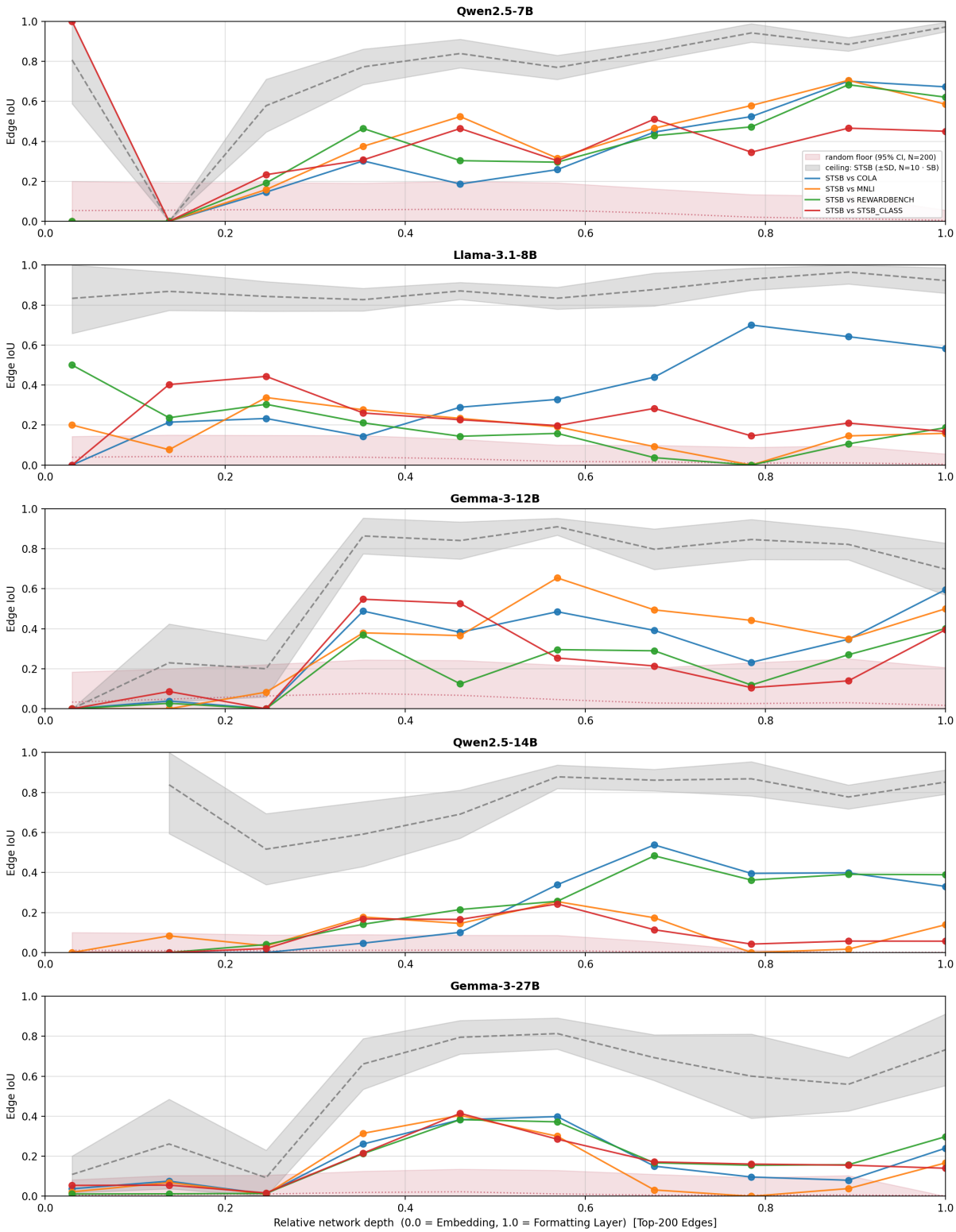


Figure 20: Per-layer Edge IoU at $K = 200$ between STS-B and every other task across the evaluated models, binned into 10 relative-depth bins. The lower band is the random-edge baseline (95% CI over 200 trials); the upper band is the within-task split-half ceiling. The ceiling is undefined for `Qwen2.5-14B` at the leftmost bin because no top-K edge lands there in any partition.

