What Does a Sparse Autoencoder Feature Do? A Weight-Based Account

Yiting Liu **Zhi-Hong Deng***
State Key Laboratory of General Artificial Intelligence
School of Intelligence Science and Technology
Peking University
{arkcia,zhdeng}@pku.edu.cn

Abstract

Sparse autoencoders (SAEs) have emerged as a powerful technique for decomposing language model representations into interpretable features. Current interpretation pipelines infer feature semantics from activation patterns, implicitly assuming every feature has a semantic explanation while overlooking the computational roles features inherit from the SAE training objective. We introduce a weight-based interpretation framework that requires no activation data and grounds claims causally through targeted feature-ablation interventions. Across Gemma-2 and Llama-3.1, three depth-dependent signatures show that SAE features inherit the base model's geometry: semantic features form a U-shape under tied embeddings but bifurcate when untied, attention participation peaks mid-layer, and the two roles couple oppositely for input- vs. output-oriented features. This weight-based view supplies the causal half missing from activation-based interpretability.

1 Introduction

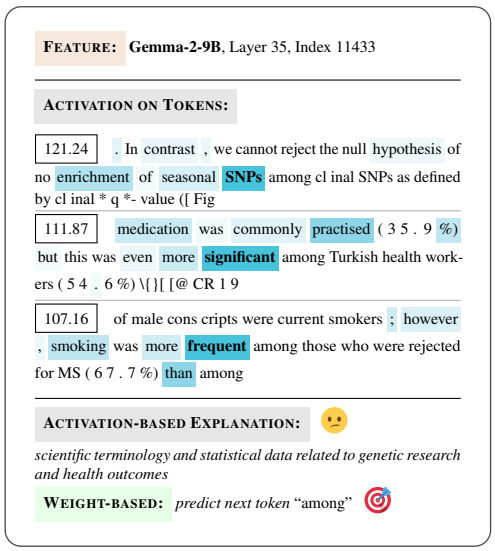
A language model represents more features than it has directions, forcing it to encode features in superposition, which likely explains why many neurons are polysemantic and difficult to explain [1, 2]. Sparse autoencoders (SAEs) have emerged as a promising approach for decomposing such representations into more interpretable features [3–6]. By training higher-dimensional reconstructions of model activations, SAEs aim to disentangle the superposed representations into sparse and semantically meaningful components.

Because SAE features emerge as unlabeled latent variables, their meanings must be inferred post-hoc. The dominant paradigm asks "*when does a feature activate?*" and infers meaning from the input contexts that drive the feature high; at scale this is automated by prompting an LLM with top-activating examples [4, 5, 7]. The resulting descriptions are often broad summaries of co-occurring text and have been shown to be superficial and oversimplified [8]; even when descriptively accurate, they cannot specify what changes in the model's computation when the feature is activated or suppressed. This is a structural limitation: SAEs are trained to reconstruct activations, so features inherit the computational function of those activations, yet activation-based interpretation only reads back the contexts in which the feature appears. Activation-based interpretation is, in this sense, correlational rather than mechanistic, leaving the computational roles of most features unexamined.

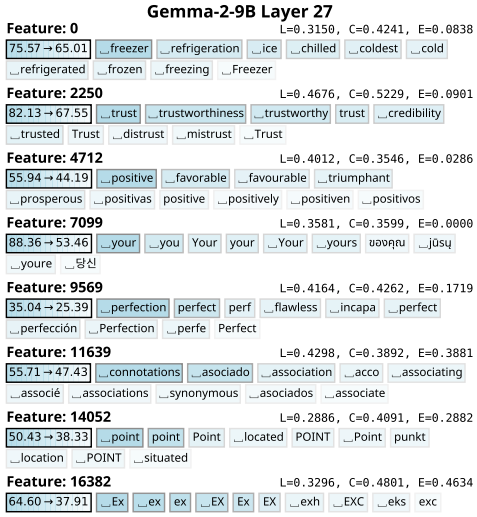
We complement this paradigm with the question that titles the paper: *what does a feature do?* We propose a weight-based methodology that analyzes features through their direct interactions with model components, requiring no datasets or external models, and uses targeted ablation to verify the structurally predicted effect. Because we have full access to the SAE and the surrounding model,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Activation-based explanation by GPT-4o-mini.



(b) Sample of weight-flagged semantic features with their top-10 predicted tokens.

Figure 1: Two views of SAE features. (a) An activation-based explanation read off the feature's top-activating contexts: tokens where the feature fires are highlighted and the highest activations are boxed, and GPT-4o-mini summarizes the co-occurring text. The summary does not specify what changes in the model's computation when the feature is suppressed. (b) A uniform sample of features that pass all three weight-based metrics in Section 4 on Gemma-2-9B at layer 27, shown with their top-10 predicted tokens. "L" is Levenshtein similarity, "C" is cosine similarity, and "E" is top-100 entropy; boxed scores correspond to the range of displayed tokens.

every feature can be projected onto the components they read from and write to, allowing causal claims to be grounded in counterfactual interventions on the feature itself rather than in correlations across activating prompts.

Our contributions are threefold. **1)** We propose a weight-based SAE interpretation framework that requires neither LLM inference nor a corpus, and grounds claims through counterfactual interventions rather than activation correlations. **2)** We investigate both the semantic and the previously overlooked computational roles of SAE features, challenging the implicit assumption underlying activation-based explanations that contexts suffice to characterize what a feature does. **3)** We surface 3 depth-dependent signatures that reveal how SAEs inherit the base model's architectural geometry, with implications for cross-architecture comparison and depth-aggregated feature analysis.

The code to reproduce the experiments is available at `https://github.com/ArkciaTheDragon/what-sae-features-do`.

2 Related Work

As discussed, the predominant approach to SAE interpretability is activation-based: textual examples that activate a feature are presented to an LLM to generate semantic explanations [4, 5, 7]. This paradigm has spurred a rich ecosystem, including online tools for interactive inspection [9] and downstream applications such as classification [10] and specialized training for rare concepts [11]. The quality of these automated explanations remains a known challenge: benchmarking efforts have revealed difficulties in effective steering [12], and providing more context can adversely lead to vaguer explanations [13]. More broadly, faithful interpretability calls for non-opaque means rather than relying on one LLM to interpret another [14].

