JUDGE CIRCUITS EXPLAIN FORMAT-INDUCED INCONSISTENCY IN LLM-AS-A-JUDGE

Anonymous authors
Paper under double-blind review

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 five open-weight instruction-tuned models (Gemma-3, Qwen2.5, Llama-3.1) across five judgment tasks. We find that judgments across structured understanding and open-ended preference tasks share a sparse *Latent Evaluator* sub-graph in the mid-to-late layers; zero-ablating it collapses judgment while damaging knowledge probes substantially less than a random ablation of equal size 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. The judgment itself can therefore be read out independently of the requested output format. Our findings imply that benchmark comparisons of judge reliability across formats partly measure the fragile formatting stage, and can understate the quality of the underlying evaluation.

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 (Zheng et al., 2023; 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 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. The latter failure mode is the phenomenon this paper explains, and we can exhibit concretely: on the STS-B minimal pair in Figure 2, Gemma-3-12B rates the semantic similarity of a matched sentence pair (*a boy jumping into a lake*, described twice) at 3 of 5 under the rating format, yet answers "No" when the same instance is posed as a binary similar/dissimilar classification. The content and the model are identical; only the requested output format changed, and the model's commitment changed with it.

Even large proprietary models show both failure modes, 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 (response length, sentiment polarity) and skip the integration of input and target aspects that holistic evaluation requires. Comparable inconsistency and calibration failures hold for judges of < 70B parameters (Girrbach et al., 2025).

No prior work has causally investigated the internal computational 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.

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

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 layers, recoverable at top-$k \leq 200$ edges.

(2) **We show that judgment modularity is architecture-dependent.** On modular models, zero-ablating the Latent Evaluator collapses judgment while damaging the knowledge probes we test substantially less than a size- and type-matched random-ablation control; Gemma-3-12B is the only model whose Latent Evaluator damages the verification probes beyond that control, while a smaller entanglement with pure factual recall appears in both Gemma-3 scales (§4.2).

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

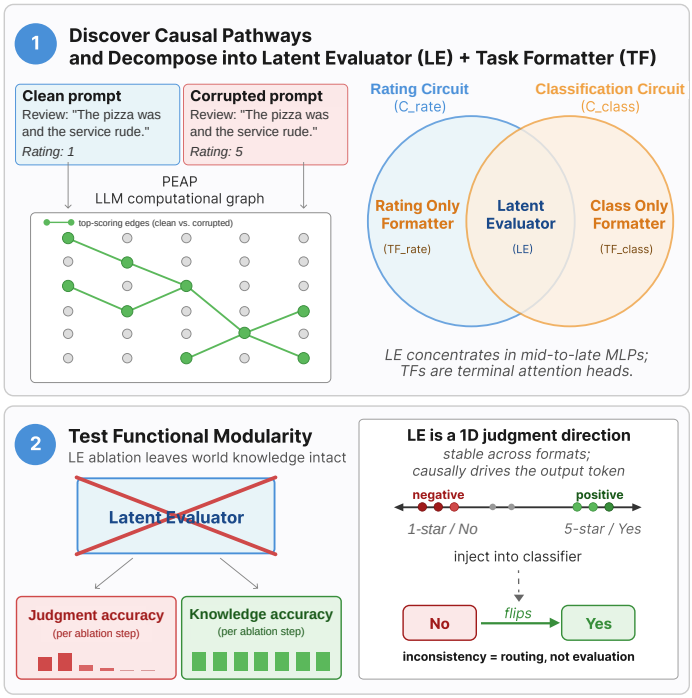


Figure 1: Overview on a Yelp minimal pair: (**1**) PEAP traces the cross-token causal edges that drive the difference between the clean and corrupted run (green) into a shared *Latent Evaluator* sub-circuit ($\mathcal{C}_{\text{LE}} := \mathcal{C}_{\text{rate}} \cap \mathcal{C}_{\text{class}}$), leaving format-specific Task Formatters ($\mathcal{C}_{\text{TF,rate}}, \mathcal{C}_{\text{TF,class}}$) in the two circuits' disjoint parts. (**2**) Zero-ablation (red $\times$) separates evaluation from world knowledge: as more LE edges are ablated (bars), judgment accuracy on the rating tasks collapses while knowledge-probe accuracy (MMLU, factual QA) drops substantially less than under a random ablation of equal size; Boundless DAS (BDAS) finds a 1D judgment direction in the LE's activation space; Task Formatters in terminal layers map that judgment scalar to the target token.

Together, these results suggest that LaaJ format inconsistency is a routing problem, so fixes can target the formatter and leave the model's judgment competence untouched. [1]

2 Experimental Setup

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

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

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. The output is a scalar rating or a categorical verdict over the candidate. 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 & 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. Within-task circuit IoU is comparable across these subset sizes (split-half reliability, App. P); the compute constraints behind the cap and our backward-pass tracing budget are deferred to App. H.

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 per-edge attributions are symmetric by construction (§3.1). We format every input as a 1–5 rating prompt; for contrastive circuit analysis (§4.1), we pair each dataset with a parallel classification-control prompt (categorical Yes/No, True/False, or Entailment/Contradiction labels) on the same instances. Templates and alignment details are in Appendices G–H.

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 multi-layer perceptrons (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).

[2]For MNLI, pairs are drawn from the entailment, contradiction subset so that clean and corrupted prompts have semantically opposed ground truth (App. H).

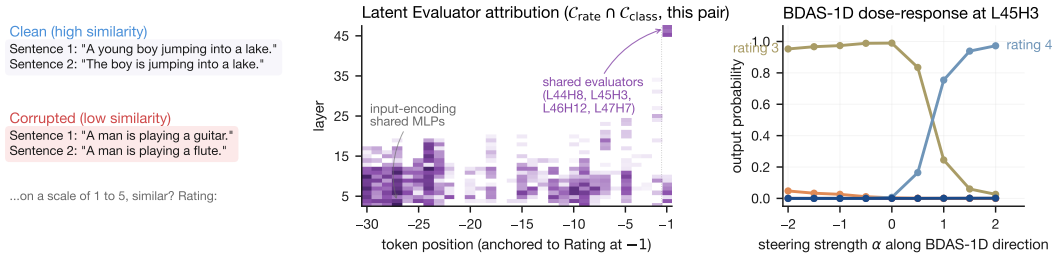


Figure 2: Anatomy of judgment on one STS-B pair (Gemma-3-12B, #149). **Left**: clean (high-sim.) and corrupted (low-sim.) prompts. **Middle**: PEAP attribution restricted to $\mathcal{C}_{\text{LE}} := \mathcal{C}_{\text{rate}} \cap \mathcal{C}_{\text{class}}$ (§4.1, top-1000 aggregate): early shared input-encoding MLPs plus late shared evaluators (App. M). **Right**: steering along BDAS-1D at L45H3 (App. E), 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.

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 $(S_{\text{clean}} - S_{\text{corr}})$. Attribution is taken with respect to the model's *expected rating value*, $\text{EV} = \sum_r r \cdot P(\text{rating} = r)$ – the probability-weighted mean of the rating-token distribution at the final position (Appendix B). The clean–corrupted *EV gap* of a minimal pair is the causal quantity that our faithfulness and steering analyses track. A single backward pass yields all receiver gradients, 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 B) that handles our bidirectional minimal pairs (§2) without canceling causal signal under naïve gradient summation. The extracted circuits are faithful to the full model (Appendix D), stable under data resampling (Appendix P), and stable under prompt-instruction paraphrases (Appendix Q).

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 & 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 trunk on every modular architecture.

On Gemma-3-12B at top-200, the circuits of any two of our tasks share about half or more of their components, including pairings of the structured tasks with the open-ended RewardBench (Figure 4; component-level numbers in Appendix C). The same shared-trunk pattern holds across the modular models, with Qwen2.5-7B showing the strongest edge-level agreement on every task pair we test (Figure 5). Qwen2.5-14B and Gemma-3-27B agree less on specific edges but still recruit largely the same components. This matches the scale-dependent redundancy documented in Appendix P: larger modular models route judgment through several equivalent sub-pathways, so the same components appear while the top edges vary across data splits. Within-task split-half overlap on Gemma-3-12B at the same sparsity meets or exceeds the cross-task overlap (Appendix P), so sample-size noise does not produce the shared trunk. A layer-wise decomposition localizes the cross-task overlap to the mid-to-late layers across the evaluated models (App. T), matching the LE/TF depth split developed in §4.1.

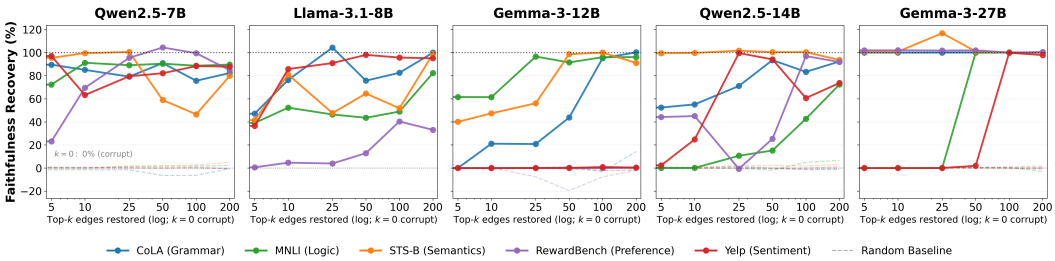


Figure 3: Sparse circuit faithfulness across the five evaluated models and five rating tasks. Each curve traces median MIB recovery as we cumulatively patch the top-k PEAP edges from a fully corrupted forward pass back toward the clean activations. Solid colored lines are the discovered circuits; the 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.

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 faithfulness metric of the Mechanistic Interpretability Benchmark (MIB) (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 D and R).

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

In 21 of the 25 (model, task) combinations we trace, restoring at most 200 edges recovers nearly all of the judgment signal (Figure 3); on Gemma-3-27B the open-ended RewardBench circuit saturates with just five edges. The remaining combinations are Gemma-3-12B on RewardBench and Yelp, where recovery stays near zero and matches that model's entanglement profile (§4.2, Table 1), and Llama-3.1-8B on MNLI (a slower climb) and Yelp (a partial peak that drifts back down). A random-edge baseline (gray dashed line) stays near zero in every configuration, so sparsity alone does not explain the recovery.

Cross-method robustness Different attribution algorithms can select different sparse subgraphs on the same model and task (Méloux et al., 2025), so we re-trace every Qwen2.5-7B and Gemma-3-12B circuit with LRPEAP, a variant we develop that keeps PEAP's position-aware edge formulation but computes the backward pass with LRP rules (Jafari et al., 2025). Both backbones recover the same mid-to-late Latent Evaluator far above the permutation null, and a comparison against ACDC points the same way (Appendix S). The circuits are also robust to prompt wording: re-traced on paraphrased instructions, they overlap the originals about as much as two halves of the same data do, and only renaming the prompt's field labels cuts noticeably into that overlap (Appendix Q).

4 Judge Circuit Modularity is Architecture-Dependent

Building on §3.2, we test whether the shared trunk is a functionally modular sub-system or merely a generic capability bottleneck (Hanna et al., 2026): if zero-ablating the Latent Evaluator collapses judgment but mostly spares world-knowledge benchmarks, the sub-graph is doing judgment-specific work. This 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

Table 1: Zero-ablation knowledge-probe domain control. Each cell reports accuracy (%) as base → LE-ablated, with a size-matched random-ablation control (mean ± SD over 3 trials) in parentheses; the **LE** column gives the composition of the ablated merged Latent Evaluator (attention heads + MLPs), merged across the per-task Latent Evaluators of CoLA, MNLI, and STS-B (App. U). All cells ablate the full LE (heads and MLPs); *entanglement* requires LE-ablation damage *exceeding* the random control. On the four modular models, LE ablation is no more damaging than random on MMLU and formal factual verification (StrategyQA, CREAK). Gemma-3-12B is the only model whose LE ablation falls below the control (bold), mostly due to its three LE attention heads (§4.2). The full circuit-topology panels in Appendix O demonstrate the corresponding two-stage Latent Evaluator / Task Formatter separation across models and tasks, supporting the same generalization.

Model	LE	Clinical DB	Abstract Alg.	Physics	StrategyQA	CREAK
Qwen2.5-7B	1H+9M	78 → 48 (20.3 ± 4.5)	56 → 25 (20.7 ± 1.2)	53 → 38 (23.0 ± 1.4)	66.5 → 58.0 (47.2 ± 3.9)	81.5 → 84.0 (48.0 ± 1.1)
Llama-3.1-8B	0H+8M	76 → 47 (22.3 ± 3.1)	35 → 29 (21.7 ± 2.1)	41 → 24 (21.0 ± 3.6)	67.0 → 61.5 (52.5 ± 1.1)	78.5 → 47.5 (48.7 ± 1.3)
Gemma-3-12B	3H+4M	79 → **18** (55.7 ± 7.8)	47 → **22** (27.7 ± 3.7)	52 → **22** (28.0 ± 4.3)	65.0 → 62.5 (61.3 ± 3.0)	80.5 → 79.5 (76.8 ± 3.2)
Qwen2.5-14B	0H+2M	88 → 85 (63.0 ± 31.2)	59 → 57 (49.3 ± 13.7)	69 → 64 (56.0 ± 14.9)	71.5 → 71.5 (61.8 ± 10.7)	87.0 → 81.0 (74.8 ± 14.1)
Gemma-3-27B	0H+3M	79 → 82 (80.0 ± 2.9)	47 → 47 (44.7 ± 4.0)	62 → 59 (56.7 ± 1.2)	50.5 → 52.5 (48.8 ± 1.9)	82.5 → 82.5 (81.0 ± 3.2)

prompt structure. Their structural overlap decomposes the model's judgment computation 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 sub-circuit highlighted as the Latent Evaluator 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 3: Contrastive tracing yields a clean Latent Evaluator / Task Formatter split.

On Gemma-3-12B (CoLA × CoLA_CLASS, top-200), 3 of 17 analyzed heads act as shared evaluators (most strongly L45H3, L46H12, L47H7), while the remaining 14 split without overlap into rating-specific (9) and classification-specific (5) formatters; Figure 2 (middle) visualizes the same $\mathcal{C}_{\text{LE}}$ on one STS-B pair. Three cross-checks support this decomposition: per-head attribution analysis (App. M), transcoder features at the circuit's MLP positions (App. N), and a layer-wise Edge IoU decomposition (App. T) all confirm the shared-then-specialized two-stage pattern.

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 QA datasets, StrategyQA (Geva et al., 2021) and CREAK (Onoe et al., 2021). These probes mirror our Task Formatter answer tokens ("Yes/No", "True/False"), while relying predominantly on factual retrieval with little abstract quality judgment. Because zeroing several mid-network MLPs is destructive regardless of their function, every ablation is compared against an equally large random set of heads and MLPs sampled uniformly outside the LE over 3 trials. We say the LE is *entangled* with a probe only if LE-ablation damage *exceeds* the random control.

Finding 4: On modular architectures, ablating the Latent Evaluator damages knowledge probes substantially less than random.

Table 1 illustrates that, on the four modular models, LE ablation is less damaging than the random control on every probe but one (Llama-3.1-8B on CREAK). 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 I, Figures 27–28). The Latent Evaluator therefore operates as a specialized sub-system: removing it destroys judgment while damaging the knowledge probes less than removing an equally sized random set of components.

Finding 5: Modularity emerges at family-specific scales.