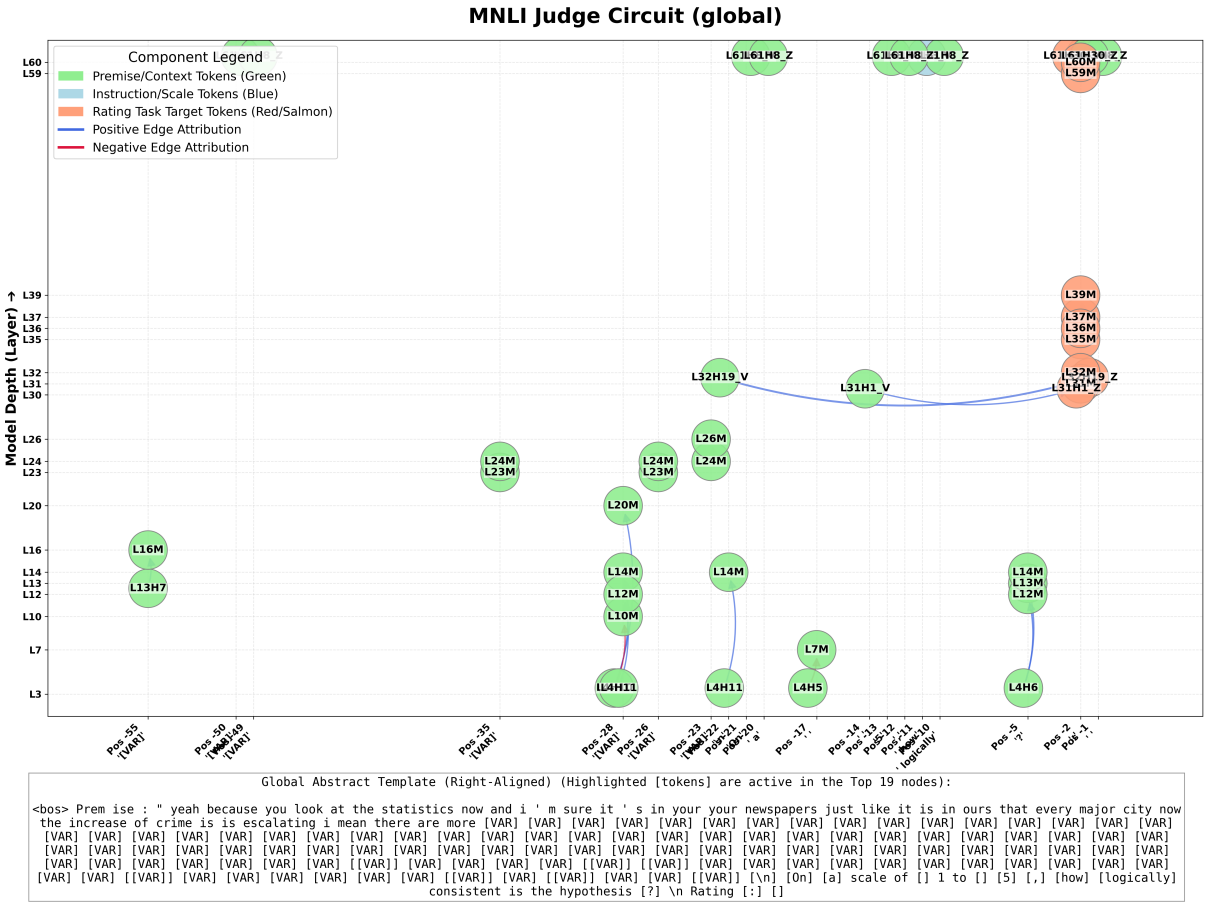
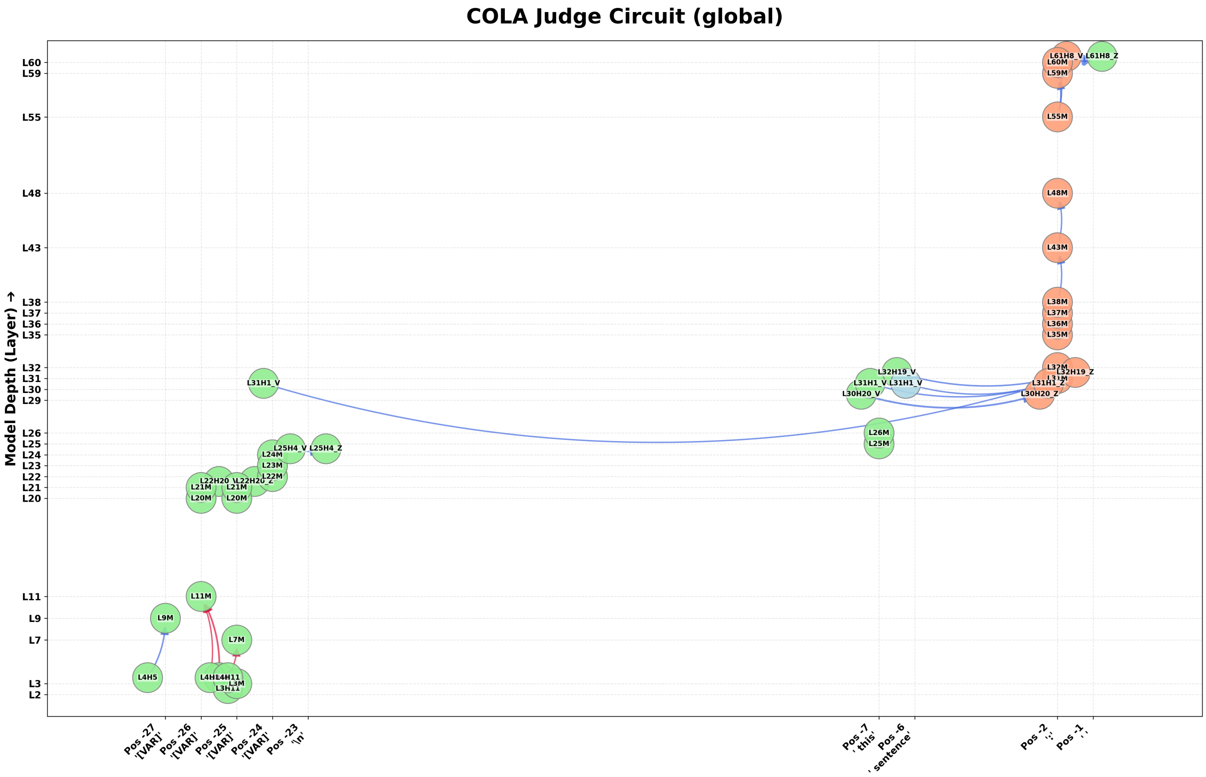


Figure 21: Global judge circuit for MNLI on Gemma-3-27B.