Computational roles have so far been touched on only in passing. Features have been treated as computational intermediates through input-specific attribution and ablation [6], but that analysis remains in-context. The closest prior work to ours is the "output-centric" analysis of [15], which projects the decoder onto the unembedding ($\boldsymbol{W}_{\text{dec}}\boldsymbol{W}_{\text{U}}$) to enrich an otherwise LLM-driven explanation pipeline;

the authors discard the embedding pairing because it scored worse in their pipeline even on tied-weight models, and do not consider attention circuits. We instead make the weight-based projections the primary unit of analysis: we retain the encoder–embedding pairing, where its divergence from the decoder–unembedding projection exposes a tied-versus-untied bifurcation between architectures (Section 4). Intervention-based scoring has also been proposed as a complementary signal [7], but it still serves to refine correlational explanations rather than to define them.

Two earlier observations underpin the present approach. Next-token prediction was applied to vanilla LLM neurons in [2], where it generally failed to meet the baseline; the gap is consistent with polysemanticity, and motivates carrying the same idea to the disentangled SAE basis. Logit-lens analysis of SAE features in [16] confirmed that many features can be read as promoting or suppressing specific tokens, and the present paper picks up that thread by quantifying when a feature behaves this way versus when its decoder writes do not concentrate on any token at all.

3 Framework

Before the experiments, we consolidate the components shared across them: how we identify a feature, the projections we use to interrogate it, and the intervention primitive that turns a structural prediction into a causal test.

Given an activation $\boldsymbol{x} \in \mathbb{R}^{d_{\text{model}}}$ at one layer of the model, an SAE encodes and then decodes it as

$$\boldsymbol{z} = \sigma\left(\boldsymbol{x}\boldsymbol{W}_{\text{enc}} + \boldsymbol{b}_{\text{enc}}\right), \qquad \hat{\boldsymbol{x}} = \boldsymbol{z}\boldsymbol{W}_{\text{dec}} + \boldsymbol{b}_{\text{dec}}, \tag{1}$$

where $\boldsymbol{z} \in \mathbb{R}^{d_{\text{sae}}}$ is the sparse feature vector and σ is an activation function chosen to induce sparsity, so that most elements of $\boldsymbol{z}$ are zero on any given input. We index features by $i \in \{1, \ldots, d_{\text{sae}}\}$. Prior work usually identifies a feature with its decoder row $\boldsymbol{W}_{\text{dec}}^{(i)} \in \mathbb{R}^{d_{\text{model}}}$ alone [17]. We instead treat a feature as the pair $(\boldsymbol{W}_{\text{enc}}^{(\cdot,i)}, \boldsymbol{W}_{\text{dec}}^{(i)})$ of an encoder column and decoder row, because the two play complementary roles in the forward pass: the encoder column *reads from* the residual stream and gates whether the feature activates, and the decoder row *writes into* the residual stream when it does. Reading and writing are distinct operations, and a feature may carry interpretable signal on either side independently.

Input and output token alignment. The first pair of projections asks what tokens a feature is tied to, on the input side and on the output side:

$$\boldsymbol{l}_{\text{E}}^{i} = \boldsymbol{W}_{\text{enc}}^{(\cdot,i)}\boldsymbol{W}_{\text{E}}^{\top} \qquad \text{(input-token recognition)} \tag{2}$$

$$\boldsymbol{l}_{\text{U}}^{i} = \text{FinalNorm}(\boldsymbol{W}_{\text{dec}}^{(i)})\,\boldsymbol{W}_{\text{U}} \qquad \text{(output-token prediction)} \tag{3}$$

Here $\boldsymbol{W}_{\text{E}} \in \mathbb{R}^{|V| \times d_{\text{model}}}$ is the model's token embedding, $\boldsymbol{W}_{\text{U}} \in \mathbb{R}^{d_{\text{model}} \times |V|}$ is the unembedding, $|V|$ is the vocabulary size, and FinalNorm is the layer normalization the model applies before the unembedding reads. Both $\boldsymbol{l}_{\text{E}}^{i}$ and $\boldsymbol{l}_{\text{U}}^{i}$ are vectors of size $|V|$, one score per token.

Intuitively, $\boldsymbol{l}_{\text{E}}^{i}$ asks which input tokens switch the feature *on*: if a token's embedding sits along the direction the encoder reads, that token in the residual stream tends to drive feature i to activate. Symmetrically, $\boldsymbol{l}_{\text{U}}^{i}$ asks which output tokens the feature pushes the model toward: when the feature writes its decoder row into the residual stream, the unembedding turns that write into a logit increase for the tokens whose unembedding columns align with it. Together these are the bottom-up analogue of activation-based interpretation: rather than collecting prompts and asking what they share, we read the model's own embedding and unembedding to recover what the feature recognises and what it predicts.

Attention participation in the next layer. The second pair of projections asks how a feature's decoder write is consumed by the very next attention layer:

$$\boldsymbol{q}_{i}^{(h)} = \text{AttnNorm}_{L+1}(\boldsymbol{W}_{\text{dec},L}^{(i)})\boldsymbol{W}_{Q,L+1}^{(h)} \qquad \text{(query participation in head } h\text{)} \tag{4}$$

$$\boldsymbol{k}_{i}^{(h)} = \text{AttnNorm}_{L+1}(\boldsymbol{W}_{\text{dec},L}^{(i)})\boldsymbol{W}_{K,L+1}^{(h)} \qquad \text{(key participation in head } h\text{)} \tag{5}$$

Here L indexes the SAE's host layer and h indexes one head in the attention layer at $L+1$; $\boldsymbol{W}_{Q,L+1}^{(h)}, \boldsymbol{W}_{K,L+1}^{(h)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{head}}}$ are that head's query and key projections; and AttnNorm_{L+1} is

the layer normalization applied before the head reads. The resulting $\boldsymbol{q}_i^{(h)}$ and $\boldsymbol{k}_i^{(h)}$ live in the head's d_{head}-dimensional QK space.