Gemma-3-12B is the only model whose LE ablation damages the verification probes substantially beyond the random control: full-LE ablation drops MMLU clinical knowledge to 18% against a random control (Table 1). Ablating only its three LE attention heads (L45H3, L46H12, L47H7) already costs 14–18 pp on all three MMLU subsets (clinical $79\% \rightarrow 63\%$; three random heads: no effect), while leaving StrategyQA, CREAK, and pure recall untouched. Two follow-up experiments (App. U) reveal what this entanglement consists of. First, a *pure-recall cloze probe* (declarative PopQA completions with no candidate answers and no verification step) shows a recall drop under LE ablation on Gemma-3-12B and 27B, but never on Qwen or Llama. Second, *factual-recall reference circuits* (CREAK, StrategyQA) traced with the identical PEAP pipeline capture most of the entangled model's LE but far less of the modular control's; this contrast is preserved at every threshold and sample size.

5 Inter-Format Inconsistencies Arise from Format-Specific Output Routing

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

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, then inject those exact activations into the same model while it runs on a corrupted *classification* prompt (whose natural output would be the negative token, e.g., "No"). This is a *blanket* activation transfer: it overwrites the entire LE pattern, in contrast to the targeted 1D subspace interventions of Appendix E. If the Latent Evaluator carries the judgment, the downstream classification head should receive the injected positive signal and flip its output token from "No" to "Yes" or "Entailment". If instead the terminal branches do the actual judgment work, the injection should have no effect. §5.2 returns to the contrast between this whole-pattern transfer and the targeted interventions.

5.1 The Latent Evaluator Carries the Judgment

Finding 6: The Task Formatter decides whether the judgment reaches the output.

The clearest causal demonstration is Qwen2.5-7B: blanket FTI flips the argmax in $\geq$ 99% of MNLI, STS-B, and RewardBench pairs (Table 2), with mean target-class probability rising from $\leq 17\%$ at baseline to $\geq 85\%$ post-injection. 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", never leaving the output format's vocabulary (no pair emits "5"). The judgment representation is stable, 1D, and shared across semantic domains

Table 2: FTI flip rates (%): how often injecting the Latent Evaluator activations of a 5-star rating prompt into a corrupted classification prompt flips the verdict to the positive label (Yes/Entailment). * marks cells with few valid pairs ($N \leq 26$). Full results with probability shifts, pair counts, and Wilson 95% CIs are in Table 5 (App. J).

Model	CoLA	MNLI	STS-B	Yelp	RewardBench
Qwen2.5-7B	100.0*	**99.1**	**100.0**	66.4	**100.0**
Llama-3.1-8B	25.0*	4.2	–	0.0	22.2*
Gemma-3-12B	14.9	3.7	50.0	0.0	7.7*
Qwen2.5-14B	91.7*	4.0	5.1	31.2	41.2
Gemma-3-27B	0.0	0.7	**96.8**	1.1	5.3*

(§4.1, App. E). The bottleneck is the terminal mapping, which varies sharply across tasks (three-way MNLI against binary classification) and across models: some formatters accept the injected pattern (e.g., Qwen2.5-7B) and some do not (e.g., Gemma-3-27B on MNLI). A failed injection could also mean that the Latent Evaluator's pattern is itself format-specific there. The cross-domain steering at the same components (App. E) and the shared late-layer geometry across formats (App. L) both point back to the formatter.

Finding 7: Transferring the evaluator's activations does not flip the verdict under several competing labels.

(a) Competing labels (MNLI). We call a label structure *multi-attractor* when the Task Formatter spreads its projected probability mass across three or more competing target tokens. Applying Logit Lens[3] (decodeing intermediate activations into vocabulary space via the unembedding matrix W_U) to the late-layer Task Formatter components of Gemma-3-27B (App. L, Figure 11) shows MNLI's classification formatter spreading mass across three target tokens (*contradiction*, *entailment*, *neutral*; max/min ratio ≈ 2.7), while STS-B's concentrates on a single token (ratio ≈ 19). This geometric difference *predicts* the FTI behavior: except for Qwen2.5-7B (which flips MNLI near-perfectly), MNLI flip rates collapse to single digits on every model (Table 2). With no unambiguous target in a three-attractor basin, the injected mass fragments and no label reaches argmax.

(b) Selectivity grows with scale (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 LE does not disappear at scale. Sparse circuits still saturate MIB faithfulness (App. D) and 1D steering still moves the output (App. E). Instead, the open-ended Task Formatter becomes more selective. The scalar direction still steers the output, but the formatter no longer accepts the full pattern carried over from the rating prompt.

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. E), reaches every downstream Task Formatter, but only becomes the top token where the formatter concentrates its mass on few enough labels (Table 2, App. L). On RewardBench and Yelp, cumulative patching (App. D) and targeted 1D subspace steering (App. E) move the output while blanket FTI flips only a minority of instances. Steering a single direction succeeds where transferring the whole pattern fails, which we read as the formatter growing more selective with scale. Deploying the LE as a LaaJ signal on open-ended tasks therefore favors targeted subspace interventions over blanket transfer.

Reverse transfer This account makes one more prediction: with five competing rating tokens, injecting a categorical judgment *into* a rating prompt should spread mass without settling on one token. Reverse FTI matches this prediction on most combinations: the expected rating moves upward on a majority of pairs, yet the predicted rating rarely changes (Appendix K, including two cells where ratings do flip at scale and one where the injection lowers ratings instead).

Finding 8: The Latent Evaluator's judgment direction can be read out as a rating without training on human labels, and is most useful on small preference datasets.

A zero-shot 1D readout (BDAS-1D) at the LE site (per-pair dose-response in Figure 2, right) tracks a fully supervised residual probe (Girrbach et al., 2025) within a few points of Spearman ρ on most cells. On small-N preference data, where the probe overfits, it rivals the probe and beats prompted argmax in four of five models. On scale-aligned vocabularies (Yelp 1–5), probability-weighted EV remains stronger (App. F).

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

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

under format perturbations is the expected signature of a stable internal judgment routed through a fragile terminal mapping. This shifts the diagnostic question from *"does the model judge consistently?"* to *"does the formatter for this output specification preserve the underlying judgment?"*.

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. What drives the Qwen vs. Gemma scale contrast (pretraining data, post-training, attention sparsity, normalization placement) is a natural follow-on.

The practical implication has a regime caveat: latent signals outperform prompted Likert outputs (Girrbach et al., 2025), and we causally identify the subspace they are recovered from. 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 stronger.

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 (chain-of-thought, structured generation, tool calling). If it does, the LaaJ inconsistency we pin down here is one instance of a broader routing-vs-computation dissociation.

7 Related Work

Behavioral critiques of LaaJ validity. LLM judges have documented biases since the paradigm's inception (position, verbosity, self-preference; Zheng et al., 2023). 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 in a large-scale comparison: 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 originates in the terminal formatting stage, which benchmark comparisons across formats partly measure.

Mechanistic precedents. Our work joins three lines of evidence: cross-task circuit overlap (Tigges et al., 2024; Ferrando & 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 layers, functionally modular on most architectures we study but entangled with world-knowledge pathways on Gemma-3-12B: modularity depends on architecture, not on scale alone. Its 1D causal direction recovers a supervised linear-probe judgment signal zero-shot and exceeds it on small-N preference data, locating the latent signal that practical reference-free rating methods rely on.

AI use statement

In this work, we used generative AI tools (Claude Code and pi.dev with Kimi-K3) **(a)** to implement major parts of the experiment code, **(b)** to clean and reformat intermediate result files, **(c)** to create and modify scientific figures, and **(d)** to try out different phrasings of the paper text for readability, where the final choice was always made by the authors. We have not used generative AI tools to propose or refine hypotheses, conduct literature research, design the research methodology or experiments, formulate mathematical claims, or interpret results. We have reviewed all AI-assisted work: AI-generated code was verified and tested for correctness by the authors, and all AI-edited text was checked against the underlying experimental results. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

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, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 238–255, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-252-7. doi: 10.18653/v1/2025.acl-short.20. URL https://aclanthology.org/2025.acl-short.20/.

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

Nitay Calderon, Roi Reichart, and Rotem Dror. The alternative annotator test for LLM-as-a-judge: How to statistically justify replacing human annotators with LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16051–16081, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.782. URL https://aclanthology.org/2025.acl-long.782/.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens (eds.), *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://aclanthology.org/S17-2001/.

Khaoula Chehbouni, Mohammed Haddou, Jackie CK Cheung, and Golnoosh Farnadi. 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*, 2025. URL https://openreview.net/forum?id=yqKfMr0yvY.

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

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=89ia77nZ8u.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=J6zHcScAo0.

Leon Eshuijs, Shihan Wang, and Antske Fokkens. Short-circuiting shortcuts: Mechanistic investigation of shortcuts in text classification. In Gemma Boleda and Michael Roth (eds.), *Proceedings of the 29th Conference on Computational Natural Language Learning*, pp. 105–125, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-271-8. doi: 10.18653/v1/2025.conll-1.8. URL https://aclanthology.org/2025.conll-1.8/.

Javier Ferrando and Marta R. Costa-jussà. On the similarity of circuits across languages: a case study on the subject-verb agreement task. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10115–10125, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.591. URL https://aclanthology.org/2024.findings-emnlp.591/.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *arXiv*, abs/2101.00027, 2020. URL https://arxiv.org/abs/2101.00027.

Mingqi Gao, Xinyu Hu, Xunjian Yin, Jie Ruan, Xiao Pu, and Xiaojun Wan. LLM-based NLG evaluation: Current status and challenges. *Computational Linguistics*, 51:661–687, June 2025. doi: 10.1162/coli_a_00561. URL https://aclanthology.org/2025.cl-2.9/.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 04 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00370. URL https://doi.org/10.1162/tacl_a_00370.

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

Elena Golimblevskaia, Aakriti Jain, Bruno Puri, Ammar Ibrahim, Wojciech Samek, and Sebastian Lapuschkin. Circuit insights: Towards interpretability beyond activations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2Jyb1yu3nN.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv*, abs/2407.21783, 2024. URL https://arxiv.org/abs/2407.21783.

Tal Haklay, Hadas Orgad, David Bau, Aaron Mueller, and Yonatan Belinkov. Position-aware automatic circuit discovery. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2792–2817, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.141. URL https://aclanthology.org/2025.acl-long.141/.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TZ0CCGDcuT.

Michael Hanna, Yonatan Belinkov, and Sandro Pezzelle. Are formal and functional linguistic mechanisms dissociated in language models? *Computational Linguistics*, pp. 1–41, 03 2026. ISSN 0891-2017. doi: 10.1162/COLI.a.24. URL https://doi.org/10.1162/COLI.a.24.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Farnoush Rezaei Jafari, Oliver Eberle, Ashkan Khakzar, and Neel Nanda. Relp: Faithful and efficient circuit discovery in language models via relevance patching. *arXiv*, abs/2508.21258, 2025. URL https://arxiv.org/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. RewardBench: Evaluating reward models for language modeling. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1755–1797, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.96. URL https://aclanthology.org/2025.findings-naacl.96/.

Michael Lan, Philip Torr, and Fazl Barez. Towards interpretable sequence continuation: Analyzing shared circuits in large language models. In Yaser Al-Onaizan, Mohit Bansal, and

Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12576–12601, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.699. URL https://aclanthology.org/2024.emnlp-main.699/.

Noah Lee, Jiwoo Hong, and James Thorne. Evaluating the consistency of LLM evaluators. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10650–10659, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.710/.

Michael A. Lepori, Thomas Serre, and Ellie Pavlick. Uncovering intermediate variables in transformers using circuit probing. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=gUNeyiLNxr.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: A comprehensive survey on llm-based evaluation methods, 2024. URL https://arxiv.org/abs/2412.05579.

Callum McDougall, Arthur Conmy, János Kramár, Tom Lieberum, Senthooran Rajamanoharan, and Neel Nanda. Gemma scope 2: Suite of SAEs and transcoders for Gemma 3, 2025. URL https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/gemma-scope-2-helping-the-ai-safety-community-deepen-understanding-of-complex-language-model-behavior/Gemma_Scope_2_Technical_Paper.pdf.

Maxime Méloux, Silviu Maniu, François Portet, and Maxime Peyrard. Everything, everywhere, all at once: Is mechanistic interpretability identifiable? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5IWJBStfU7.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple Word2Vec-style vector arithmetic. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5030–5047, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.281. URL https://aclanthology.org/2024.naacl-long.281/.

Joseph Miller, Bilal Chughtai, and William Saunders. Transformer circuit evaluation metrics are not robust. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=zSf8PJyQb2.

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, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=sSrOwve6vb.

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. The quest for the right mediator: Surveying mechanistic interpretability for nlp through the lens of causal mediation analysis. *Computational Linguistics*, pp. 1–48, 02 2026. ISSN 0891-2017. doi: 10.1162/COLI.a.572. URL https://doi.org/10.1162/COLI.a.572.

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

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

URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/5737c6ec2e0716f3d8a7a5c4e0de0d9a-Paper-round2.pdf.

Bruno Puri, Aakriti Jain, Elena Golimblevskaia, Patrick Kahardipraja, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. FADE: Why bad descriptions happen to good features. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17138–17160, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.881. URL https://aclanthology.org/2025.findings-acl.881/.

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, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv*, abs/2412.15115, 2025. URL https://arxiv.org/abs/2412.15115.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

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

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.25. URL https://aclanthology.org/2024.blackboxnlp-1.25/.

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, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan

Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report. *arXiv*, abs/2503.19786, 2025. URL https://arxiv.org/abs/2503.19786.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. Llm circuit analyses are consistent across training and scale. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 40699–40731. Curran Associates, Inc., 2024. doi: 10.52202/079017-1287. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/47c7edadfee365b394b2a3bd416048da-Paper-Conference.pdf.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv*, abs/2308.10248, 2024. URL https://arxiv.org/abs/2308.10248.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 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, 2020. Curran Associates Inc. ISBN 9781713829546. URL https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040/.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=nRfClnMhVX.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. Jump to conclusions: Short-cutting transformers with linear transformations. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 9615–9625, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.840/.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 649–657, Cambridge, MA, USA, 2015. MIT Press. URL https://proceedings.neurips.cc/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.

A LIMITATIONS

A primary limitation of our mechanistic investigation is the **compute cost** of tracing large architectures end-to-end. For context, mapping the CoLA judgment graph in Gemma-3-12B requires evaluating approximately 1.46 million candidate edges. While PEAP traces cross-position edges efficiently, edge patching at this density – especially on the largest models like Gemma-3-27B (roughly 50,000 components) – required us to cap each task's dataset at 100 to 500 minimal pairs. We partially mitigate this concern by reporting split-half circuit reliability (Appendix P): 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 cover grammar, entailment, sentiment, and preference judgments, **how models route highly subjective** or culturally biased evaluations remains open 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 D, 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 O).

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 F) 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 S), 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 3–28$\times$ enrichment over the permutation null in every cell; and component-level cross-checks via the per-head PEAP attribution analysis (Appendix M) and the transcoder feature analysis at circuit-identified MLP positions (Appendix N). The comparison against ACDC (Appendix S.3), whose iterative edge-pruning objective differs more sharply from PEAP than LRPEAP's rule-level substitution, covers both models but is restricted by compute to a small pair subset on one task pair; extending it to further tasks would strengthen the method-independence claim. The transcoder analysis (Appendix N) 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. The factual-recall reference circuits of Appendix U now provide the non-judgment external reference this com-

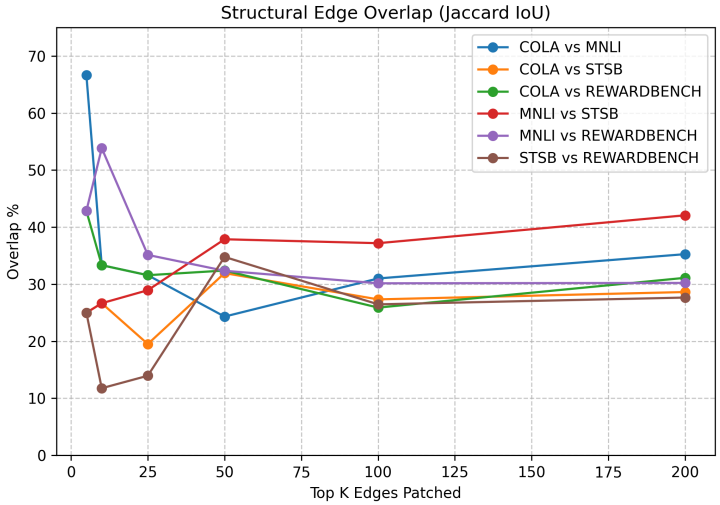


Figure 4: Cross-task Edge IoU on Gemma-3-12B across Top-K patching thresholds. Edges are PEAP-attributed connections between sub-components; higher curves indicate more shared structure at a given sparsity.

parison previously lacked, but only for two recall tasks on two models; extending the reference-circuit panel to all five models would further sharpen the interpretation of the LaaJ shared-trunk magnitude.