Global Abstract Template (Right-Aligned) (Highlighted [tokens] are active in the Top 9 nodes):

<bos> Sentence [VAR] [[VAR]] [[VAR]] [[VAR]] [[VAR]] [\n] On a scale of 1 to 5 , how grammat ically acceptable is [this] [sentence] ? \n Rating [:] []

Figure 22: Global judge circuit for CoLA on Gemma-3-27B.

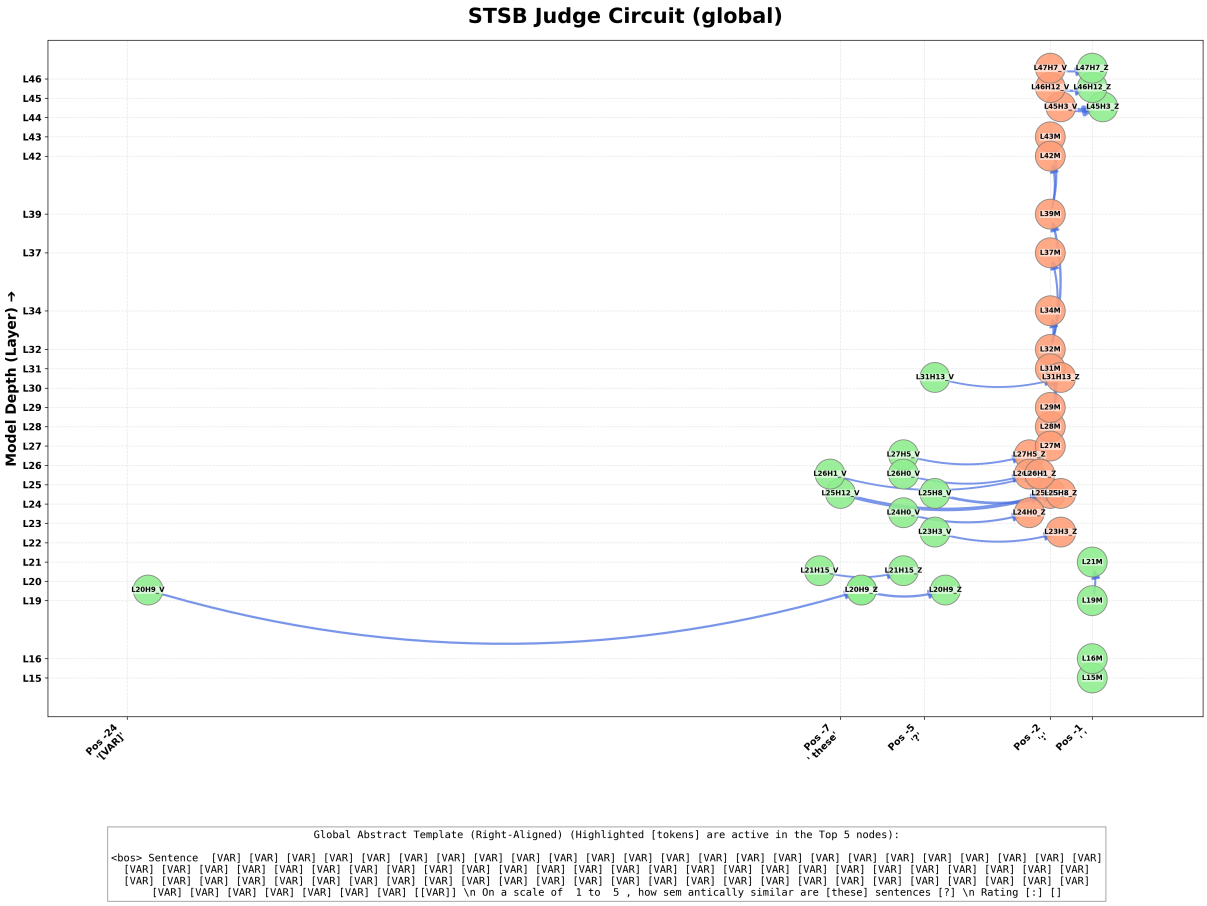


Figure 23: Global judge circuit for STS-B on Gemma-3-12B.