Intuitively, when the feature is active at a token position, its decoder row is written into the residual stream there; the next layer's head h then reads that position through its query and key projections. $\boldsymbol{q}_i^{(h)}$ measures how that write tilts the token's outgoing query, and $\boldsymbol{k}_i^{(h)}$ measures how it tilts the token's incoming key. The first asks "does this feature, when active at a position, change what that position is searching for?" and the second asks "does it change how attractive this position is as a target of attention from later positions?" Token-level interpretation cannot see either of these effects: a feature can have a precise computational role in routing attention without ever lining up with a recognisable token on the input or output side.

Intervention primitive. To turn a structural prediction made by any of the four projections into a causal claim, we use a single counterfactual operation [18]. Given a feature i at SAE layer L on a prompt where it activates with value z_i, we set

$$\mathrm{do}(z_i = 0) \tag{6}$$

inside the SAE and re-decode, holding the rest of the forward pass fixed. Equivalently, this subtracts $z_i \boldsymbol{W}_{\text{dec},L}^{(i)}$ once from the residual stream at layer L. Because every other component is untouched, any downstream change is attributable to the feature's own contribution. Sections 4 and 5 pair the four projections with this primitive to test whether each structurally predicted effect is borne out causally.

Three points worth noting. The projections above depend on three modeling choices that attentive readers may reasonably question: (i) the per-feature application of $\mathrm{FinalNorm}$ and $\mathrm{AttnNorm}$ rather than the in-context normalization the model actually uses; (ii) whether the encoder–embedding and decoder–unembedding projections carry distinct signal, given that for tied-embedding models they would coincide if the encoder column and decoder row of a feature were themselves aligned; and (iii) the asymmetry that the do-intervention in Equation (6) validates claims routed through the decoder write but is not the right test for the read-side encoder–embedding projection in Equation (2). Section D addresses each in turn, with the supporting empirical evidence (rank-correlation between normalized and unnormalized scores, encoder–decoder cosine statistics, and the scope of the do-operation).

4 Experiment 1: Semantic Features

Activation-based interpretability assumes that each feature carries semantic meaning and derives a description from its top activating contexts. Here we quantify token-level semantic alignment on the weight side, complementing context-based accounts with an analysis that requires no prompts or LLM annotators. A feature qualifies as semantic under this operational definition when its decoder–unembedding projection $\boldsymbol{l}_{\text{U}}^i$ from Equation (3) or its encoder–embedding projection $\boldsymbol{l}_{\text{E}}^i$ from Equation (2) concentrates on a coherent token cluster. Quantifying this population is the prerequisite for examining how semantic alignment relates to computational participation in Section 6.

4.1 Method

We designed 23 candidate metrics (Section A) and analyzed the pairwise correlations among them (Figure 6) to select a comprehensive set. Ultimately, we chose the following 3 metrics, which demonstrate only moderate correlation (mean absolute Spearman across layers $0.53 < \overline{|\rho|} < 0.69$ for Gemma-2-9B; see Figure 7 for an example of ρ in layer 27), thereby providing complementary perspectives on a feature's properties. Throughout the experiments, semantic and non-semantic label features that pass and fail the joint criterion defined by the following 3 metrics:

Levenshtein similarity. Semantic features should have a low average Levenshtein distance [19] among the top-10 predicted tokens, indicating that these tokens are mostly lexical variants of the same underlying concept (e.g., "http", "HTTP", "https", with or without a leading space).

Cosine similarity. Semantic features should have a high cosine similarity among the embeddings of the top-10 predicted tokens, indicating that the model represents them as closely related. While the

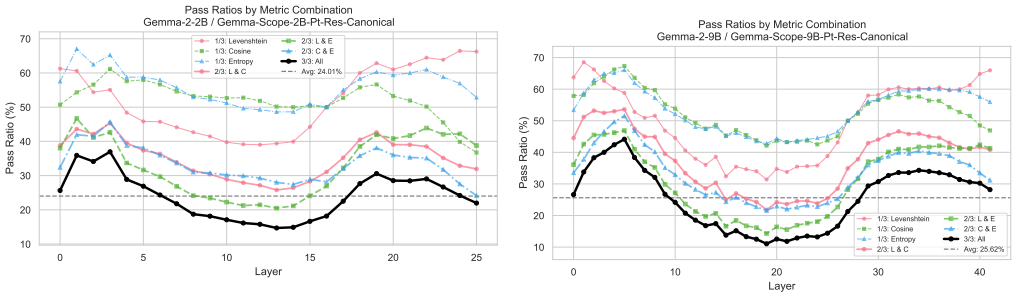


Figure 2: Joint and per-metric pass rates for the decoder–unembedding classifier on Gemma-2-2B and 9B. Both sizes show a U-shaped distribution; ablating any single metric uniformly increases the pass rate, confirming the three are complementary.

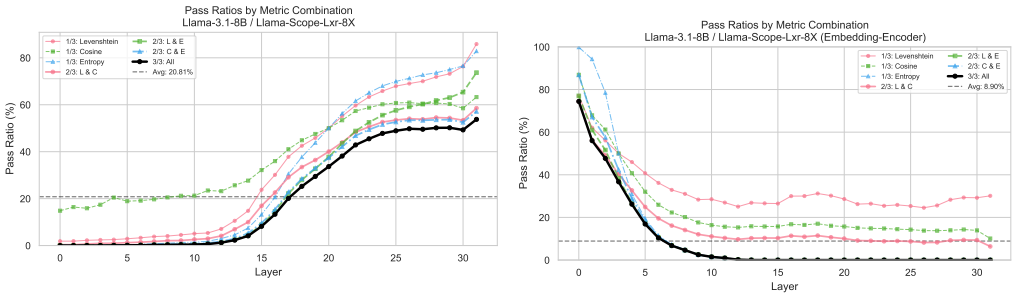


Figure 3: Pass rates for Llama-3.1-8B under the two projections of the dual analysis. The decoder–unembedding pairing (left) is near-zero in early layers and rises with depth; the encoder–embedding pairing (right) is high in early layers and falls. The two trajectories partition the network into an input-recognition regime and an output-prediction regime, each captured by a different projection.

Levenshtein similarity reflects what humans expect, cosine similarity captures the model's perspective and complements the former.

Top-100 Entropy. Semantic features should have low entropy over logits, resulting in a "spiky" distribution where the feature's influence is decisive, often affecting only a single token.