B 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 $m = \mathrm{sgn}(\mathrm{EV}_{\text{clean}} - \mathrm{EV}_{\text{corr}})$ 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.

C 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

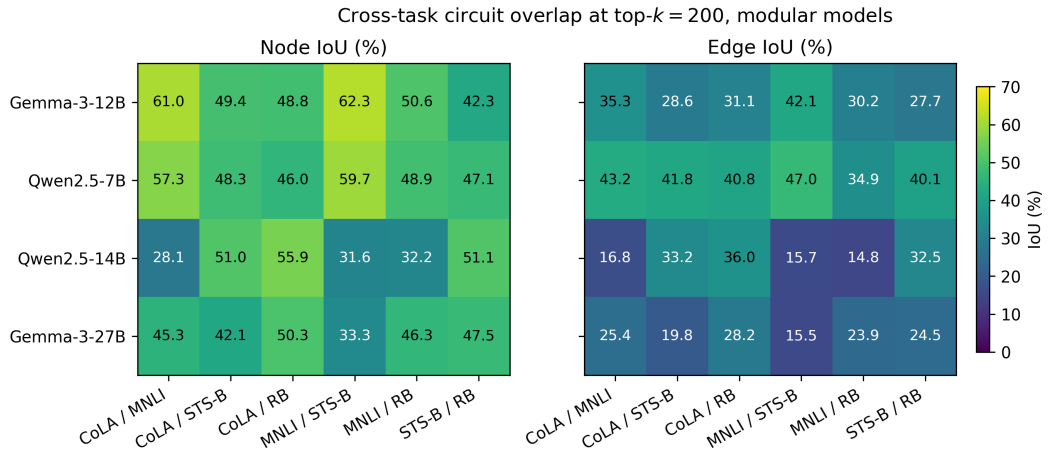


Figure 5: Cross-task circuit overlap at top-200 across all four architecturally modular models. The shared trunk is recoverable on every model, but the per-pair magnitude reflects each model's circuit redundancy (§3.2, App. P): smaller models route through fewer equivalent paths, so their top-200 edges 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.

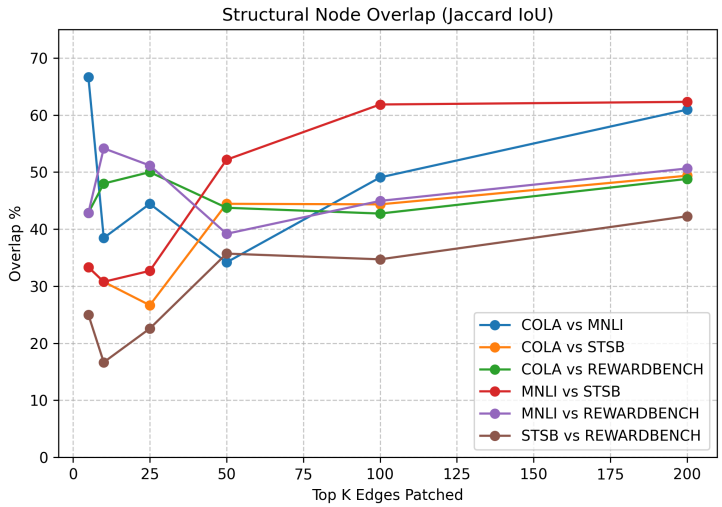


Figure 6: Cross-task Node IoU on Gemma-3-12B across Top-K patching thresholds. Companion to Figure 4.

specific cross-token connections that Edge IoU restricts to. Across all task pairs the Node IoU curve lies 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.

D Circuit Faithfulness

D.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 R. We also cross-validate the resulting circuits with two independent causal probes (activation steering and DAS alignment, Appendix E; 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:

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

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 R as a sensitivity analysis.

D.2 Results

Per-cell saturation points. The 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 reaches ≈ 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 $\times$ 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 $(\text{EV}(\mathcal{C}_k) - \text{EV}_{\text{corr}})/(\text{EV}_{\text{clean}} - \text{EV}_{\text{corr}})$ 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 directly follows 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 P), but the top-200 subgraph is not sparse-recoverable because the circuit is densely interleaved with non-judgment pathways (Table 1, Appendix U). The flat shape therefore encodes an architectural property of Gemma-3-12B. We treat it as a bounded scope condition on the sparse-circuit claim and say so explicitly in the Limitations (§A).

E CAUSAL SUBSPACE STEERING OF THE LATENT EVALUATOR

E.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.

We probe the Latent Evaluator's judgment encoding with two complementary interventions that operate at different granularities and involve no shared machinery: a training-free *directional mean-difference steering* protocol applied at every PEAP-discovered LE component – an instance of the activation-addition family (Turner et al., 2024; Rimsky et al., 2024) – and a *trained* 1*D subspace alignment* at a single LE head via Boundless DAS (Wu et al., 2023). The first tests whether an estimated linear direction is causally sufficient to move the judgment across tasks; the second tests whether a learned orthogonal basis can isolate the judgment variable inside one dimension of a single head. We report which protocol underlies each result explicitly, as the two make different claims and admit different controls.

Protocol 1: directional mean-difference steering. 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 $m = \text{sgn}(\text{EV}_{\text{clean}} - \text{EV}_{\text{corr}})$ 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.

Protocol 2: Boundless DAS alignment. At a single PEAP-identified LE head (L45H3 on Gemma-3-12B) we train an orthogonal rotation $\mathbf{R} \in \mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$ of the head's final-token activation with the Interchange Intervention Training (IIT) objective (Wu et al., 2023): the first rotated dimension of the corrupted run's activation is replaced by the clean run's value, and $\mathbf{R}$ is optimized so that the patched run reproduces the clean run's own predicted rating token (cross-entropy on the rating-token logits; the supervision signal is the model's own clean prediction, never a human label). We train one rotation per (site, task) on 100 minimal pairs. At evaluation, the interchange is scaled by α along the aligned dimension, $r \leftarrow r + \alpha\,(r_{\text{clean}} - r_{\text{corr}})$, yielding a dose-response in a learned 1D subspace. The trained rotation underlies the worked-example dose-response in Figure 2 (right), the random-rotation control below, and the zero-shot BDAS-1D judge readout (M4) in Appendix F.

E.2 RESULTS

Finding: The Latent Evaluator's judgment is encoded in a 1D steerable subspace (Protocol 1). 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 $\rightarrow$ MNLI, CoLA $\rightarrow$ STS-B). Qwen2.5-7B – the smallest model – matches Qwen2.5-14B and Gemma-3-27B in 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

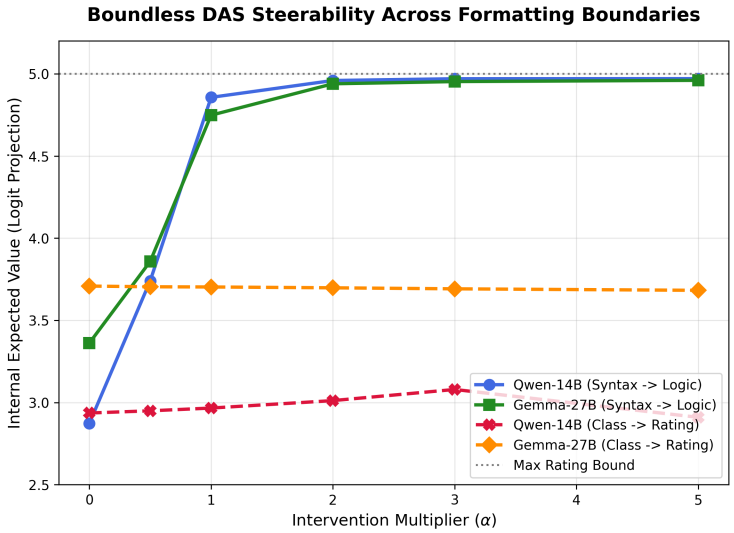


Figure 7: Cross-domain causal steering of the Latent Evaluator. By extracting a 1-dimensional directional steering vector at the LE components on a source domain (e.g., CoLA) and injecting it into the corresponding hooks of a distinct target domain (e.g., MNLI), we control the model's final output. The 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.

Table 3: Cross-task subspace steering at $\alpha = 2.0$ via the directional mean-difference vector (Protocol 1) at the PEAP-discovered Latent Evaluator components (§E). Steering succeeds across syntactically and semantically distinct source/target domains (boldfaced) but fails between distinct output formats (binary Classification → ordinal Rating).

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

monotonic and smooth in α (Figure 7, Figure 8), consistent with a 1D linear encoding rather than a nonlinear or multi-dimensional one.

Finding: Steering fails precisely at the cross-format boundary (Protocol 1). 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 → Rating" rows). This deliberate negative control reinforces the Latent Evaluator / Task Formatter split: the mean-difference vector carries the abstract judgment direction but cannot re-route a categorical output into an ordinal one, because that mapping is implemented by the non-linear terminal Task Formatter. This 1D-geometric finding is independently corroborated by the Format Transfer Injection experiment in

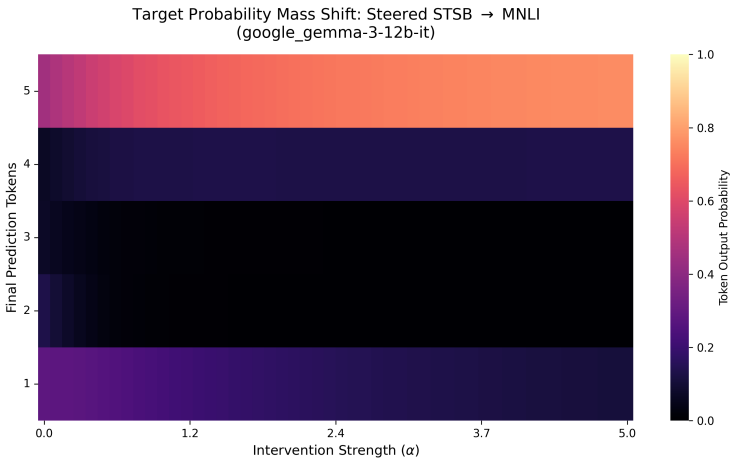


Figure 8: Steering probability heatmap for STS-B → MNLI across intervention strength (x-axis) and predicted rating tokens (1–5) (y-axis) for Gemma-3-12B. As α increases, probability mass shifts monotonically from lower ratings toward 5, demonstrating smooth, continuous control over the judgment output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

§5.1, which patches full activation patterns with no averaging into a single vector and reaches the same conclusion.

Finding: Steering extends to open-ended tasks on all five models (Protocol 1). 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 → Yelp reaches 4.87 ± 0.04. Qwen2.5-14B and Gemma-3-27B show the same pattern (Qwen2.5-14B CoLA → RewardBench: 4.98 ± 0.01; Gemma-3-27B CoLA → RewardBench: 4.92 ± 0.19). Gemma-3-12B's MNLI → RewardBench steering saturates and is bidirectional (EV 2.32 at $\alpha = 0$, 5.00 at $\alpha = 3$, down to 1.02 at $\alpha = -3$), and the CoLA → RewardBench cell shows partial monotone transfer (EV $2.33 \rightarrow 2.89$ from $\alpha = 0$ to $\alpha = 5$). Steering vectors sourced from MNLI into either open-ended target are substantially weaker on the modular models (e.g., Qwen2.5-14B MNLI → RewardBench: 3.15 ± 0.88), mirroring 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 on all five models – even where the blanket FTI intervention fails to flip the final argmax, as on Gemma-3-27B's open-ended tasks and the entangled Gemma-3-12B – strengthening the targeted-steering conclusion in §5.2.

Finding: Norm-matched random directions degrade judgment; only the mean-difference direction steers it (Protocol 1). A matched control for the mean-difference protocol asks whether a perturbation of the same magnitude at the same sites would move the output on its own. For each of ten seeds we replace every per-site steering vector $\bar{v}_{\ell,p,h}$ with a random direction of identical norm and re-evaluate at $\alpha = 2.0$ on Gemma-3-12B ($N = 100$ target pairs per cell, so baselines differ slightly from the full-N cells in Table 3). On MNLI → STS-B, the real vectors lift mean target EV from 3.44 to 4.61, while all ten random sets push it *down*, to between 1.80 and 2.21. On CoLA → MNLI, the real vectors lift EV from 3.48 to 3.82, and all ten random sets again land below baseline (1.90–2.80). Matched-norm random perturbations at the LE sites therefore degrade the judgment computation instead of steering it: EV falls in 20 of 20 cross-domain random runs, and no random sample approaches the real vectors' positive shift. The lift in Table 3 is thus attributable to the direction of $\bar{v}$, and no perturbation of equal size reproduces it. On the cross-format STS-B Binary → Rating cell, where Protocol 1 fails in Table 3, the real vectors behave like the random ones (2.17 vs. 2.92 ± 0.48 mean EV), consistent with the absent cross-format transfer.

Finding: Random-rotation control shows the trained BDAS rotation is directionally specific (Protocol 2). A skeptical reading of the alignment result is that any sufficiently large perturbation in activation space would shift the output, and the trained rotation is therefore not specifically aligned

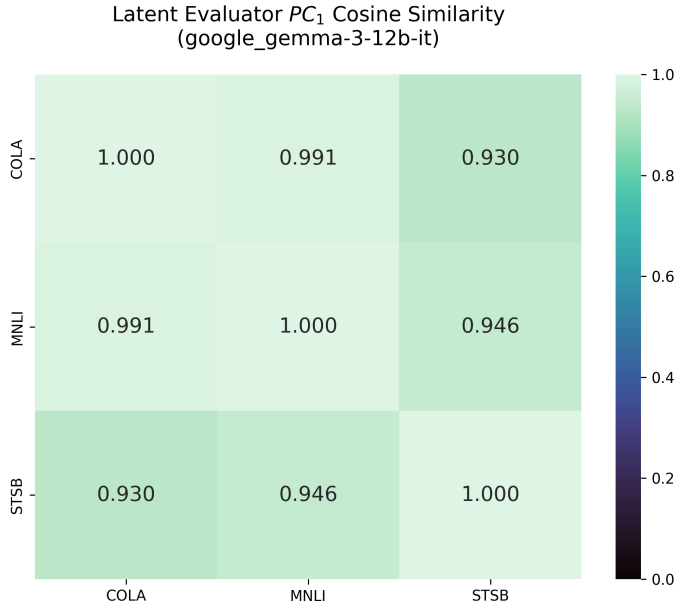


Figure 9: Pairwise cosine similarity between the PC_1 direction of the Latent Evaluator's clean/corrupt activation difference matrix across CoLA, MNLI, and STS-B (Gemma-3-12B).

to a judgment direction. We rule this out via a Haar-uniform random-rotation control on Gemma-3-12B: at $\alpha = 2.0$, the α-scaled interchange along the trained rotation's first dimension at L45H3 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 shift 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 perturbation. The absolute effect sizes here are naturally smaller than the Protocol 1 lifts in Table 3: the interchange perturbs one learned dimension of a single head, whereas the mean-difference protocol adds vectors at every LE component. On the cross-format STS-B Binary $\rightarrow$ Rating pair where Protocol 1 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 steering succeeds (cross-domain), the trained rotation is $\sim 50\times$ more effective than random. Where steering fails (cross-format), real and random rotations alike are inert, so the failure reflects a missing steerable direction; the same intervention strength succeeds wherever the direction exists.