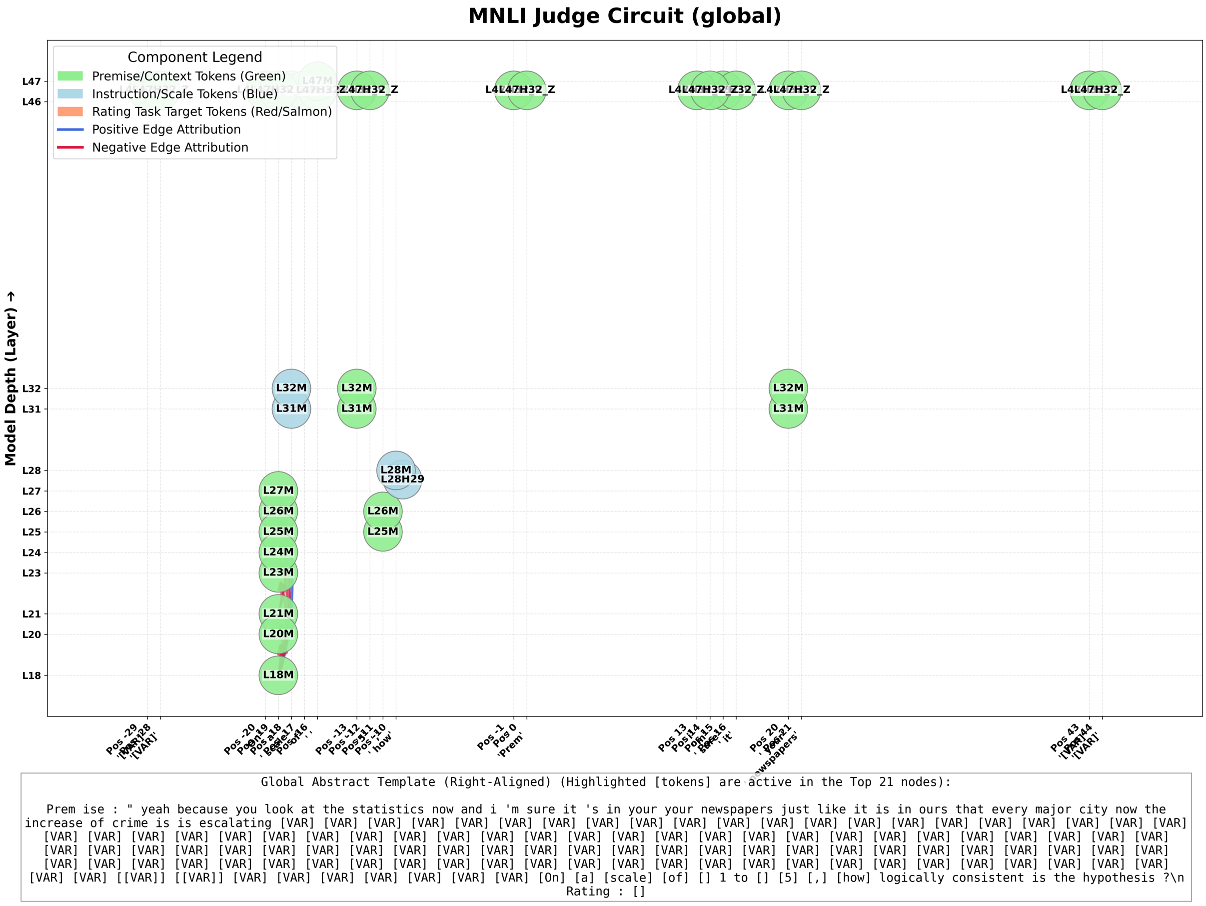
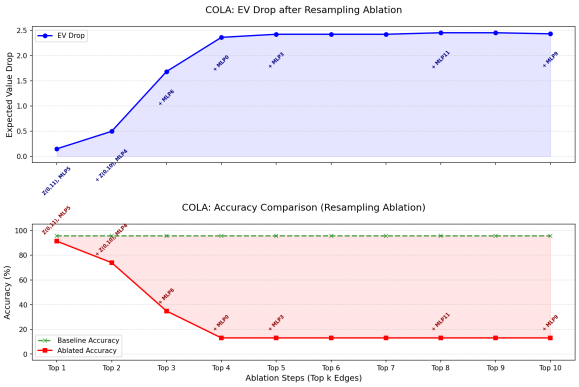


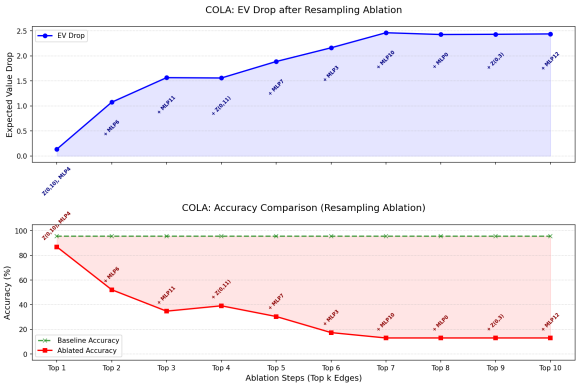
Figure 24: Global judge circuit for MNLI on Qwen2.5-14B.