With the three metrics in hand, the next question is how to use them to delimit a semantic population. Each metric is naturally continuous with no apparent cutoff (see Figure 8), but setting a threshold per metric allows us to count, compare, and intervene on populations across layers and models at no additional inference cost. This contrasts with the opaque alternative of using an LLM judge to directly label all features in all layers, which represents the activation-based approach we compare against. The resulting classification depends on the chosen thresholds, so we assess its sensitivity alongside the cross-layer trends.

We calibrate the thresholds by manually inspecting different layers at various values, eventually setting each cutoff at the 50th percentile of the layer at 2/3 depth for the decoder–unembedding pairing (Table 9). We propose this choice as a rule of thumb for future interpretation needs, as it provides sufficient depth to capture well-formed predictive features while avoiding the excessive specialization that occurs at later layers. Figure 1b samples the resulting calibration-layer population. For the encoder–embedding pairing, we calibrate an early layer of the encoder where the population is most informative. We then apply these values at every other layer of the model to surface cross-layers trends. We assess calibration sensitivity through a joint percentile sweep, keeping each calibration layer fixed (full setup in Section B.1). Finally, we employ an LLM judge to verify that our calibration aligns with expectations.

4.2 Results

We apply both projections of the dual analysis ($\boldsymbol{l}^i_{\mathrm{U}}$ and $\boldsymbol{l}^i_{\mathrm{E}}$) to all three models using the 50th-percentile calibration. For Gemma-2-2B and Gemma-2-9B (Figure 2), the decoder–unembedding projection identifies a substantial population of semantic features with mean joint pass rate of 24.01% and

Table 1: Stability of the per-layer joint pass-rate profiles. Each $\rho_{\min}$ is the minimum Spearman correlation with the same cell's $p = 50$ profile over the tested percentiles in the specified band. Calibration layers remain fixed throughout the sweep.

Projection	Model	Calibration layer	$\rho_{\min}, p \in [25, 75]$	$\rho_{\min}, p \in [40, 60]$
Decoder–unembedding	Gemma-2-2B	16	0.971	0.992
	Gemma-2-9B	27	0.953	0.993
	Llama-3.1-8B	20	0.971	0.997
Encoder–embedding	Gemma-2-2B	3	0.720	0.878
	Gemma-2-9B	6	0.905	0.978
	Llama-3.1-8B	3	0.785	0.964

25.62%, respectively. The pass rate is high in the initial layers, decreases through the middle, and rises again toward the final layers, forming a U-shape across depth. The encoder–embedding projection produces a quantitatively similar U-shape across both Gemma sizes (Figure 12), as expected from the tied embedding $\boldsymbol{W}_{\mathrm{E}} = \boldsymbol{W}_{\mathrm{U}}^{\top}$ and the high alignment between the encoder and decoder vectors of Gemma 2 SAEs (Figure 11).

For Llama-3.1-8B (Figure 3) the two projections diverge. Under decoder–unembedding, the pass rate starts near zero in the initial layers and climbs steadily to over 50% at the final layer. Under encoder–embedding the pass rate is exceptionally high in the initial layers and declines rapidly, becoming negligible after the first third of the model. The two projections, asked of the same SAE features, describe two different stages. Early-layer features specialize in recognizing input tokens, visible to the encoder–embedding projection and invisible to the "output-centric" decoder–unembedding one, while late-layer features specialize in predicting output tokens. Untied embeddings in the model shapes the SAE to allocate capacity to whichever side of the residual stream a feature serves.

The percentile sweep supports these architecture-dependent depth patterns: across all six model–projection combinations, the minimum Spearman correlation with the $p = 50$ baseline is approximately 0.878 over $p \in [40, 60]$ (Table 1). The decoder–unembedding profiles remain strongly correlated over the wider range $p \in [25, 75]$, with a minimum correlation of approximately 0.953 across the three models.

To validate the classifier's token-level semantic judgments, we compare its labels with those of Llama-3-70B-Instruct [20] on 1000 semantic and 1000 non-semantic features per layer, across 8 layers per (model, projection) pair, with the judge blinded to label (full setup in Section B.2). The judge's per-feature decisions agree with the classifier at mean Cohen's $\kappa = 0.83$ across all six cells (range 0.80–0.87; Table 3), which is substantial-to-near-perfect under the Landis–Koch bands, and the agreement holds uniformly across model sizes, the tied-vs-untied divide, and both projections despite the calibration layer and absolute thresholds varying. This agreement supports using the inference-free classifier for large-scale analysis of token-level semantic alignment.

The projection-based criterion is a structural prediction. To verify that semantic features influence the model on real inference passes as we predict, we apply $\mathrm{do}(z_i = 0)$ from Equation (6) to 100 semantic features per layer in Gemma-2-9B and Llama-3.1-8B, on each feature's max-activating prompt, and read the change in the next-token logit at the feature's top-1 associated token. Each sampled feature is paired with a matched-norm non-semantic feature drawn from the same layer that fires at the same position, isolating coherent-token alignment as the only difference between the two interventions. Across 16 sweeps, removing the flagged feature produces a strictly larger $|\Delta\text{logit}|$ than its matched-norm baseline in 88–100% of pairs, with median $|\Delta\text{logit}|$ of 0.8–7.1 versus 0.02–0.13 for the baseline. Table 4 reports the per-layer numbers, and the full setup is detailed in Section B.3. The structural projection thus carries a causal signal on the model's output.

The judge validates token-cluster coherence, and the interventions confirm effects on the associated output tokens. Taken together, our results identify a token-coherent subpopulation, challenging the assumption that semantic descriptions alone characterize what every feature does and motivating a complementary analysis of computational roles.

5 Experiment 2: Computational Features

Experiment 1 identifies a token-coherent subpopulation under our weight-based criterion. To characterize feature function more fully, we also examine how decoder directions participate in the model's internal computation. Attention provides a concrete setting for this analysis: decoder directions project into the next layer's query and key circuits, which govern information routing between token positions. We ask whether individual features exhibit specialized attention roles, concentrating on a small set of keys or attracting strong attention from multiple queries. We assess this property across the full feature population, independently of the semantic classification, and trace its variation with depth and architecture.