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.

F 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 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

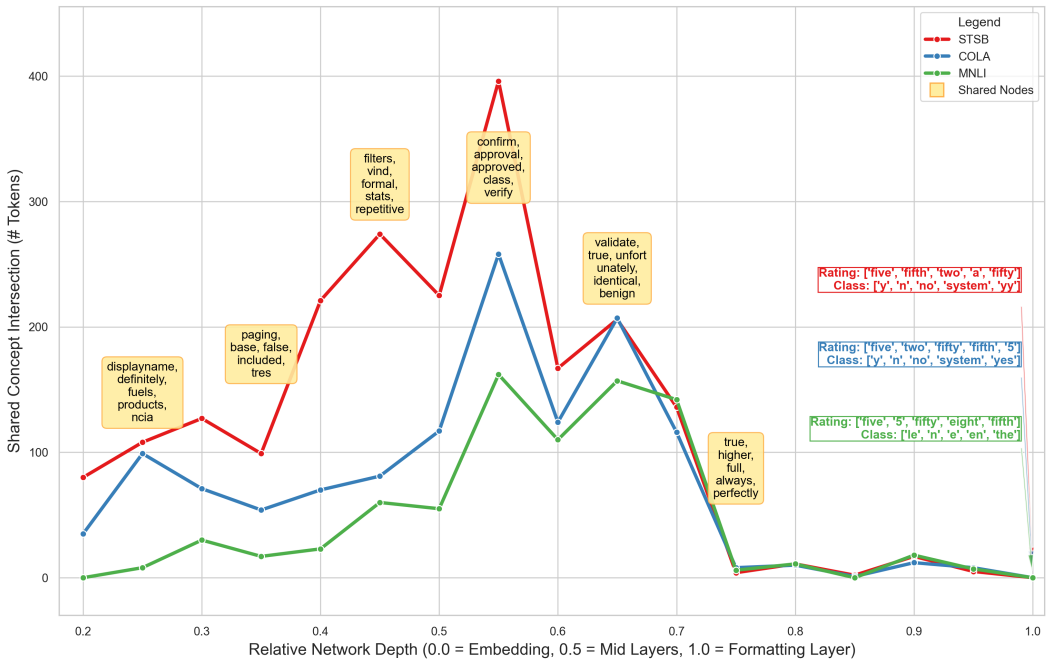


Figure 10: Timeline of the geometric token intersection overlap between ordinal rating (1-5) and categorical classification models. Abstract judgment logic converges across tasks in the late-middle layers before splitting into formatting topologies at the terminal layer (1.0).

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 F.

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

$\mathbf{R}$ trained via the Boundless DAS alignment (Protocol 2, App. E) 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 E.

Three regimes emerge from Table 4. *(i)* On STS-B, the supervised residual probe is the strongest signal on four of five models and BDAS-1D tracks it within a few percentage points without supervised training (mean $\Delta = 4.7$ pp), beating argmax on Llama-3.1-8B and both Gemma-3 models. *(ii)* On RewardBench, BDAS-1D matches or exceeds the supervised probe in 3 of 5 models (Gemma-

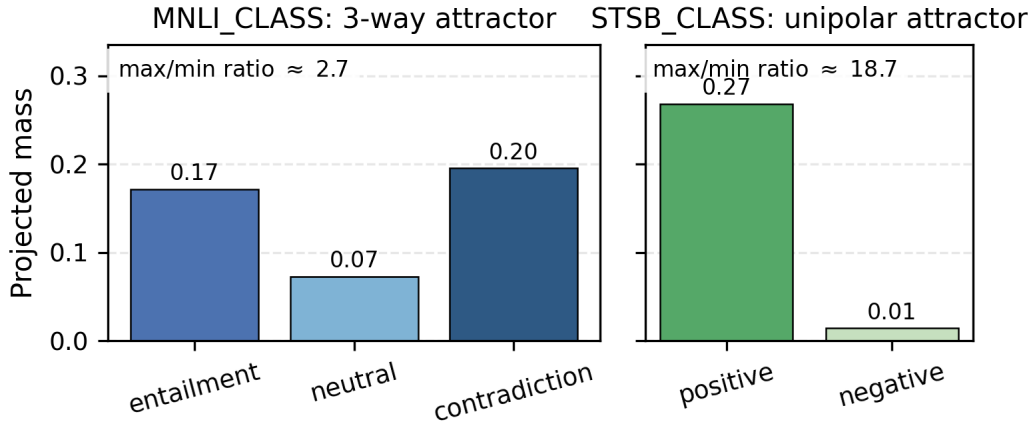


Figure 11: Late-layer Task Formatter attractor geometry on Gemma-3-27B (Logit Lens; Appendix L). MNLI's 3-class formatter splits mass roughly evenly across three target tokens (max/min ratio $\approx$ 2.7); STS-B's binary formatter concentrates mass on a single positive token (ratio $\approx$ 19). The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

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 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.

G Prompt Design

This section provides the exact string templates used to query the evaluated models, reproduced verbatim from `PROMPT_TEMPLATES` in `utils.py` of the code release. 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. Clean and corrupted prompts are two different instances with opposed ground truth under the *same* template (Appendix H): the content is the only differential between them, so PEAP attribution is, by construction, attribution to content evaluation.

1. CoLA
   Sentence: "{`sentence`}*"*
   On a scale of 1 to 5, how grammatically acceptable is this sentence?
   Rating:
2. MultiNLI
   Premise: "{`premise`}*"*
   Hypothesis: "{`hypothesis`}*"*
   On a scale of 1 to 5, how logically consistent is the hypothesis?
   Rating:
3. STS-B
   Sentence 1: "{`s1`}*"*
   Sentence 2: "{`s2`}*"*
   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:** *Given the premise: "{premise}" and the hypothesis: "{hypothesis}" 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:*

Knowledge-probe templates (Appendix U):

1. **StrategyQA:** *Question: "{question}" Answer:*
2. **CREAK:** *Claim: "{claim}" Is this claim true or false? 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.

H 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). PEAP's element-wise gradient computations require the clean and corrupted prompts within a pair to tokenize to the exact same length.

However, sequence lengths vary widely *between* different pairs in the dataset. To aggregate edge scores across the dataset into a generalized circuit, we right-align sequences and index positions from the end. The evaluation token (e.g., `Rating:`) is thus anchored at position -1 for all inputs, so the causal graphs superimpose regardless of the prompt length.

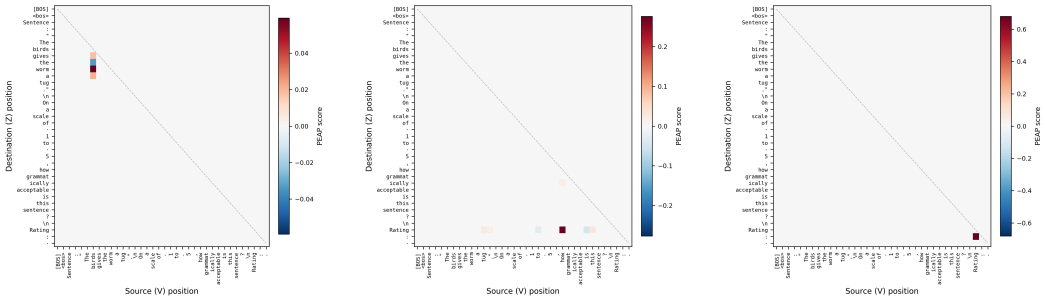
Per-task selection rules. Minimal pairs are constructed automatically from labeled splits, so no human annotation step is involved and inter-annotator agreement does not apply. Per task:

- 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), ≤ 500 (MNLI; we cap at 500), 189 (STS-B), 150–200 (RewardBench), and 145–200 (Yelp).

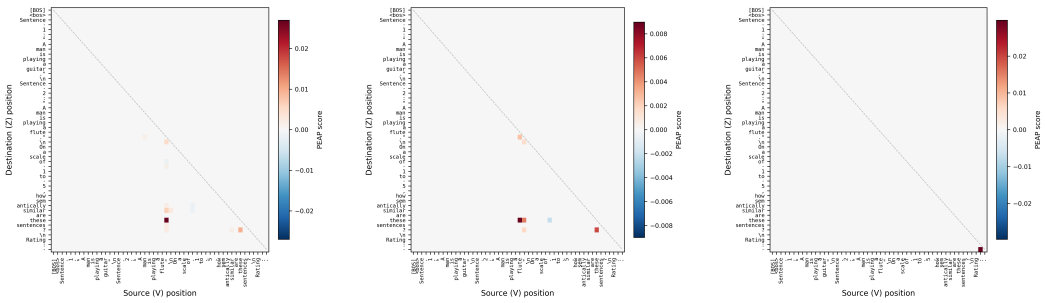
Backward-pass tracing budget. Dense backward-pass tracing has quadratic attention overhead in sequence length and is the binding cost for end-to-end attribution at the architectures we consider: mapping the full CoLA judgment graph in Gemma-3-12B requires evaluating approximately 1.46 million candidate edges, and Gemma-3-27B contains 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).

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.

I 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 tasks: STS-B classification is the most robust, while MNLI judgment is extremely fragile, with accuracy often dropping after ablating only the single top-ranked edge. Model scale also influences robustnes: smaller models (e.g., Qwen2.5-7B) have notably less robust judge circuits than larger models (e.g., Gemma-3-27B). All evaluation tasks demonstrate a characteristic phase transition, where accuracy remains relatively stable up to a critical edge-ablation threshold, beyond which performance collapses completely. Crucially, classification subtasks are consistently much more robust than their rating counterparts. This points to redundant processing pathways in the classification routers, whereas rating circuits compress into a few 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.

J Full Format Transfer Injection Results

Table 5 reports the complete FTI panel behind the flip rates in Table 2: per-cell base and patched probability of the positive target token, the flip rate, and its Wilson 95% confidence interval.

Table 5: FTI probability shifts on all five tasks with Wilson 95% CIs on the flip rate. 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}). Cells marked * ($N \leq 26$) are underpowered. Per-cell discussion in §5.1.

Model	Dataset	Base Prob.	Patched Prob.	Flip Rate	Wls. 95% CI
Qwen2.5-7B	CoLA ($N = 26$)*	17.4% ± 14.9%	93.5% ± 3.9%	100.0%	[87.1, 100]
	MNLI ($N = 324$)	3.4% ± 6.3%	88.0% ± 10.5%	99.1%	[97.3, 99.7]
	STS-B ($N = 134$)	1.6% ± 2.2%	85.6% ± 9.6%	100.0%	[97.2, 100]
	Yelp ($N = 437$)	0.1% ± 0.4%	59.5% ± 38.5%	66.4%	[61.8, 70.6]
	RewardBench ($N = 84$)	1.6% ± 5.1%	96.9% ± 10.1%	**100.0%**	[95.6, 100]
Llama-3.1-8B	CoLA ($N = 8$)*	28.6% ± 7.6%	41.7% ± 13.9%	25.0%	[7.1, 59.1]
	MNLI ($N = 120$)	10.4% ± 12.6%	13.1% ± 17.0%	4.2%	[1.8, 9.4]
	Yelp ($N = 94$)	7.1% ± 7.0%	11.1% ± 5.4%	0.0%	[0.0, 3.9]
	RewardBench ($N = 9$)*	26.1% ± 18.3%	39.6% ± 20.8%	22.2%	[6.3, 54.7]
Gemma-3-12B	CoLA ($N = 47$)	6.1% ± 11.8%	17.6% ± 26.8%	14.9%	[7.4, 27.7]
	MNLI ($N = 402$)	6.3% ± 11.0%	11.8% ± 15.9%	3.7%	[2.3, 6.1]
	STS-B ($N = 44$)	15.1% ± 13.6%	53.5% ± 33.1%	50.0%	[35.8, 64.2]
	Yelp ($N = 92$)	0.0% ± 0.1%	0.0% ± 0.1%	0.0%	[0.0, 4.0]
	RewardBench ($N = 26$)*	7.2% ± 12.4%	10.7% ± 17.3%	7.7%	[2.1, 24.1]
Qwen2.5-14B	CoLA ($N = 12$)*	24.3% ± 14.1%	69.0% ± 18.7%	91.7%	[64.6, 98.5]
	MNLI ($N = 125$)	4.2% ± 9.4%	4.9% ± 14.2%	4.0%	[1.7, 9.0]
	STS-B ($N = 156$)	0.0% ± 0.5%	16.0% ± 16.0%	5.1%	[2.6, 9.8]
	Yelp ($N = 189$)	0.4% ± 0.8%	41.0% ± 16.9%	31.2%	[25.0, 38.1]
	RewardBench ($N = 51$)	1.5% ± 5.0%	43.4% ± 36.4%	41.2%	[28.8, 54.8]
Gemma-3-27B	CoLA ($N = 54$)	8.6% ± 10.6%	9.9% ± 11.2%	0.0%	[0.0, 6.6]
	MNLI ($N = 151$)	8.0% ± 11.9%	8.0% ± 11.9%	0.7%	[0.1, 3.7]
	STS-B ($N = 31$)	30.5% ± 7.3%	78.1% ± 13.7%	**96.8%**	[83.8, 99.4]
	Yelp ($N = 189$)	1.3% ± 3.2%	3.2% ± 5.9%	1.1%	[0.3, 3.8]
	RewardBench ($N = 19$)*	20.8% ± 18.1%	22.1% ± 20.7%	5.3%	[0.9, 24.6]

K Reverse Format Transfer Injection

Reverse FTI runs the transfer of §5 in the opposite direction: we capture the Latent Evaluator activations of a confidently positive *classification* prompt (source probability of the positive label ≥ 0.8) and inject them into the same model running the matched *rating* prompt, restricted to instances whose baseline rating argmax is below 4. If the rating formatter accepted a blanket categorical judgment the way exposed classification formatters accept a rating judgment (Table 5), the injection should lift the predicted rating to 4 or 5. The attractor account of §5.1 predicts the opposite: with five competing rating tokens, the injected mass should spread without settling on a single one.

Table 6 matches this prediction on most cells: the expected rating value moves upward on a majority of pairs in 17 of 24 cells, but the argmax rating flips to 4 or 5 on fewer than 7% of pairs everywhere except two cells. The exceptions are Llama-3.1-8B on RewardBench and Gemma-3-12B on STS-B, where the upward shift is large enough to cross into the adjacent rating on a substantial share of pairs. Three deviations are worth stating plainly. First, Qwen2.5-7B on CoLA moves ratings *down* (ΔEV -0.64, with only 2% of pairs moving up), so on this cell the blanket classification pattern actively conflicts with the rating computation instead of merely fragmenting. Second, Qwen2.5-14B shows small negative shifts on four of five tasks. Third, Gemma-3-27B is essentially inert in every cell ($|\Delta\text{EV}| \leq 0.02$), consistent with the insulation this model already shows under forward FTI (Table 5). The Gemma-3-27B RewardBench cell is missing because the classification-format PEAP trace for that pair has not been run.

Table 6: Reverse FTI: injecting a confidently positive classification Latent Evaluator into a rating prompt whose baseline argmax rating is below 4. Flip rate is the fraction of pairs whose patched argmax rating reaches 4 or 5; ΔEV>0 is the fraction of pairs whose expected rating value moves upward. * marks underpowered cells ($N \leq 21$).