Head	Description
Shared Evaluators	
L20H9	Contrastive token detector; attends to semantically differentiating tokens in the second sentence and simultaneously distributes the resulting comparison signal into all three downstream pathways (shared evaluation, rating-specific, and classification-specific)
L24H0	Routes within the late instruction span, consolidating information across the question template tokens without forwarding to the Rating position
L25H8	Routes from the sentence-final question mark toward the output boundary region, propagating the consolidated judgment signal rather than contributing directly to the Rating token
L25H12	Aggregates from both the second sentence and the full instruction span toward the Rating position, serving as a key bridge between the second sentence representation and the final output
L26H0	Routes within the instruction span toward the late boundary region around the question mark, functioning as a straightforward instruction-level consolidator
L26H1	Routes from both the late instruction span and the second sentence toward the output boundary region spanning the question mark, Rating token, and colon
L44H8	Routes within the late instruction span toward the output boundary with minimal variation
L45H3	Consistently routes from the end of the instruction span to the Rating position; exhibits the same terminal forwarding behavior in both CoLA and STS-B circuits
L46H12	Mirrors L45H3 in consistently routing from the late instruction span to the Rating position across both tasks
L47H7	Format-agnostic terminal router, consistently forwarding from the late instruction region to the Rating position regardless of task or assigned role
Rating Formatters	
L18H10	Routes from the second sentence span toward mid-instruction positions, with occasional local intra-sentence activity
L19H0	Routes primarily from semantically relevant tokens within the second sentence toward the instruction span, with very occasional contributions from the first sentence
L21H15	Routes between neighboring tokens within the second sentence while also forwarding toward the instruction boundary, performing local semantic integration within the second sentence
L23H3	Routes from the second sentence span toward the instruction region while also routing within the instruction span toward its later parts
L26H5	Routes within the mid-instruction span toward the late instruction region with minimal attribution
L27H5	Routes from the instruction span toward the end of the instruction and the start of the output boundary region
L31H13	Routes from the late instruction span toward the output boundary region spanning the question mark and Rating token
Class Formatters	
L20H2	Routes from both the first sentence content tokens and the instruction span toward the "sentences" boundary token, consolidating semantic content at the final instruction boundary
L21H15	Routes between neighboring tokens within the second sentence while forwarding toward the instruction boundary; dual role reflects format-agnostic intra-sentence integration
L24H13	Routes within the late instruction span toward the output boundary, with occasional contributions from the second sentence
L24H15	Distributes signal across various instruction tokens without a strong directional preference toward the output
L26H0	Routes within the instruction span toward the late boundary region; see shared evaluator entry
L26H10	Routes from the mid-instruction span toward the output position with predominantly negative attribution
L27H8	Routes from the end of the instruction span toward the Rating position with consistently negative attribution, functioning as an inhibitory terminal head
L27H9	Routes within the mid-to-late instruction span with minimal attribution
L43H3	Selectively picks up an accumulated signal from a specific second sentence position alongside instruction tokens, rarely active
L47H6	Inhibitory terminal head applying consistently negative attribution at the Rating position, dampening competing format signals at the output boundary

Table 10: STS-B circuit attention head functional roles derived from per-head PEAP attribution analysis. Heads are grouped by role and sorted by layer within each group. Heads with multiple roles (L20H9, L21H15, L24H0, L25H12, L26H0, L26H1) appear in their primary role group.

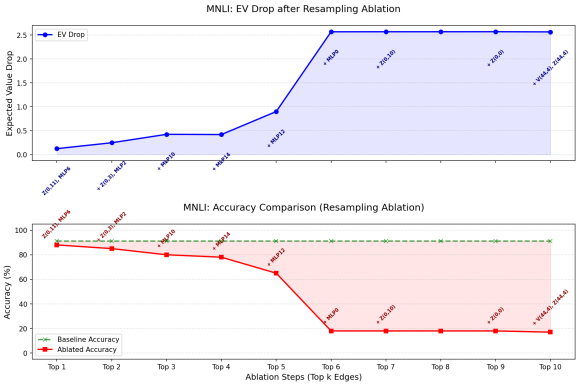


(a) Classification Ablation.

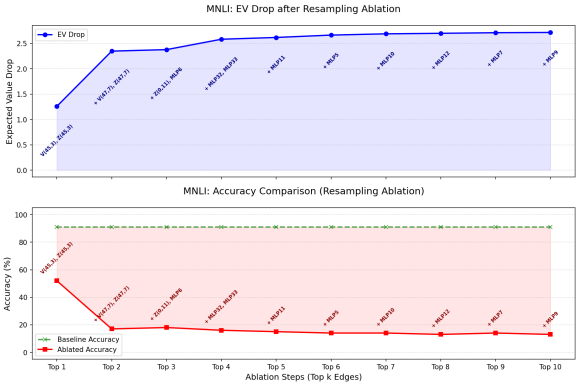


(b) Numerical Judgment Ablation.

COLA dataset.

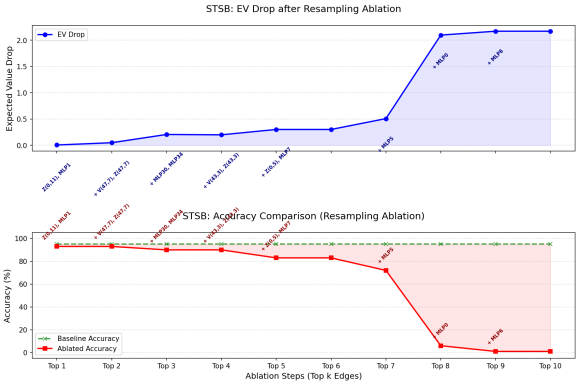


(c) Classification Ablation.