5.1 Method

Given that SAEs are trained to reconstruct activations that participate in the forward pass, it is natural to expect that their features would inherit these computational roles. The primary mechanism for information routing and composition within a transformer layer is the attention head [21]. We therefore hypothesize that many of these features are specialized for participating in attention mechanisms. In this experiment, we test this by measuring the alignment of SAE features from a given layer with the query and key (QK) circuits of the subsequent layer's attention heads.

The query and key participations $\boldsymbol{q}_i^{(h)}$, $\boldsymbol{k}_i^{(h)}$ defined in Equations (4) and (5) let us read off the pre-softmax attention score between feature i as a query and feature j as a key:

$$s_{i,j}^{(h)} = \frac{\boldsymbol{q}_i^{(h)} \cdot \boldsymbol{k}_j^{(h)}}{\sqrt{d_{\text{head}}}} \quad \in \mathbb{R}. \tag{7}$$

The matrix $\boldsymbol{S}^{(h)} = (s_{i,j}^{(h)}) \in \mathbb{R}^{d_{\text{sae}} \times d_{\text{sae}}}$ records the raw QK alignment between every pair of features at head h (derivation in Section C).

Pre-softmax scores expose raw participation strength but cannot say whether a feature wins a given query's competition: attention is row-wise normalized, so what matters downstream is the relative ordering of scores across keys. We therefore apply softmax row-wise to $\boldsymbol{S}^{(h)}$, obtaining the simulated post-softmax attention matrix $\boldsymbol{A}^{(h)} \in \mathbb{R}^{d_{\text{sae}} \times d_{\text{sae}}}$ whose entry $a_{i,j}^{(h)}$ is the fraction of attention feature i (as a query) would allocate to feature j (as a key) in head h.

We now define two complementary post-softmax criteria that consider both the query and key aspects to identify computationally specialized features. Let $\boldsymbol{A}_{i,\cdot}^{(h)}, \boldsymbol{A}_{\cdot,j}^{(h)} \in \mathbb{R}^{d_{\text{sae}}}$ denote the i-th row and the j-th column, respectively, of the post-softmax attention matrix $\boldsymbol{A}^{(h)}$.

Query-based specialization. A feature i is classified as a *query specialist* if, for at least one head h, the sum of its top-k attention allocations exceeds threshold τ_Q:

$$\textstyle\sum_{j \in \mathcal{I}_k(\boldsymbol{A}_{i,\cdot}^{(h)})} a_{i,j}^{(h)} > \tau_Q, \tag{8}$$

where $\mathcal{I}_k(\boldsymbol{v}) = \{j_1, \ldots, j_k\}$ denotes the index set of the k largest components of vector $\boldsymbol{v}$. This identifies features that concentrate their attention on a small set of targets, acting as specialized "lookers" for certain keys.

Key-based specialization. A feature j is classified as a *key hub* if, for at least one head h, the mean attention it receives from its top-k attending queries exceeds threshold τ_K. Then:

$$\tfrac{1}{k} \textstyle\sum_{i \in \mathcal{I}_k(\boldsymbol{A}_{\cdot,j}^{(h)})} a_{i,j}^{(h)} > \tau_K. \tag{9}$$

This identifies features that are emphasized by multiple queries, serving as information hubs.

5.2 Results

Pre-softmax. To visualize the raw computational potential across layers, we first analyze the evolution of pre-softmax attention scores. Positive pre-softmax values signify a tendency for constructive alignment within a head's QK circuit, while negative values suggest anti-alignment.

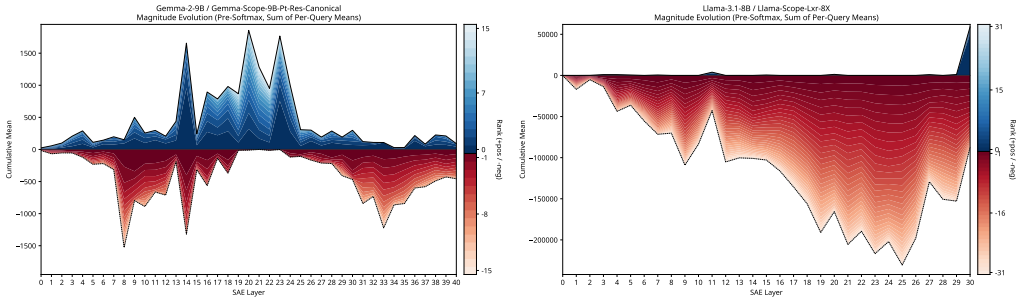


Figure 4: Main results for Experiment 2 on Gemma-2-9B and Llama-3.1-8B: Magnitude of attention participation across layers, with the total height at each layer representing the aggregate pre-softmax score. These scores are calculated as the mean across all keys for each query feature, then summed for each head. The stacking direction of the head values is determined based on whether they are positive or negative. Each color represents a rank of magnitude, rather than a specific head index, across all layers.

In Gemma-2-9B (Figure 4 left) and Gemma-2-2B (Figure 13), we observe a distinct low–high–low pattern in the aggregate attention potential. The scores are negative in early and late layers and become strongly positive in the middle layers, mirroring the inverse of the semantic U-shape from Experiment 1. This suggests that middle-layer features are indeed configured for constructive, high-potential interactions within QK circuits.

At first glance, Llama-3.1-8B (Figure 4 right) appears to operate "wrongly". Its aggregate pre-softmax scores are almost uniformly negative across all layers. However, this is not necessarily a contradiction. Consider that the softmax function is shift-invariant; only the relative differences between scores matter for the final attention distribution. In line with this, the absolute magnitude of Llama's scores follows a similar low–high–medium pattern, peaking in the middle layers where computational activity is expected to be highest. The pre-softmax magnitude evolution thus places both architectures on the same depth-dependent computational rhythm, even though Gemma achieves it via signed alignment and Llama via signed disalignment.

Post-softmax. We set $k = 10$, $\tau_Q = 0.95$, and $\tau_K = 0.25$. These thresholds are chosen to surface clearly focused queries and clearly hub-like keys; qualitative depth profiles are stable to substantial loosening (e.g., τ_Q down to 0.25, τ_K down to 0.05; full rationale in Section E), confirming that the shapes are architectural signatures. Across all three models, both criteria identify non-trivial specialist populations at every depth (Figures 14 to 16); per-head mean counts remain low while the any-head union is large, indicating that each feature plays a specialist role only in a few specific head contexts rather than uniformly across heads.