Model	Dataset	Base → Patched EV	ΔEV $\pm$ SD	Flip Rate	ΔEV>0
Qwen2.5-7B	CoLA ($N = 114$)	2.29 → 1.65	-0.64 ± 0.34	0.9%	1.8%
	MNLI ($N = 21$)*	1.07 → 1.05	-0.02 ± 0.16	0.0%	66.7%
	STS-B ($N = 174$)	1.59 → 1.78	$+0.19 \pm 0.30$	1.7%	69.5%
	Yelp ($N = 483$)	1.06 → 1.08	$+0.01 \pm 0.08$	0.0%	61.7%
	RewardBench ($N = 250$)	1.57 → 1.84	$+0.26 \pm 0.48$	6.4%	79.2%
Llama-3.1-8B	CoLA ($N = 50$)	2.35 → 2.47	$+0.12 \pm 0.13$	2.0%	86.0%
	MNLI ($N = 7$)*	2.31 → 2.54	$+0.23 \pm 0.36$	0.0%	85.7%
	STS-B ($N = 177$)	2.19 → 2.20	$+0.01 \pm 0.51$	2.8%	41.8%
	Yelp ($N = 96$)	1.17 → 1.24	$+0.08 \pm 0.09$	0.0%	91.7%
	RewardBench ($N = 43$)	2.26 → 3.33	$+1.07 \pm 0.50$	**58.1%**	100.0%
Gemma-3-12B	CoLA ($N = 121$)	1.67 → 1.77	$+0.11 \pm 0.14$	0.0%	100.0%
	MNLI ($N = 65$)	1.51 → 1.43	-0.08 ± 0.16	3.1%	9.2%
	STS-B ($N = 177$)	2.42 → 2.92	$+0.50 \pm 0.37$	**37.9%**	96.6%
	Yelp ($N = 95$)	1.01 → 1.01	-0.00 ± 0.04	0.0%	92.6%
	RewardBench ($N = 64$)	1.51 → 1.80	$+0.29 \pm 0.63$	4.7%	96.9%
Qwen2.5-14B	CoLA ($N = 113$)	1.91 → 1.73	-0.18 ± 0.17	0.0%	8.0%
	MNLI ($N = 135$)	1.56 → 1.35	-0.21 ± 0.24	0.0%	5.9%
	STS-B ($N = 189$)	1.66 → 1.61	-0.06 ± 0.13	0.0%	23.8%
	Yelp ($N = 197$)	1.08 → 1.07	-0.01 ± 0.06	0.0%	52.8%
	RewardBench ($N = 103$)	2.07 → 2.15	$+0.08 \pm 0.26$	5.8%	69.9%
Gemma-3-27B	CoLA ($N = 125$)	1.62 → 1.62	$+0.00 \pm 0.02$	0.0%	56.0%
	MNLI ($N = 66$)	1.38 → 1.38	-0.00 ± 0.01	0.0%	53.0%
	STS-B ($N = 89$)	2.08 → 2.08	-0.00 ± 0.01	0.0%	64.0%
	Yelp ($N = 197$)	1.11 → 1.13	$+0.02 \pm 0.06$	0.0%	97.0%
	RewardBench	not run (RewardBench classification trace unavailable)			

L 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[4]: projecting the contrastive steering representations directly into the vocabulary space using the model's unembedding weights (W_U). This exposes the semantic content of the nodes without depending on black-box auto-interpretability.

To compare structure across architectures without an arbitrary threshold, we project the contrastive evaluator vectors onto the unembedding matrix and measure cosine similarity. Normalization removes noise from untrained tokens and reveals geometric alignment across the nodes. Figure 10 quantifies this alignment between ordinal (Rating) and categorical (Classification) evaluators across network depth.

The projections trace a shared geometry in the late-middle depth band ($0.50 – 0.85$). 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 best to binary or ordinal tasks, while multi-class judgment tasks partially bypass the shared trunk. Applying a strict cross-architecture intersection to discard tokenizer-specific artifacts leaves a set of abstract evaluation tokens shared across all tasks. The highest-probability tokens at the evaluator nodes are the same across architectures: {*confirm*, *verify*, *validate*, *identical*, *perfectly*}.

Just before the terminal output-formatting boundary (depth 1.0), however, this shared coherence collapses across networks. The rating tasks route their probability mass toward discrete number

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

Table 7: 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.

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

tokens (e.g., *five*, *5*, *1*), abandoning the abstraction entirely. The classification variants polarize into categorical literals (e.g., *false*, *true*, *contradiction*) at the same depth. This supports our Task Formatter hypothesis: the Latent Evaluator calculates a continuous judgment signal in the deep layers, and task-specific routers overwrite that geometry strictly only at the terminal formatting stage. 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.

M 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.

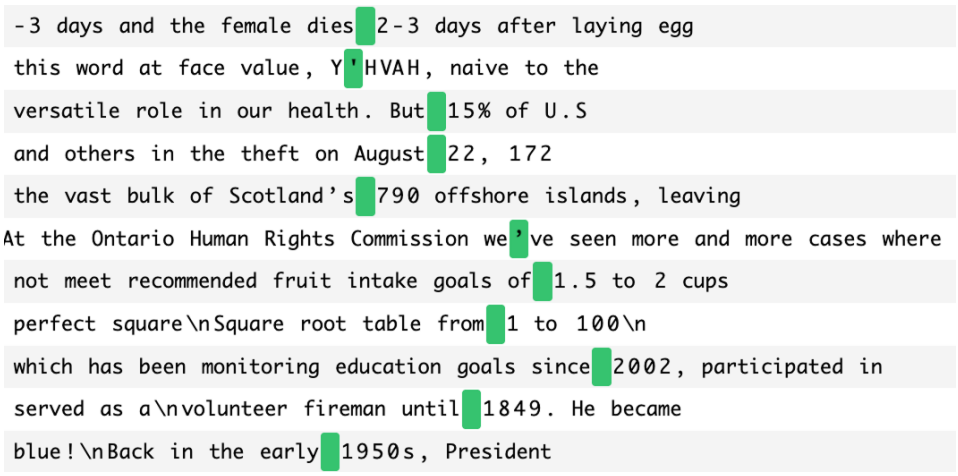


Figure 13: Activation pattern of transcoder feature F169 at layer 25 (position -2) in the CoLA circuit. The feature fires consistently on the leading digit of numerical expressions across diverse corpus contexts, representing the format-level numerical output constraint active across all prompts.

CoLA Circuit Head Analysis. Table 7 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 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 17 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). 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.

N 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 acti-

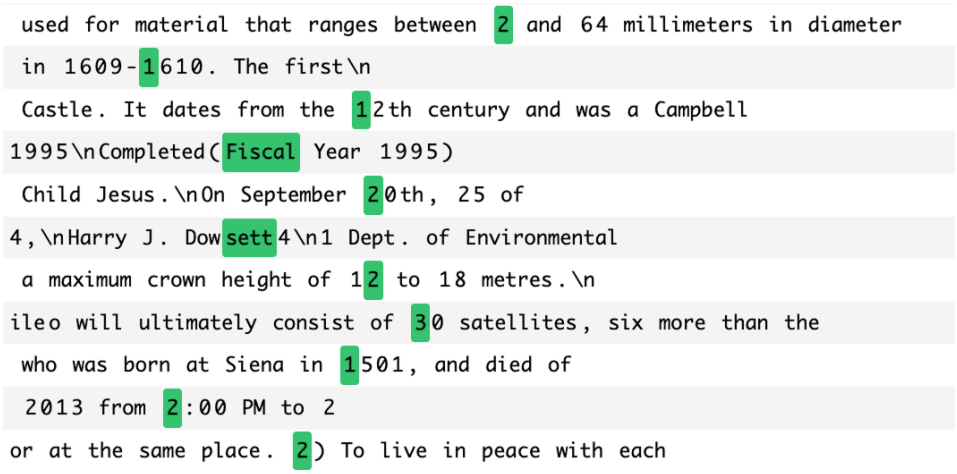


Figure 14: Transcoder feature F4 at layer 47 (position -1) in the CoLA circuit. It activates on tokens 1–3 in general corpus contexts and is more active on low grammaticality prompts, illustrating the value-encoding group: prompt-specific and judgment-sensitive, contrasting with format-level features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

vating 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 most 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. This pattern 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. F1201 in layer 25 and F255 in layer 26 are representative examples: they 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.

O 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.

P Split-Half Circuit Reliability

A recurring concern with circuit-level interpretability is whether circuits discovered on modest sample sizes reflect stable 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 $(\text{sender}, \text{receiver})$ 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.

Table 8: Split-half Edge IoU (%) at top-100, mean ± 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.

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

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 8 reports the resulting 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 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 D) 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 therefore reduces to a Gemma-3-12B-specific entanglement effect.

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.

Q Prompt-Paraphrase Robustness

A natural concern about any circuit traced on templated prompts is that the recovered structure reflects the exact wording of the template. To test this, we re-trace circuits with the identical PEAP

pipeline on two paraphrase levels of each rating prompt, on Qwen2.5-7B and Gemma-3-12B for CoLA, STS-B, and MNLI (the three structured tasks with the largest minimal-pair sets):

- **PARA1** rewords the instruction while keeping the field labels, e.g., CoLA's "On a scale of 1 to 5, how grammatically acceptable is this sentence?" becomes "Rate the grammatical acceptability of this sentence on a scale from 1 to 5."
- **PARA2** additionally renames the field labels and rephrases the question, e.g., "Sentence:" becomes "Input:" and MNLI's "Premise:"/"Hypothesis:" become "Statement A:"/"Statement B:", with the instruction "To what degree does Statement B follow from Statement A?".

Table 9 reports the Jaccard IoU (§3.2) between each paraphrased circuit and the original at three sparsity levels. Overlap under PARA1 is comparable to the within-task split-half ceiling of Appendix P (e.g., 74.9% edge IoU on Qwen2.5-7B MNLI at top-200, against a 65–70% split-half ceiling and a random null below 5%), so rewording the instruction changes the circuit about as much as resampling the data does. Renaming the field labels (PARA2) costs a further 10–25 IoU points; the weakest cell (Qwen2.5-7B STS-B at top-50) falls to roughly the cross-task overlap floor, indicating that the field labels themselves carry a share of the position-anchored structure at high sparsity. Node IoU stays above 45% in every cell, so the same components are recruited even where the specific edges shift.

Table 9: Circuit overlap between original and paraphrased rating prompts at three top-k thresholds. PARA1 rewords the instruction; PARA2 additionally renames the prompt's field labels. Reference points from Appendix P: within-task split-half edge IoU (the effective ceiling) is 65–80% on these cells, the random-edge null is 0.7–5%, and cross-task overlap (the floor for related judgment tasks) is 24–49%.

		Edge IoU / Node IoU (%)		
Model / Task	**Variant**	top-50	top-100	top-200
Qwen2.5-7B CoLA	PARA1	62.7 / 62.5	67.6 / 75.6	73.0 / 81.6
	PARA2	38.8 / 52.7	50.0 / 65.5	55.7 / 63.2
Qwen2.5-7B STS-B	PARA1	62.1 / 81.4	59.6 / 73.6	66.5 / 79.7
	PARA2	17.7 / 45.1	31.6 / 50.6	42.4 / 55.0
Qwen2.5-7B MNLI	PARA1	67.3 / 84.1	74.0 / 82.7	74.9 / 86.0
	PARA2	40.0 / 48.1	48.2 / 55.1	52.4 / 62.5
Gemma-3-12B CoLA	PARA1	49.2 / 54.8	56.2 / 63.4	55.0 / 74.3
	PARA2	31.0 / 38.5	34.1 / 56.1	42.4 / 58.3
Gemma-3-12B STS-B	PARA1	42.2 / 52.9	59.4 / 61.3	54.3 / 61.9
	PARA2	42.9 / 48.6	45.6 / 60.0	49.7 / 58.5
Gemma-3-12B MNLI	PARA1	54.2 / 50.9	53.6 / 57.7	53.5 / 71.6
	PARA2	39.4 / 55.0	43.1 / 61.5	44.4 / 57.5

R Pooled-Directional Faithfulness

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

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

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

S CROSS-METHOD VALIDATION VIA LRPEAP

PEAP belongs to the strongest circuit-localization family in the MIB benchmark (Mueller et al., 2025), and the position-aware extension further improves the faithfulness-per-edge trade-off over position-invariant formulations (Haklay et al., 2025). The faithfulness results (Appendix D) address metric fragility; this appendix addresses the complementary non-identifiability concern by comparing PEAP against a second attribution backbone. On the (Qwen2.5-7B, Gemma-3-12B) $\times$ 10-task panel, top-200 PEAP and LRPEAP edge sets share 42% mean Jaccard IoU on edges and 54% on components (permutation null $p_{99} \leq 5.8\%$); the Latent Evaluator subgraph $\mathcal{C}_{\text{LE}} = \mathcal{C}_{\text{rate}} \cap \mathcal{C}_{\text{class}}$ computed under each method recovers at 0.54 mean component IoU, peaking at 0.68 on RewardBench. 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.

S.1 METHODOLOGY

LRPEAP retains PEAP's position-aware edge attribution (Appendix B) 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 $m = \text{sgn}(\text{EV}_{\text{clean}} - \text{EV}_{\text{corr}})$ – 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.

S.2 RESULTS

Table 10: PEAP vs LRPEAP Jaccard IoU at $K{=}200$ (edge / component); null is the permutation p_{99}.

	Qwen2.5-7B		Gemma-3-12B	
Task	Edge	Comp.	Edge	Comp.
CoLA	0.41	0.52	0.28	0.47
CoLA_CLASS	0.44	0.60	0.20	0.42
MNLI	0.37	0.51	0.56	0.58
MNLI_CLASS	0.50	0.57	0.52	0.63
STS-B	0.45	0.58	0.43	0.51
STS-B_CLASS	0.44	0.60	0.38	0.58
Yelp	0.45	0.55	0.50	0.56
Yelp_CLASS	0.42	0.53	0.28	0.47
RewardBench	0.40	0.53	0.37	0.47
RewardBench_CLASS	0.37	0.48	0.60	0.69
Mean	**0.43**	**0.55**	**0.41**	**0.54**
Null p_{99}	0.031	–	0.058	–

Table 10 reports per-task PEAP↔LRPEAP IoU at $K{=}200$: mean edge IoU is 0.43 on Qwen2.5-7B and 0.41 on Gemma-3-12B against null p_{99} of 0.031 and 0.058, a 4–22$\times$ enrichment at $K{=}200$, with agreement above the null at every $k \in \{25, \ldots, 500\}$. Cross-method agreement is comparable on the two models and uniform across tasks (edge IoU 0.37–0.50 on Qwen2.5-7B).

Table 11: Cross-method Latent Evaluator IoU at $K=200$ between PEAP's $\mathcal{C}_{\text{LE}}$ and LRPEAP's $\mathcal{C}_{\text{LE}}$, each computed as $\mathcal{C}_{\text{rate}} \cap \mathcal{C}_{\text{class}}$.