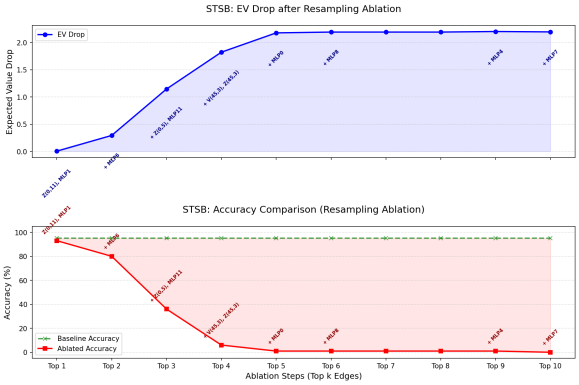


(d) Numerical Judgment Ablation.

MNLI dataset.



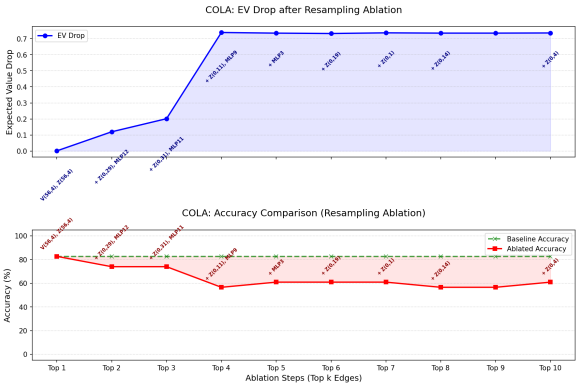
(e) Classification Ablation.



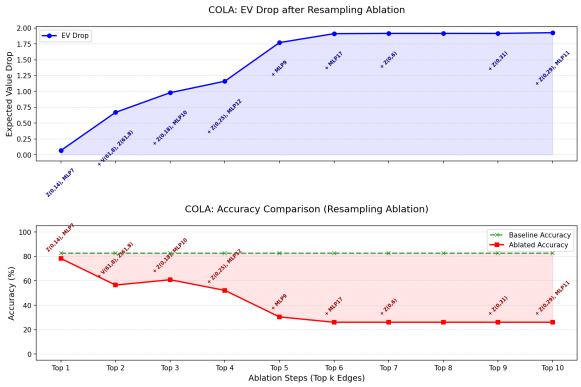
(f) Numerical Judgment Ablation.

STSB dataset.

Figure 25: Ablation phase-transition study (Gemma-3-12B).

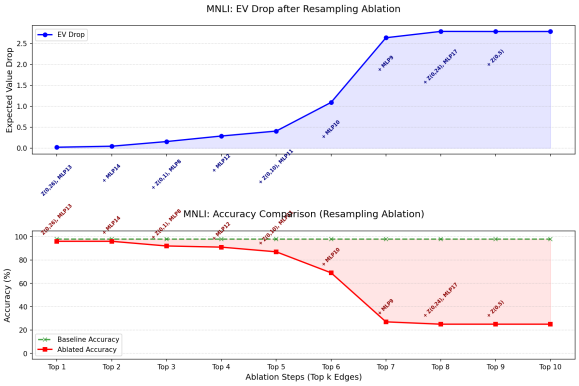


(a) Classification Ablation.

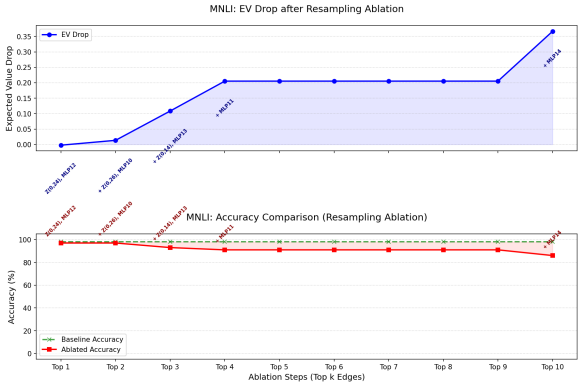


(b) Numerical Judgment Ablation.

COLA dataset.

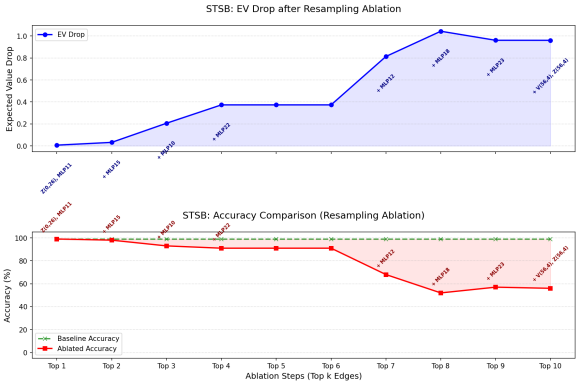


(c) Classification Ablation.