A more architecture-revealing pattern emerges in how those populations are paced through depth. In both Gemma-2 SAEs, an oscillatory pattern emerges with a 2-layer period (Figures 14 and 15). This period matches Gemma-2's documented alternation between local-window and full attention across consecutive layers [22]. The oscillation is absent in Llama-3.1-8B, confirming that an SAE's computational properties track the model architecture from which they originated.

Intervention validation. We test the causal effect of the weight-identified specialists by applying Equation (6) to flagged features on their max-activating prompt: removing a query specialist should perturb the attention row at the firing position in the flagged head, and removing a key hub should influence the incoming attention from queries later in the sequence. Each flagged feature is paired with a non-specialist baseline matched on decoder norm and firing on the same prompt at the same position, isolating the structural property under inspection. Across a total of 35 sweeps, each comprising up to 100 feature pairs, the flagged feature's effect on its target attention quantity exceeds its matched baseline by a median per-pair ratio of $43\times$–$2161\times$ (queries) and $9\times$–$37\times$ (keys), with win-rates of 0.93–1.00 and 0.88–0.98. Our framework therefore identifies features that causally steer the next layer's attention during inference. Full setup and per-cell numbers are presented in Section F.

Together with Section 4, these results expose two populations of features at the resolution of single layers: a semantic population that writes into the embedding/unembedding path, and a computational population that writes into the next layer's QK circuits.

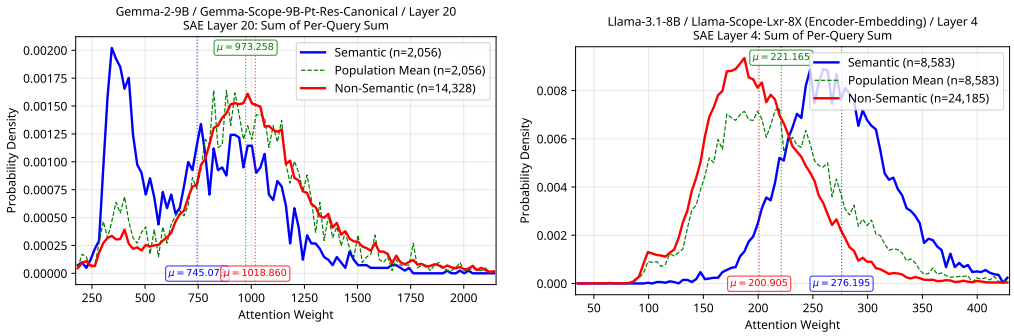


Figure 5: Per-layer query-side attention-weight densities for semantic vs. non-semantic features. Left: Gemma-2-9B layer 20, where the semantic distribution sits below the non-semantic one. Right: Llama-3.1-8B layer 4, in the input-semantic phase, where the relation inverts and semantic features carry the higher attention weights.

6 Experiment 3: Semantic vs. Computational Specialization

Experiments 1 and 2 characterized semantic and computational features as independent properties. We now ask how the two populations relate: do semantic features show systematically different attention participation than non-semantic ones, and does that relationship vary with depth and architecture? If the two roles are inversely related, semantic features should engage QK circuits less; if orthogonal, both populations should display similar attention distributions; if positively coupled in some layers, the same features would simultaneously serve interpretive and computational functions.

6.1 Method

For each feature i at layer L, we summarize its participation in the attention of layer $L{+}1$ from two sides. As a query, we sum its top-k pre-softmax scores over the keys it most strongly attends to, across all heads:

$$w_i^Q = \sum_{h=1}^{n_{\text{heads}}} \sum_{j \in \mathcal{I}_k(\boldsymbol{s}_i^{(h)})} s_{i,j}^{(h)}, \tag{10}$$

where $s_{i,j}^{(h)}$ is from Equation (7) and $\mathcal{I}_k(\cdot)$ selects the indices of the k largest components. As a key, we sum the top-k scores over the queries that most strongly attend to it:

$$w_j^K = \sum_{h=1}^{n_{\text{heads}}} \sum_{i \in \mathcal{I}_k(\boldsymbol{s}_{\cdot,j}^{(h)})} s_{i,j}^{(h)}. \tag{11}$$

w_i^Q thus measures how concentrated a feature's outgoing attention is when it acts as a query, and w_j^K how concentrated the incoming attention is when it acts as a key. We use $k = 10$ as in Experiment 2.

We then compare semantic features (passing all three metric thresholds) against non-semantic features per-layer via density estimates and boxplots, and across layers via mean, median, and the share of features exceeding the global P_{75} threshold pooled across all layers.

6.2 Results

Per-layer distributions. The semantic and non-semantic populations overlap substantially, but the semantic distribution is shifted toward lower values (Figure 5, left). The non-semantic side carries a significantly heavier right tail, which the box-plot confirms (Figure 18).

Cross-layer trends. The picture sharpens when we move from individual layers to depth-wide aggregates. In both Gemma-2-2B (Figures 25 and 29) and Gemma-2-9B (Figures 26 and 30), non-semantic features hold a mean/median lead across most layers, and the separation widens at the upper tail: the share of features exceeding the global P_{75} threshold is markedly lower for semantic features

(Figures 33 and 34). Non-semantic features therefore dominate the most computationally influential regime, and the effect reproduces across two Gemma sizes, indicating it is not an artifact.

Llama's dual-phase architecture. The untied Llama-3.1-8B admits two semantically aligned subpopulations that occupy disjoint depth ranges. We split it at layer 11, where the two pass-rate trajectories cross below 1% (Figure 3), and analyse each phase against their respective non-semantic baseline. The output-semantic phase mirrors Gemma (Figure 23): semantic features sit below non-semantic features in mean (Figure 27), median (Figure 31), and tail (Figure 35). The input-semantic phase, however, inverts the pattern: embedding-aligned semantic features can attain *higher* attention weights than non-semantic features (Figure 5, right). A natural reading is that recognizing and encoding incoming tokens for downstream consumption is itself an attention-heavy job at the beginning of the layers.