	Qwen2.5-7B		Gemma-3-12B	
Task pair	Edge	Comp.	Edge	Comp.
CoLA pair	0.45	0.49	0.17	0.43
MNLI pair	0.46	0.54	0.42	0.62
STS-B pair	0.42	0.57	0.29	0.57
Yelp pair	0.42	0.60	0.20	0.44
RewardBench pair	0.31	0.49	0.42	0.68
Mean	**0.41**	**0.54**	**0.30**	**0.55**

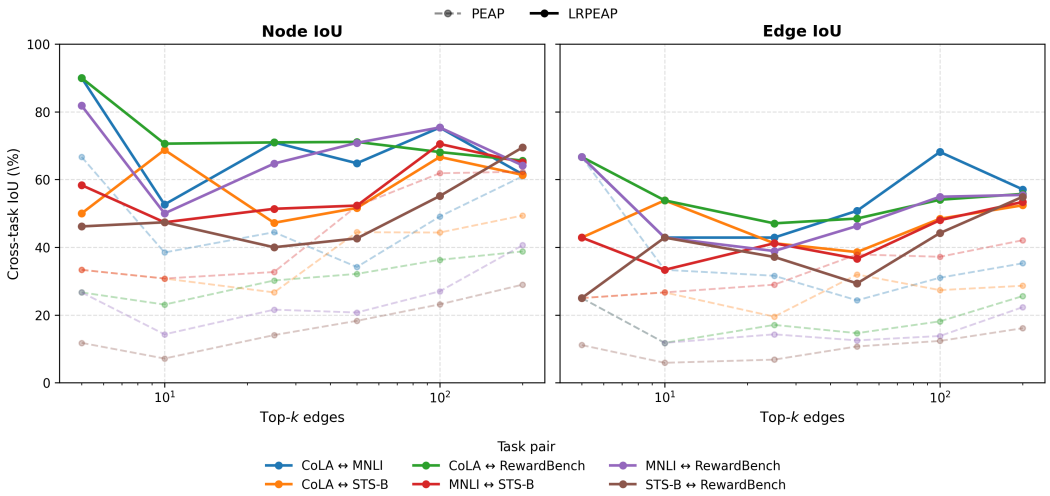


Figure 15: Cross-task structural overlap on Gemma-3-12B: LRPEAP (solid) overlaid on PEAP (dashed, faded) at matched task-pair color. Node IoU agrees on the structurally easy pairs (CoLA × MNLI, MNLI × STS-B); Edge IoU is consistently higher under LRPEAP, with both metrics diverging in LRPEAP's favor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

Restricting to the Latent Evaluator (Table 11), the LE subgraph is recovered with 0.36 edge / 0.54 component IoU on average, peaking at 0.68 on Gemma-3-12B RewardBench.

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 M) 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 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-

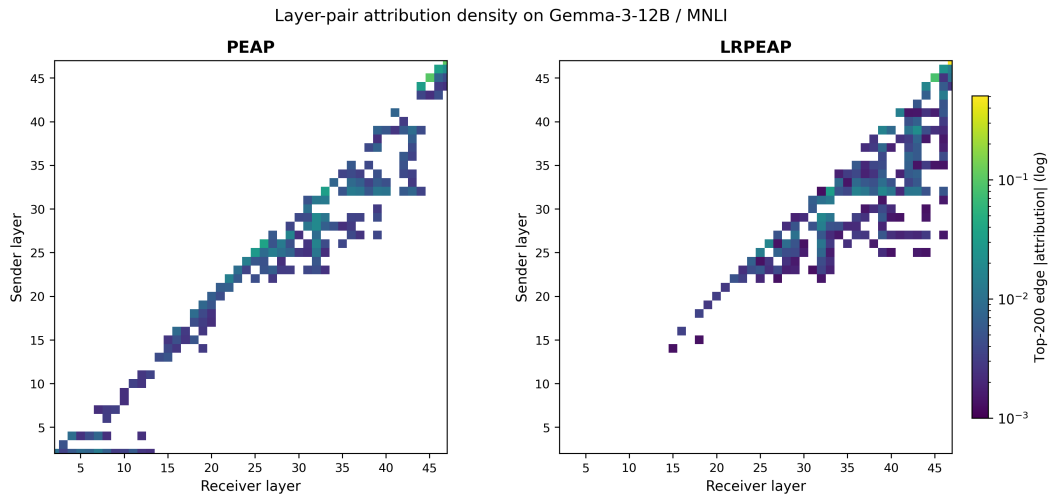


Figure 16: Layer-pair attribution density of the top-200 edges on Gemma-3-12B MNLI under PEAP (left) and LRPEAP (right). Both methods concentrate attribution in the same mid-to-late diagonal band, the LE region of §4.1. LRPEAP additionally suppresses early-layer attribution that PEAP picks up, possibly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

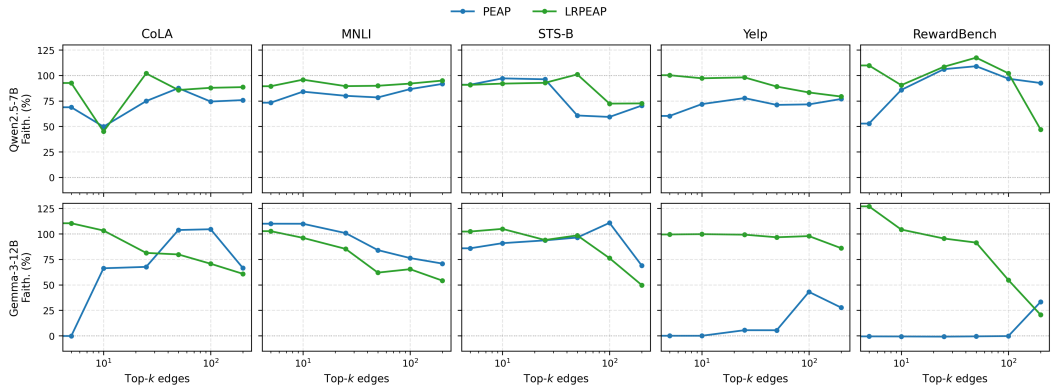


Figure 17: Sparse-circuit faithfulness with PEAP (blue) and LRPEAP (green) on Qwen2.5-7B and Gemma-3-12B across the five rating tasks.

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. There, LRP-rule relevance redistribution does not preserve the per-pair sign that PEAP's symmetric polarity correction (§3.1) handles by construction.

S.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 both Qwen2.5-7B and Gemma-3-12B 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 is an order of magnitude below the PEAP↔LRPEAP overlap at the edge level, but stays above chance throughout: 6.6–24.5× the permutation null across the four cells (Table 12, Figure 18). At the component level the gap narrows; on Gemma-3-12B ACDC agrees with both gradient methods at 0.28–0.31 IoU (2.2–3.0× null) against the gradient-gradient pairing's 0.37. The triple intersection at $K{=}200$ is small (6–10 edges), with the shared

Table 12: Three-way pairwise Jaccard IoU at $K{=}200$ between PEAP, LRPEAP, and ACDC on the shared 10-pair subset, with each cell's enrichment over its permutation p_{99} null in parentheses.

Model	Pair	Edge	Comp.
Qwen2.5-7B	PEAP ↔ LRPEAP	**0.31** (125×)	**0.55** (5.9×)
	PEAP ↔ ACDC	0.03 (11×)	0.16 (1.5×)
	LRPEAP ↔ ACDC	0.03 (13×)	0.14 (1.4×)
Gemma-3-12B	PEAP ↔ LRPEAP	**0.21** (85×)	**0.37** (3.8×)
	PEAP ↔ ACDC	0.02 (7.9×)	0.31 (2.8×)
	LRPEAP ↔ ACDC	0.05 (18×)	0.29 (2.9×)

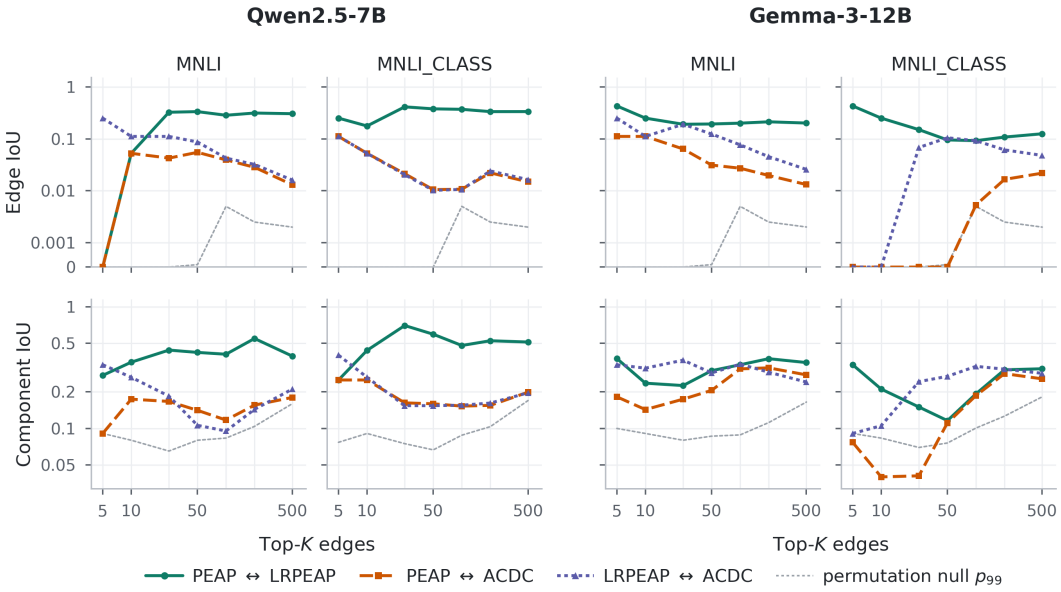


Figure 18: Pairwise Jaccard IoU as a function of K for PEAP, LRPEAP, and ACDC on the shared 10-pair subset, at the edge level (top) and the component level (bottom), for both models and both tasks. The dotted line is the permutation null p_{99}, which is exactly zero at small K. The edge-level axis is logarithmic above 0.001 and linear below, so exact zeros are drawn at zero.

components falling in the mid-to-late band that contains the LE (§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 score every edge from a single backward pass via the first-order gradient of the metric.

T 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 the discovered circuits overlap.

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

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

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 P.

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. P): 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. P), 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 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.

U Knowledge-Probe Controls, Recall Reference Circuits, and LE Threshold Stability

This appendix reports the full data behind the entanglement decomposition of §4.2: a heads-only ablation panel, a pure-recall cloze probe, factual-recall reference-circuit containment, and the Latent Evaluator's stability across the top-k threshold.

U.1 Heads-Only Ablation Panel

Table 13 isolates the contribution of the LE attention heads by ablating only the heads (leaving all LE MLPs intact), with a matched random-heads control. On Gemma-3-12B, the three LE heads alone account for the MMLU verification damage (14–18 pp across subsets) while StrategyQA and CREAK stay within noise of the control; on Qwen2.5-7B the single LE head is inert on every probe. Llama-3.1-8B, Qwen2.5-14B, and Gemma-3-27B have no attention heads in their merged LE, so their heads-only cells are vacuous by construction and are omitted.

U.2 Pure-Recall Cloze Probe (PopQA)

The verification probes of Table 1 contain an evaluative step (deciding which answer is more correct), so damage to them cannot by itself distinguish knowledge entanglement from the probes recruiting judgment machinery. We therefore run a declarative cloze completion on PopQA (e.g.,

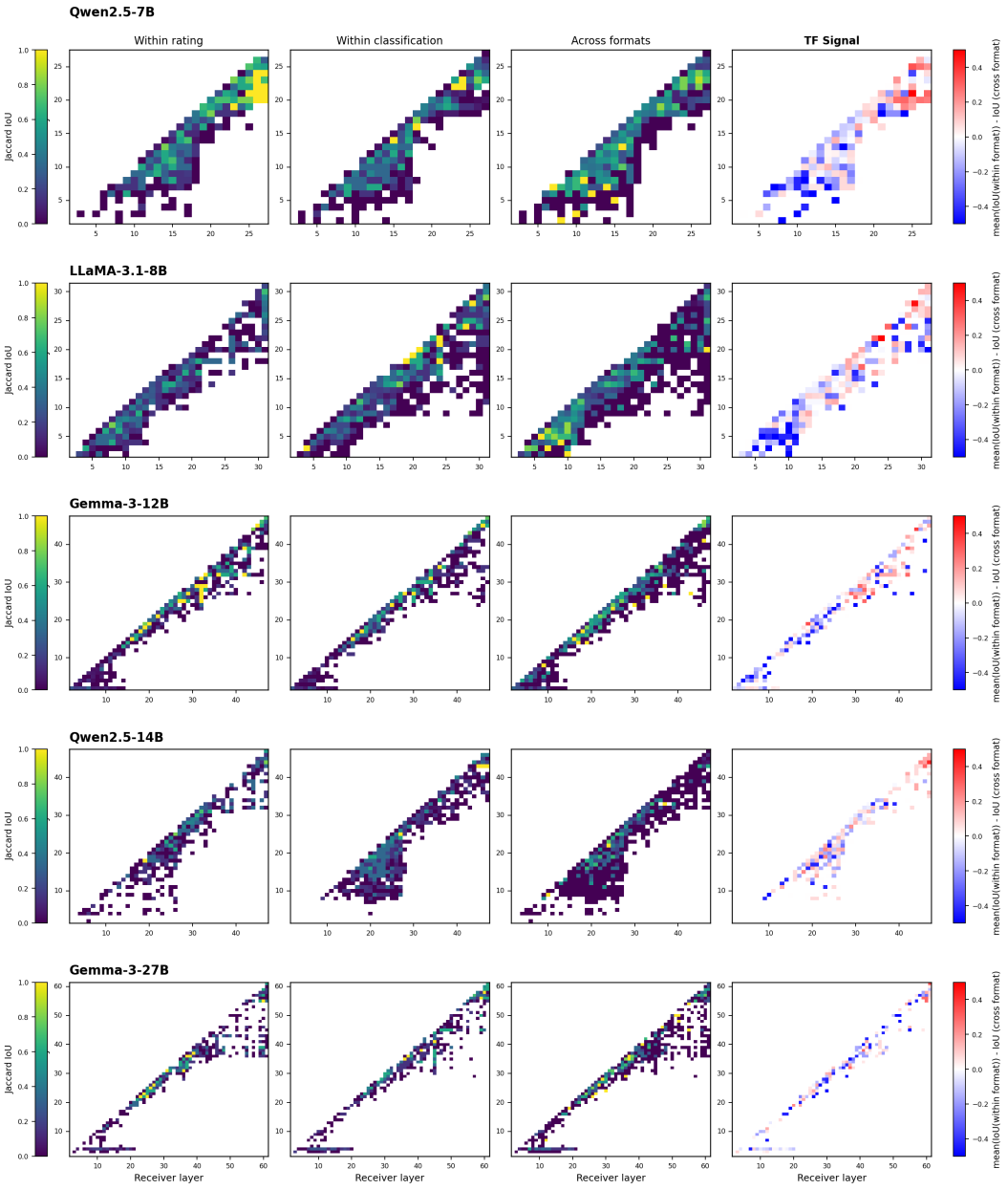


Figure 19: Per-model (sender, receiver) Edge IoU grids at top-200, one row per model. The first three columns show mean Jaccard IoU within rating pairs, within classification pairs, and across tasks; the fourth column shows the TF signal (App. T). Positive (red) cells in the TF signal panel localize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

"The capital of $\{X\}$ *is* ___"; no candidate answers, no verification step; LAMA-style first-token P@1; $N = 300$) under LE zero-ablation vs. the same size/type-matched random-ablation control (Table 14). The probe decomposes the Gemma-3-12B result of §4.2: full-LE ablation drops pure recall 13 pp below the control (MLP-carried recall entanglement, also present at 27B with 16 pp), while ablating only the three LE heads – which suffice to collapse MMLU (Table 13) – leaves pure recall untouched. On Qwen2.5-7B/14B and Llama-3.1-8B, LE ablation never exceeds the control: recall does not route through their LEs.

Table 13: Heads-only zero-ablation (%) with a size-matched random-heads control (3 trials). Bold marks cells where damage exceeds the control. Gemma-3-12B's verification damage in Table 1 is carried by its three LE heads; models whose LE contains no heads are omitted (vacuous).