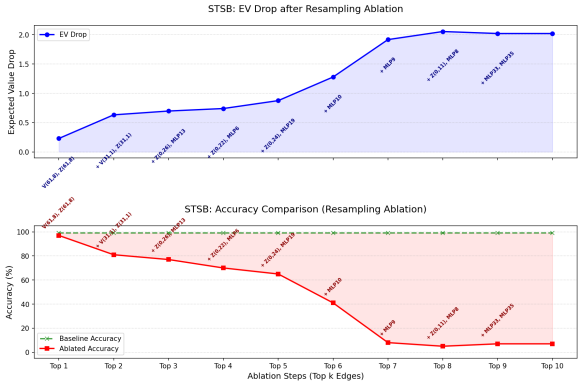


(d) Numerical Judgment Ablation.

MNLI dataset.



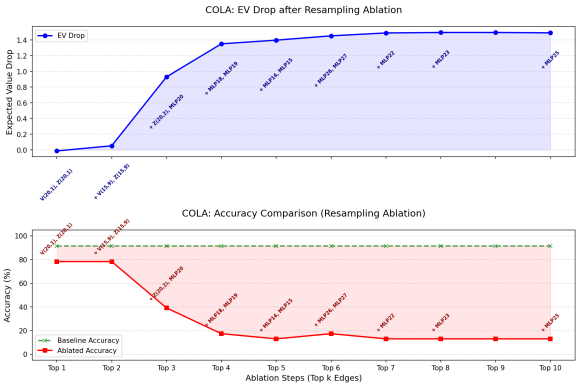
(e) Classification Ablation.



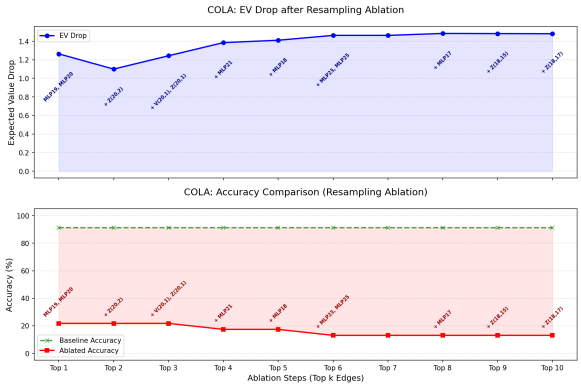
(f) Numerical Judgment Ablation.

STSB dataset.

Figure 26: Ablation phase-transition study (Gemma-3-27B).

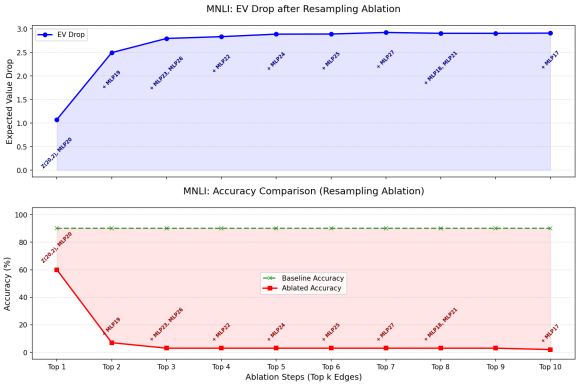


(a) Classification Ablation.

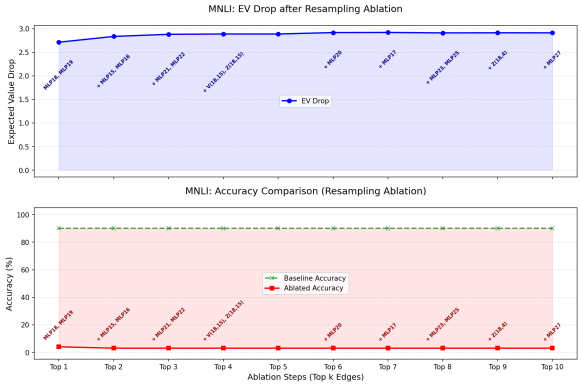


(b) Numerical Judgment Ablation.

COLA dataset.

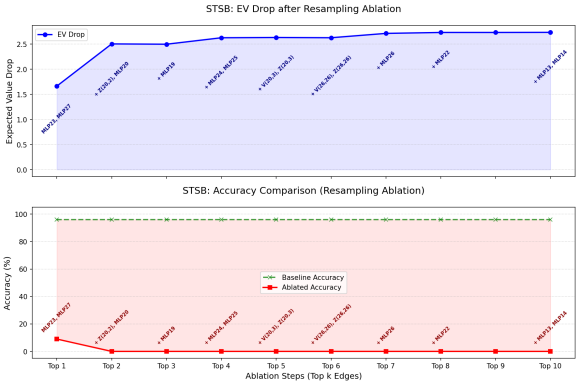


(c) Classification Ablation.

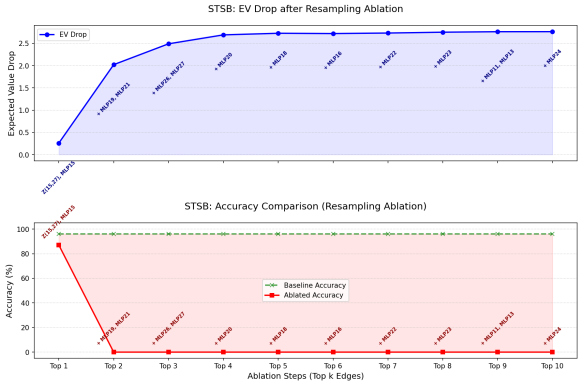


(d) Numerical Judgment Ablation.