These results refine the picture from Experiments 1 and 2. The relationship between the two roles is not universally inverse. It is shaped by depth, by the embedding–unembedding structure, and by whether a feature serves input encoding or output prediction.

7 Conclusion

We reframed SAE interpretation around the question *what does a feature do?*, replacing prompts and LLM annotators with four weight-space projections and a single ablation primitive that turns each structural prediction into a causal test. Reading the encoder and decoder against the model's own embedding, unembedding, and next-layer QK circuits surfaces three depth-dependent signatures that all trace back to the host architecture. First, semantic depth profiles form a U-shape in tied-embedding Gemma and separate into input-recognition and output-prediction regimes in untied Llama. Second, a population of features specializes in steering the next layer's attention, with a depth profile that tracks the host's computational schedule. Third, whether semantics and computation compete or coincide depends on the purpose a feature serves and the trade-off involved in how features allocate their representational capacity to specialize.

Each of the three signatures is an architectural fingerprint apparent only from the weight side. Activation-based interpretation reads features one prompt at a time and cannot see this structure. The two views answer different halves of the same question, and we call on future interpretability research to incorporate the weight-based aspects that have been neglected.

Acknowledgments and Disclosure of Funding

The authors received no third-party funding or support for this work. The authors declare no competing financial interests relevant to this work.

References

[1] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah, "Toy Models of Superposition," Sept. 2022.

[2] S. Bills, N. Cammarata, D. Mossing, H. Tillman, L. Gao, G. Goh, I. Sutskever, J. Leike, J. Wu, and W. Saunders, "Language models can explain neurons in language models." `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023.

[3] R. Huben, H. Cunningham, L. R. Smith, A. Ewart, and L. Sharkey, "Sparse Autoencoders Find Highly Interpretable Features in Language Models," in *The Twelfth International Conference on Learning Representations*, Oct. 2023.

[4] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. L. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah, "Towards monosemanticity: Decomposing language models with dictionary learning," Oct. 2023.

[5] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu, "Scaling and evaluating sparse autoencoders," in *The Thirteenth International Conference on Learning Representations*, Oct. 2024.

[6] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, A. Tamkin, E. Durmus, T. Hume, F. Mosconi, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan, "Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet," May 2024.

[7] G. S. Paulo, A. T. Mallen, C. Juang, and N. Belrose, "Automatically Interpreting Millions of Features in Large Language Models," in *Forty-Second International Conference on Machine Learning*, June 2025.

[8] G. Ma, S. Pfrommer, and S. Sojoudi, "Revising and Falsifying Sparse Autoencoder Feature Explanations," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, Oct. 2025.

[9] J. Lin, "Neuronpedia: Interactive reference and tooling for analyzing neural networks," 2023. Software available from neuronpedia.org.

[10] J. Gallifant, S. Chen, K. Sasse, H. Aerts, T. Hartvigsen, and D. Bitterman, "Sparse Autoencoder Features for Classifications and Transferability," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing* (C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, eds.), (Suzhou, China), pp. 29927–29951, Association for Computational Linguistics, Nov. 2025.

[11] A. Muhamed, M. T. Diab, and V. Smith, "Decoding Dark Matter: Specialized Sparse Autoencoders for Interpreting Rare Concepts in Foundation Models," in *Findings of the Association for Computational Linguistics: NAACL 2025* (L. Chiruzzo, A. Ritter, and L. Wang, eds.), (Albuquerque, New Mexico), pp. 1604–1635, Association for Computational Linguistics, Apr. 2025.

[12] Z. Wu, A. Arora, A. Geiger, Z. Wang, J. Huang, D. Jurafsky, C. D. Manning, and C. Potts, "AxBench: Steering LLMs? Even Simple Baselines Outperform Sparse Autoencoders," in *Forty-Second International Conference on Machine Learning*, June 2025.

[13] C. Juang, G. Paulo, J. Drori, and N. Belrose, "Open Source Automated Interpretability for Sparse Autoencoder Features." https://blog.eleuther.ai/autointerp/, July 2024.

[14] J. Huang, A. Geiger, K. D'Oosterlinck, Z. Wu, and C. Potts, "Rigorously assessing natural language explanations of neurons," in *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP* (Y. Belinkov, S. Hao, J. Jumelet, N. Kim, A. McCarthy, and H. Mohebbi, eds.), (Singapore), pp. 317–331, Association for Computational Linguistics, Dec. 2023.

[15] Y. Gur-Arieh, R. Mayan, C. Agassy, A. Geiger, and M. Geva, "Enhancing Automated Interpretability with Output-Centric Feature Descriptions," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, eds.), (Vienna, Austria), pp. 5757–5778, Association for Computational Linguistics, July 2025.

[16] J. L. Joseph Bloom, "Understanding sae features with the logit lens." `https://www.lesswrong.com/posts/qykrYY6rXXM7EEs8Q/understanding-sae-features-with-the-logit-lens`, 2024.

[17] A. Galichin, A. Dontsov, P. Druzhinina, A. Razzhigaev, O. Y. Rogov, E. Tutubalina, and I. Oseledets, "I Have Covered All the Bases Here: Interpreting Reasoning Features in Large Language Models via Sparse Autoencoders," Mar. 2025.

[18] J. Pearl, *Causality: Models, Reasoning and Inference*. New York: Cambridge University Press, 2 ed., 2009.

[19] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet physics. Doklady*, vol. 10, pp. 707–710, 1965.