Model	Probe	Base	Heads abl.	Random heads
Gemma-3-12B (3 LE heads)	Clinical DB	79.0	**63.0**	80.0 ± 0.8
	Abstract Alg.	47.0	**29.0**	48.1 ± 3.0
	Physics	52.0	**38.0**	51.7 ± 0.5
	StrategyQA	65.0	64.0	64.8 ± 0.2
	CREAK	80.5	78.0	80.8 ± 0.8
Qwen2.5-7B (1 LE head)	Clinical DB	78.0	77.0	77.3 ± 0.9
	Abstract Alg.	56.0	54.0	55.3 ± 0.9
	Physics	53.0	55.0	51.7 ± 1.2
	StrategyQA	66.5	66.0	65.2 ± 1.2
	CREAK	81.5	82.0	81.8 ± 1.0

Table 14: Declarative PopQA cloze accuracy (first-token P@1, %, $N = 300$) under LE zero-ablation vs. a size/type-matched random-ablation control (3 trials). Bold marks LE-ablation damage exceeding the control: an MLP-carried recall entanglement specific to the Gemma-3 family. Heads-only rows are omitted for the models whose LE contains no attention heads (vacuous).

Model	Ablated	Base	LE-abl.	Random ctrl.
Qwen2.5-7B	1H+9M	44.7	4.7	3.4 ± 1.3
Qwen2.5-7B	heads only (1H)	44.7	38.7	41.2 ± 0.6
Llama-3.1-8B	0H+8M	62.7	5.7	11.5 ± 9.2
Gemma-3-12B	3H+4M	60.3	**49.7**	62.9 ± 4.5
Gemma-3-12B	heads only (3H)	60.3	60.0	60.8 ± 0.4
Qwen2.5-14B	0H+2M	62.3	54.0	53.9 ± 16.9
Gemma-3-27B	0H+3M	59.3	**46.3**	62.3 ± 3.6

U.3 Factual-Recall Reference Circuits

As a mechanism-level criterion separating retrieval from judgment, we trace factual-recall circuits on CREAK (claim verification) and StrategyQA (implicit-reasoning QA) with the identical PEAP pipeline, at both $N = 100$ and $N = 200$ minimal pairs. These circuits double as the non-judgment external reference whose absence §A previously flagged. For each recall circuit we measure the fraction of the merged Latent Evaluator (the intersection of the CoLA/MNLI/STS-B per-task LEs) that it recruits (Table 15). Claim verification routes through the LE far more on the entangled Gemma-3-12B than on the modular Qwen2.5-7B at every threshold and sample size, a graded contrast consistent with the cloze-probe dissociation of Appendix U.2; the permutation null for size-matched random subsets is $\approx 5\%$ throughout. StrategyQA's containment is high on *both* models, consistent with implicit-reasoning QA itself recruiting evaluative machinery – which is exactly why the cloze probe, with no verification step, is the discriminating control.

U.4 LE Stability Across the Top-k Threshold

Because the Latent Evaluator is defined as an intersection of circuits pruned at a fixed top-k, we verify that its identity is not an artifact of the chosen threshold. Computing $\mathcal{C}_{\text{LE}}(k)$ for $k \in \{50, 100, 200, 500, 1000\}$ from the cached attribution scores, the LE is *strictly nested*: $\mathcal{C}_{\text{LE}}(k) \subseteq \mathcal{C}_{\text{LE}}(1000)$ with 100% containment at every k, for all of CoLA/MNLI/STS-B on both Gemma-3-12B and Qwen2.5-7B (Table 16). Varying k grows the intersection but never swaps out components, so the *identity* of the shared evaluator is threshold-stable (raw IoU across k differs only because set sizes differ), and the named shared-evaluator heads (L45H3, L46H12, L47H7) are present at every k from 50 to 1000 on all three tasks. Our operating point $k = 200$ is where faithfulness saturates for 21 of 25 cells (Figure 3); the top-1000 instance in Figure 2 is a visualization superset of the same nested hierarchy.

Table 15: Merged-LE containment (%): the share of the merged Latent Evaluator's components recruited by each factual-recall reference circuit, at two recall-circuit sample sizes (N pairs) and two pruning thresholds (top-k edges). The claim-verification contrast (bold vs. the Qwen column) is preserved at every threshold and sample size; permutation null $\approx 5\%$.

		Gemma-3-12B		**Qwen2.5-7B**	
Recall circuit	N	top-100	top-200	top-100	top-200
CREAK	100	82.4	**91.9**	36.0	48.7
	200	70.6	**78.4**	40.0	64.1
StrategyQA	100	52.9	75.7	76.0	82.1
	200	94.1	75.7	88.0	89.7

Table 16: Latent Evaluator size $|\mathcal{C}_{\mathrm{LE}}(k)|$ (components) across pruning thresholds. Every set is fully contained in the next larger one (100% nesting along each row), and the named shared-evaluator heads appear at every k on all Gemma-3-12B tasks.

Model	**Task**	k=50	k=100	k=200	k=500	k=1000
Gemma-3-12B	CoLA	28	54	77	124	177
	MNLI	34	46	70	107	151
	STS-B	39	61	88	137	201
Qwen2.5-7B	CoLA	23	43	77	143	222
	MNLI	23	35	55	122	209
	STS-B	24	45	82	140	227

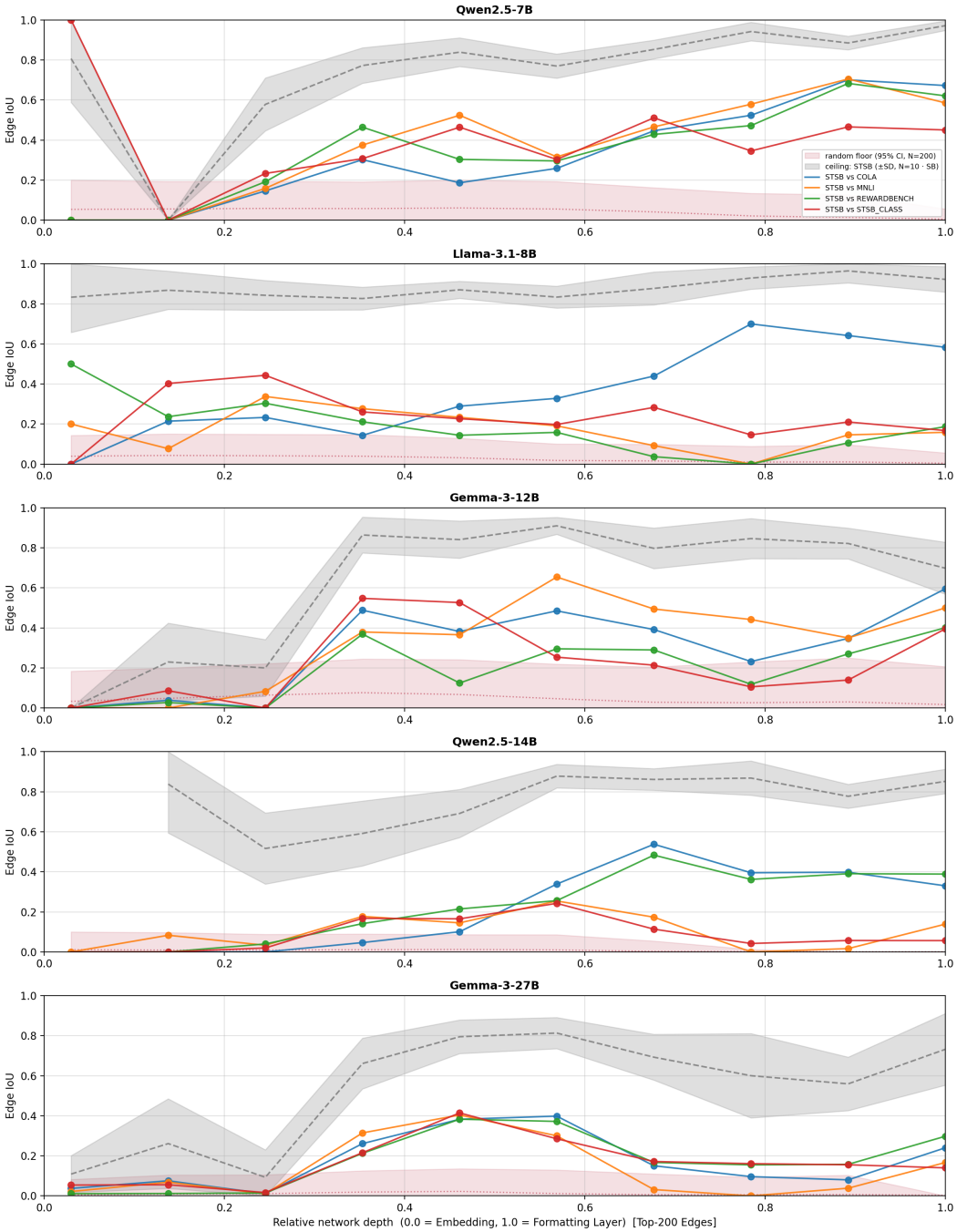
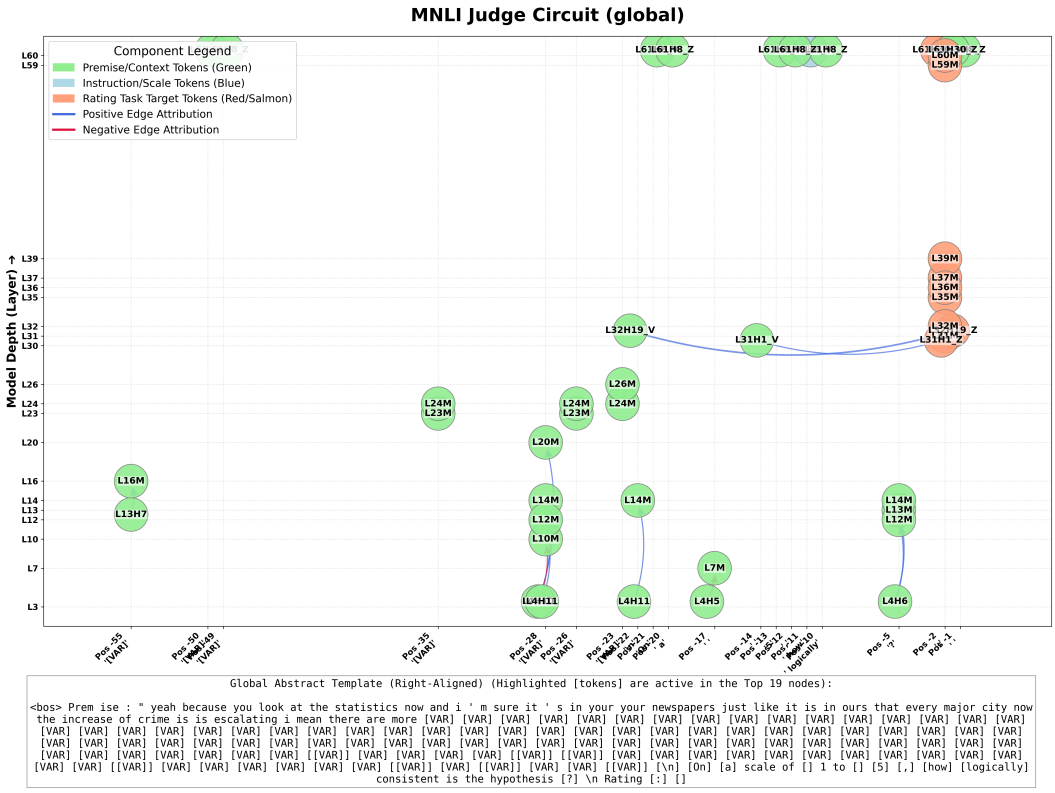


Figure 20: Per-layer Edge IoU at $K = 200$ between STS-B and every other task across the evaluated models, binned into 10 relative-depth bins. The lower band is the random-edge baseline (95% CI over 200 trials); the upper band is the within-task split-half ceiling. The ceiling is undefined for `Qwen2.5-14B`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


MNLI Judge Circuit (global)
Component Legend
Premise/Context Tokens (Green)
Instruction/Scale Tokens (Blue)
Rating Task Target Tokens (Red/Salmon)
Positive Edge Attribution
Negative Edge Attribution
Model Depth (Layer) →
Global Abstract Template (Right-Aligned) (Highlighted [tokens] are active in the Top 19 nodes):
<bos> Prem ise : " yeah because you look at the statistics now and i ' m sure it ' s in your your newspapers just like it is in ours that every major city now the increase of crime is is escalating i mean there are more [VAR] [VAR] [VAR]

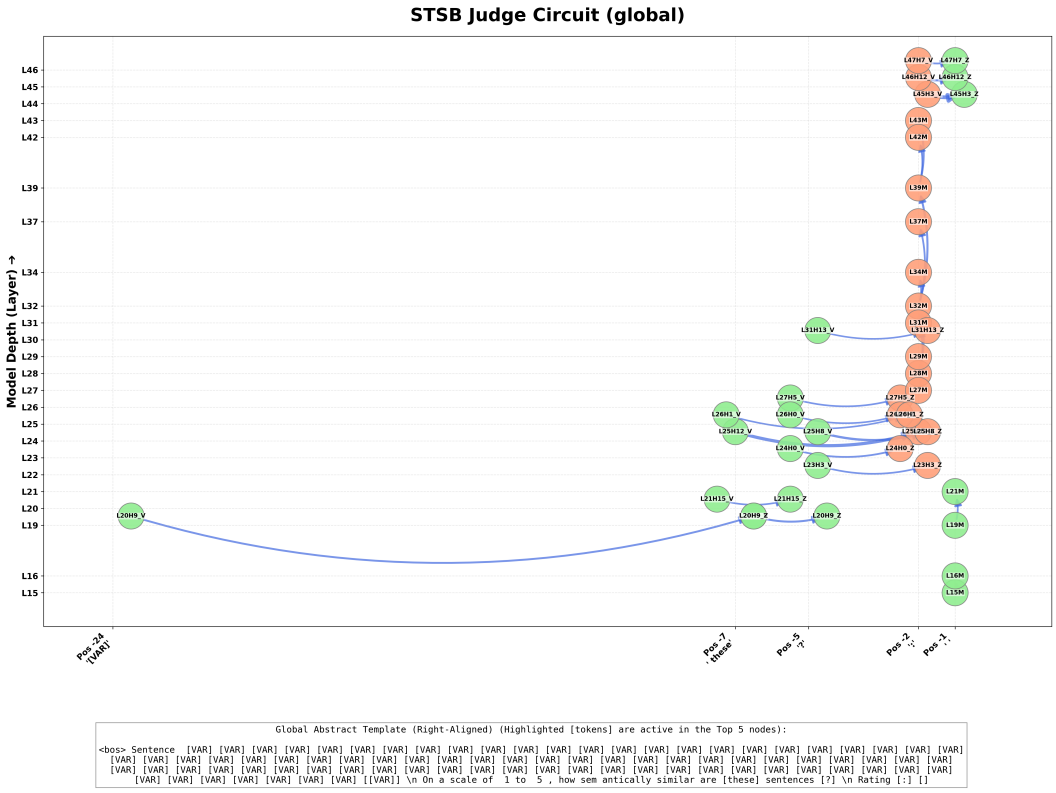
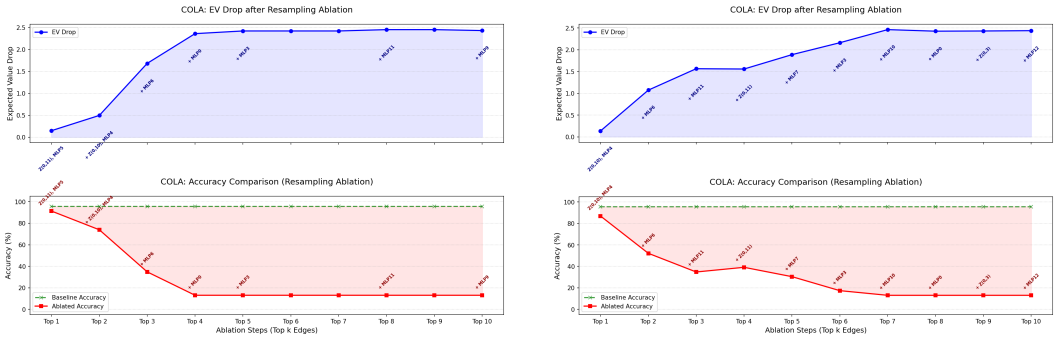

STSB Judge Circuit (global)
Model Depth (Layer) →
Global Abstract Template (Right-Aligned) (Highlighted [tokens] are active in the Top 5 nodes):
<bos> Sentence [VAR] [VAR] [VAR]

Table 17: 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.