MNLI dataset.



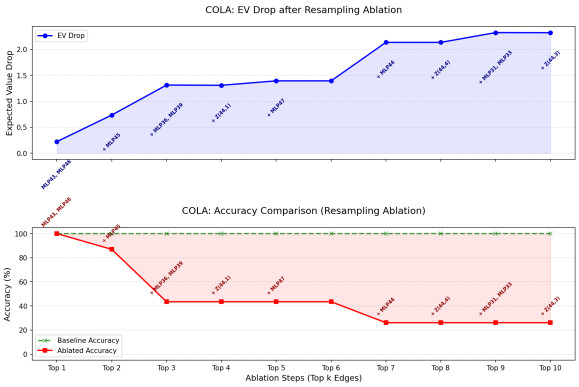
(e) Classification Ablation.



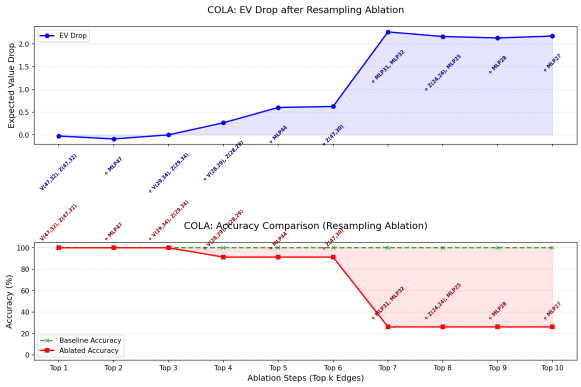
(f) Numerical Judgment Ablation.

STSB dataset.

Figure 27: Ablation phase-transition study (Qwen2.5-7B-Instruct).

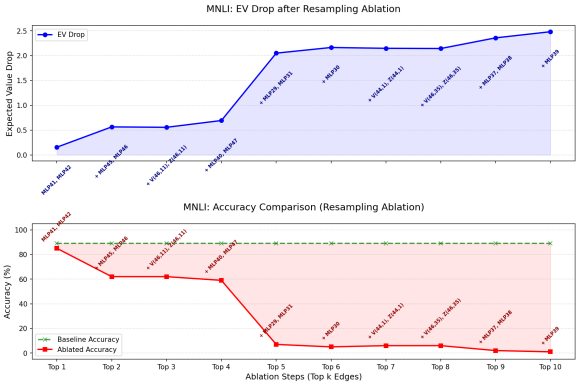


(a) Classification Ablation.

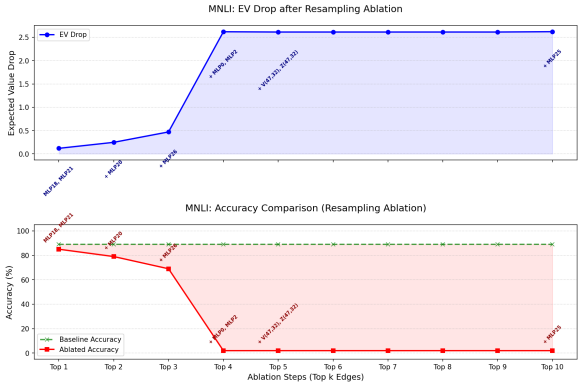


(b) Numerical Judgment Ablation.

COLA dataset.

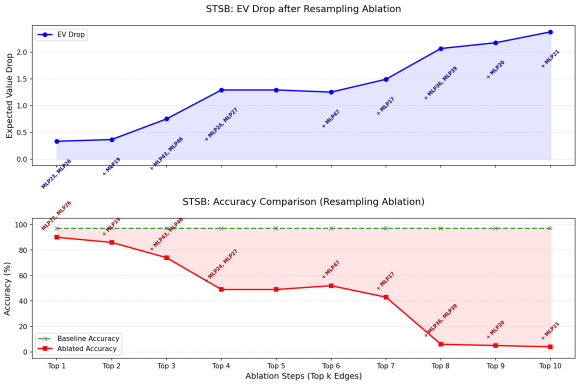


(c) Classification Ablation.

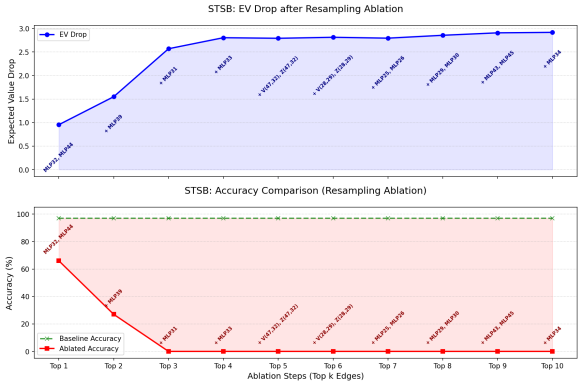


(d) Numerical Judgment Ablation.

MNLI dataset.



(e) Classification Ablation.



(f) Numerical Judgment Ablation.

STSB dataset.

Figure 28: Ablation phase-transition study (Qwen2.5-14B-Instruct).