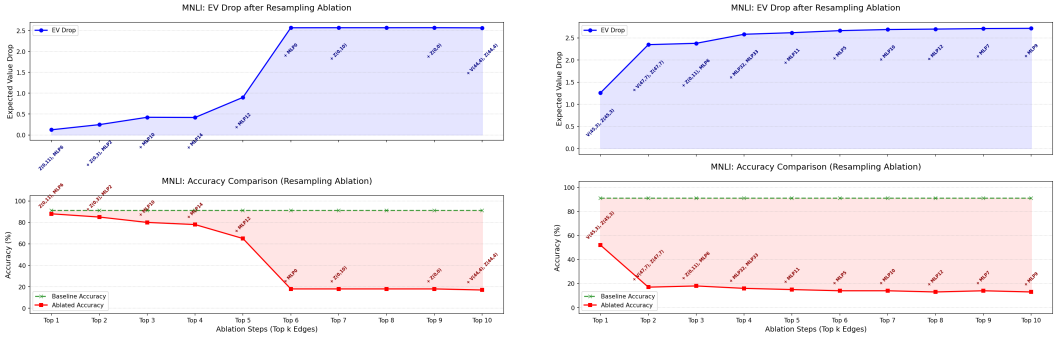
Head	Description
Shared Evaluators	
L20H9	Contrastive token detector; attends to semantically differentiating tokens in the second sentence and simultaneously distributes the resulting comparison signal into all three downstream pathways (shared evaluation, rating-specific, and classification-specific)
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



(a) Classification Ablation.

(b) Numerical Judgment Ablation.

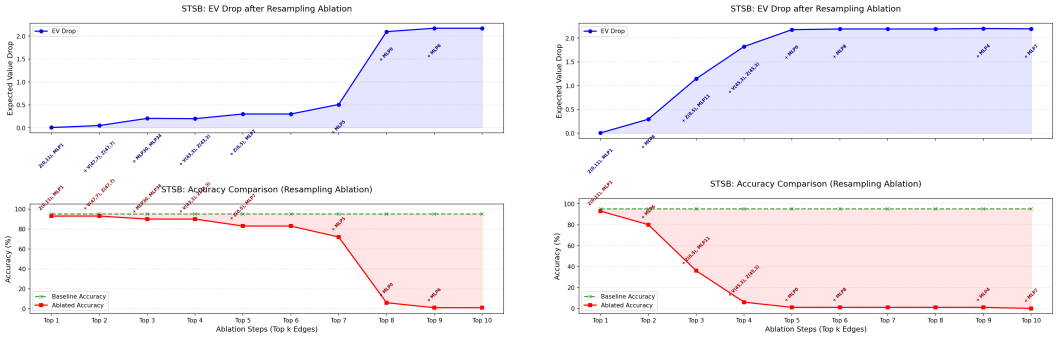
COLA dataset.



(c) Classification Ablation.

(d) Numerical Judgment Ablation.

MNLI dataset.

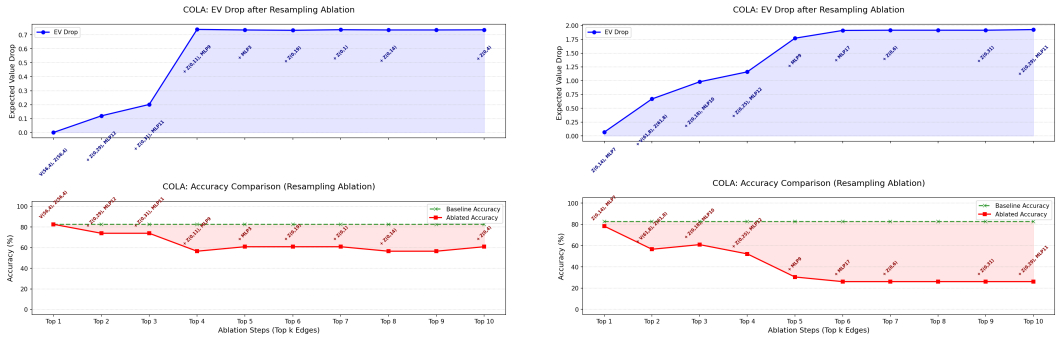


(e) Classification Ablation.

(f) Numerical Judgment Ablation.

STSB dataset.

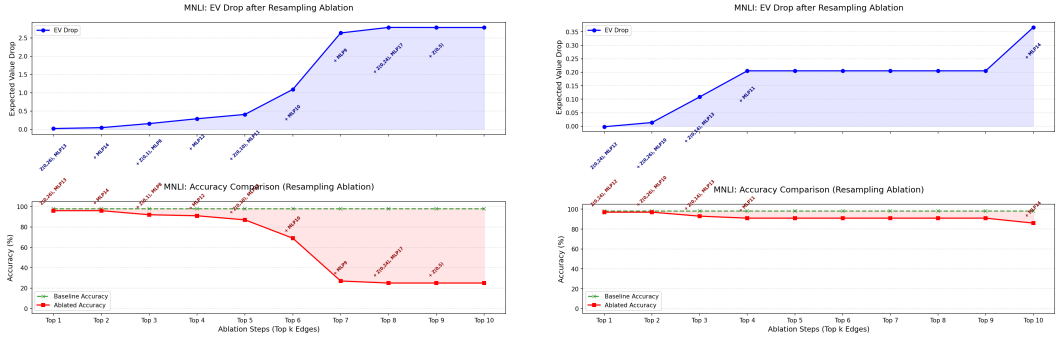
Figure 25: Ablation phase-transition study (Gemma-3-12B).



(a) Classification Ablation.

(b) Numerical Judgment Ablation.

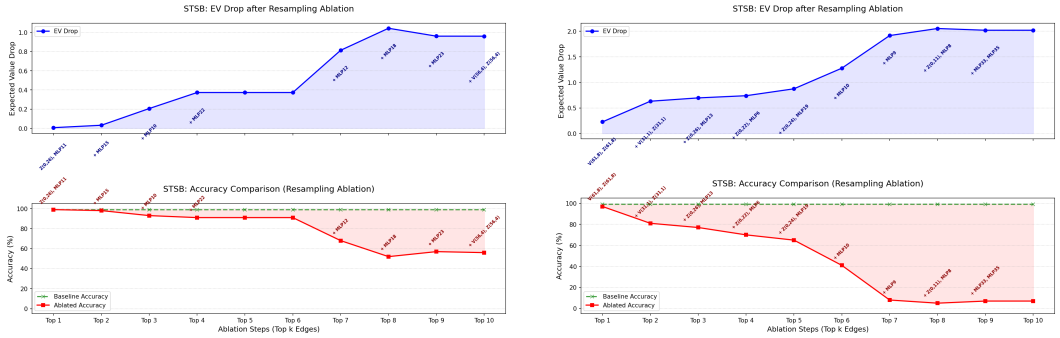
COLA dataset.



(c) Classification Ablation.

(d) Numerical Judgment Ablation.

MNLI dataset.

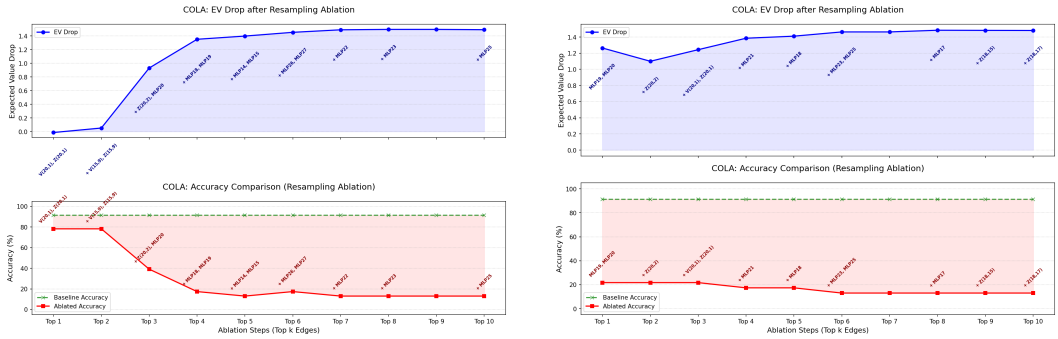


(e) Classification Ablation.

(f) Numerical Judgment Ablation.

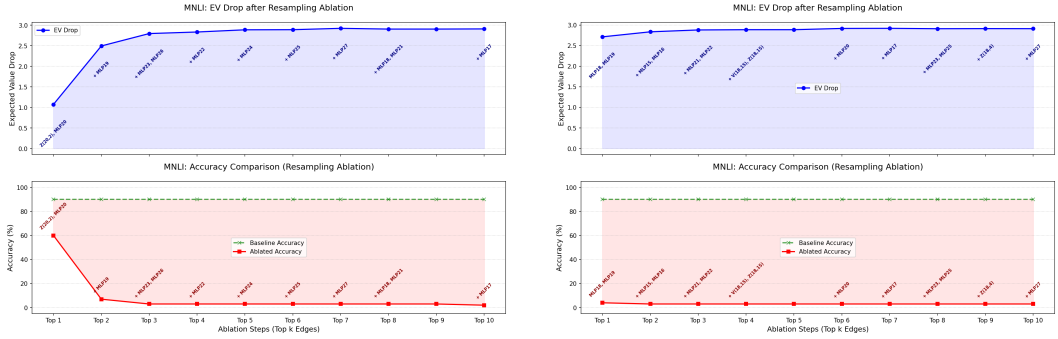
STSB dataset.

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



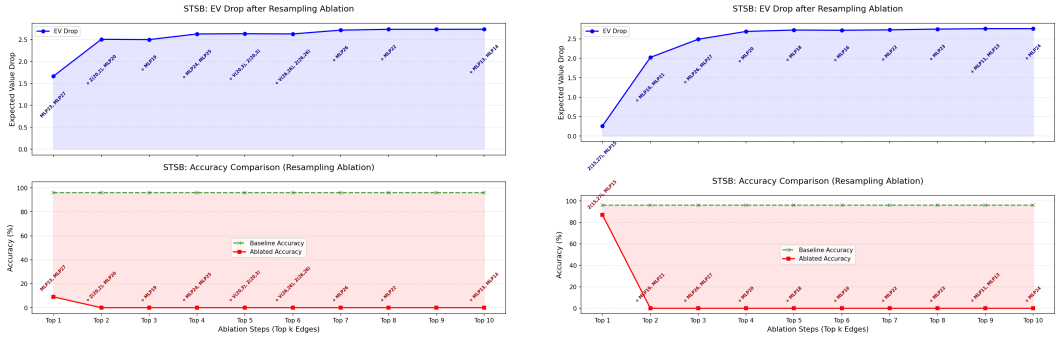
(a) Classification Ablation. (b) Numerical Judgment Ablation.

COLA dataset.



(c) Classification Ablation. (d) Numerical Judgment Ablation.

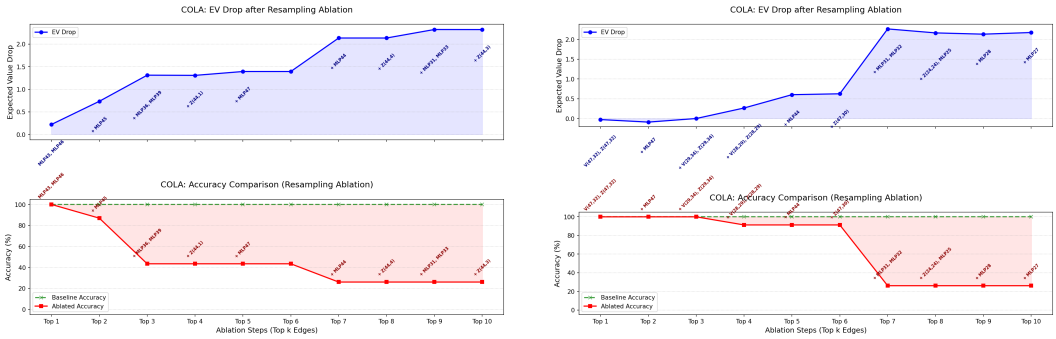
MNLI dataset.



(e) Classification Ablation. (f) Numerical Judgment Ablation.

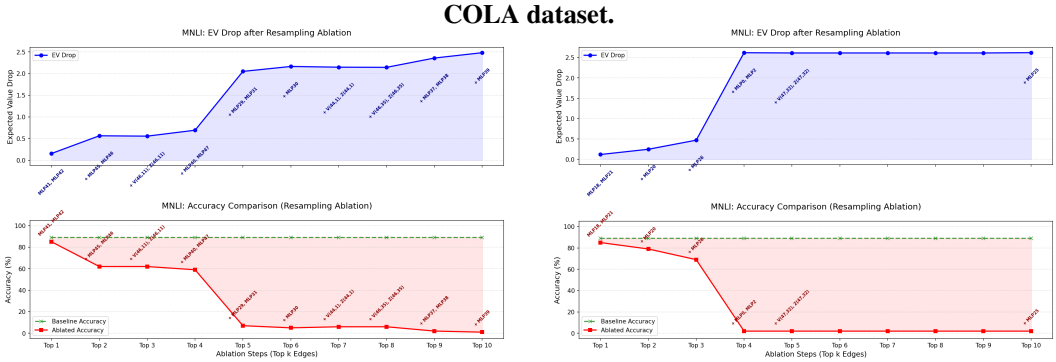
STSB dataset.

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



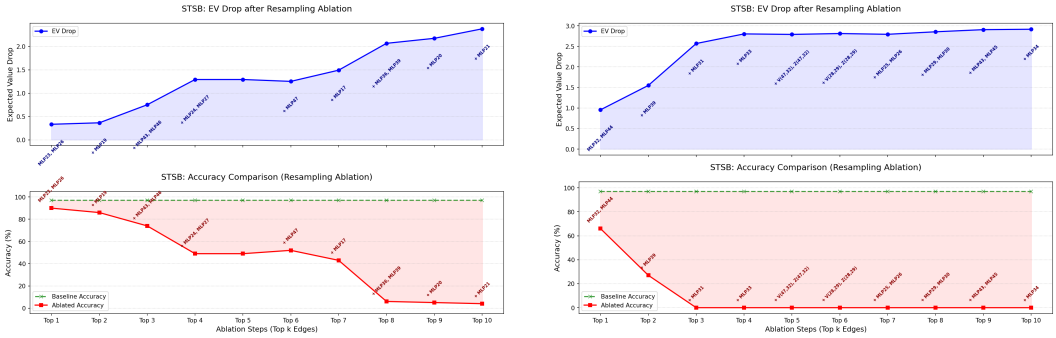
(a) Classification Ablation. (b) Numerical Judgment Ablation.

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).