[20] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal,

N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. E. Tan, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Grattafiori, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Vaughan, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Franco, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Wyatt, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Ozgenel, F. Caggioni, F. Guzmán, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Thattai, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, I. Damlaj, I. Molybog, I. Tufanov, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Prasad, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Huang, K. Chawla, K. Lakhotia, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Tsimpoukelli, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. P. Laptev, N. Dong, N. Zhang, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Li, R. Hogan, R. Battey, R. Wang, R. Maheswari, R. Howes, R. Rinott, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Kohler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Albiero, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wang, X. Wu, X. Wang, X. Xia, X. Wu, X. Gao, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Hao, Y. Qian, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, and Z. Zhao, "The Llama 3 Herd of Models," Aug. 2024.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. ukasz Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[22] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, J. Ferret, P. Liu, P. Tafti, A. Friesen, M. Casbon, S. Ramos, R. Kumar, C. L. Lan, S. Jerome, A. Tsitsulin, N. Vieillard, P. Stanczyk, S. Girgin, N. Momchev, M. Hoffman, S. Thakoor, J.-B. Grill, B. Neyshabur, O. Bachem, A. Walton, A. Severyn, A. Parrish, A. Ahmad, A. Hutchison, A. Abdagic, A. Carl, A. Shen, A. Brock, A. Coenen, A. Laforge, A. Paterson, B. Bastian, B. Piot, B. Wu, B. Royal, C. Chen, C. Kumar, C. Perry, C. Welty, C. A. Choquette-Choo, D. Sinopalnikov, D. Weinberger, D. Vijaykumar, D. Rogozińska, D. Herbison, E. Bandy, E. Wang, E. Noland, E. Moreira, E. Senter, E. Eltyshev, F. Visin, G. Rasskin, G. Wei, G. Cameron, G. Martins, H. Hashemi, H. Klimczak-Plucińska, H. Batra, H. Dhand, I. Nardini, J. Mein, J. Zhou, J. Svensson, J. Stanway, J. Chan, J. P. Zhou, J. Carrasqueira, J. Iljazi, J. Becker, J. Fernandez, J. van Amersfoort, J. Gordon, J. Lipschultz, J. Newlan, J.-y. Ji, K. Mohamed, K. Badola, K. Black, K. Millican, K. McDonell, K. Nguyen, K. Sodhia, K. Greene, L. L. Sjoesund, L. Usui, L. Sifre, L. Heuermann, L. Lago, L. McNealus, L. B. Soares, L. Kilpatrick, L. Dixon, L. Martins, M. Reid, M. Singh, M. Iverson, M. Görner, M. Velloso, M. Wirth, M. Davidow, M. Miller, M. Rahtz, M. Watson, M. Risdal, M. Kazemi, M. Moynihan, M. Zhang, M. Kahng, M. Park, M. Rahman, M. Khatwani, N. Dao, N. Bardoliwalla, N. Devanathan, N. Dumai, N. Chauhan, O. Wahltinez, P. Botarda, P. Barnes, P. Barham, P. Michel, P. Jin, P. Georgiev, P. Culliton, P. Kuppala, R. Comanescu,

R. Merhej, R. Jana, R. A. Rokni, R. Agarwal, R. Mullins, S. Saadat, S. M. Carthy, S. Cogan, S. Perrin, S. M. R. Arnold, S. Krause, S. Dai, S. Garg, S. Sheth, S. Ronstrom, S. Chan, T. Jordan, T. Yu, T. Eccles, T. Hennigan, T. Kocisky, T. Doshi, V. Jain, V. Yadav, V. Meshram, V. Dharmadhikari, W. Barkley, W. Wei, W. Ye, W. Han, W. Kwon, X. Xu, Z. Shen, Z. Gong, Z. Wei, V. Cotruta, P. Kirk, A. Rao, M. Giang, L. Peran, T. Warkentin, E. Collins, J. Barral, Z. Ghahramani, R. Hadsell, D. Sculley, J. Banks, A. Dragan, S. Petrov, O. Vinyals, J. Dean, D. Hassabis, K. Kavukcuoglu, C. Farabet, E. Buchatskaya, S. Borgeaud, N. Fiedel, A. Joulin, K. Kenealy, R. Dadashi, and A. Andreev, "Gemma 2: Improving Open Language Models at a Practical Size," Oct. 2024.

[23] M. Brzozowski, "Bimodality of Sparse Autoencoder Features is Still There and Can Be Fixed," in *Mechanistic Interpretability Workshop at NeurIPS 2025*, Sept. 2025.

[24] D. Chanin, J. Wilken-Smith, T. Dulka, H. Bhatnagar, S. Golechha, and J. I. Bloom, "A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, Oct. 2025.

A Interpretability Metrics

We introduce a comprehensive suite of interpretability metrics designed to quantify different aspects of SAE feature behavior based on their logit attribution patterns.

A.1 Semantic Coherence Metrics

Cosine Similarity measures the mean pairwise cosine similarity between embeddings of the top-k tokens. Let $\mathcal{T}_k = \{t_1, \ldots, t_k\}$ denote the top-k tokens by attribution score, and $e_i \in \mathbb{R}^d$ their corresponding embeddings. We compute:

$$\text{CosineSim}_k = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{e_i \cdot e_j}{\|e_i\| \|e_j\|} \tag{12}$$

Higher values indicate that top tokens occupy similar regions in embedding space, suggesting semantic coherence.

Levenshtein Similarity quantifies string-level similarity between tokens using normalized edit distance. For each pair of tokens (t_i, t_j) in the top-k set:

$$\text{LevenSim}_k = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \left(1 - \frac{d_{\text{Lev}}(t_i, t_j)}{\max(|t_i|, |t_j|)}\right) \tag{13}$$

where d_{Lev} is the Levenshtein distance and $|t|$ denotes string length. This metric captures syntactic patterns such as shared morphological features.

String Overlap measures the fraction of unique normalized tokens in the top-k set. After removing non-word characters and lowercasing, we compute:

$$\text{Overlap}_k = 1 - \frac{|\{\text{normalize}(t) : t \in \mathcal{T}_k\}|}{k} \tag{14}$$

Higher values indicate repeated tokens (possibly with different capitalization or punctuation).

A.2 Distribution Concentration Metrics

Entropy quantifies the concentration of attribution mass across the vocabulary. Given attribution scores $s = (s_1, \ldots, s_V)$ for vocabulary size V, we compute the Shannon entropy of the softmax distribution:

$$H_k(s) = -\sum_{i \in \text{top-}k} p_i \log p_i, \quad p_i = \frac{\exp(s_i/\tau)}{\sum_{j \in \text{top-}k} \exp(s_j/\tau)} \tag{15}$$

where τ is temperature (default 1.0). Lower entropy indicates more concentrated attribution, suggesting clearer feature semantics. We compute this at multiple scales: top-10, top-25, top-50, top-100, and full vocabulary.

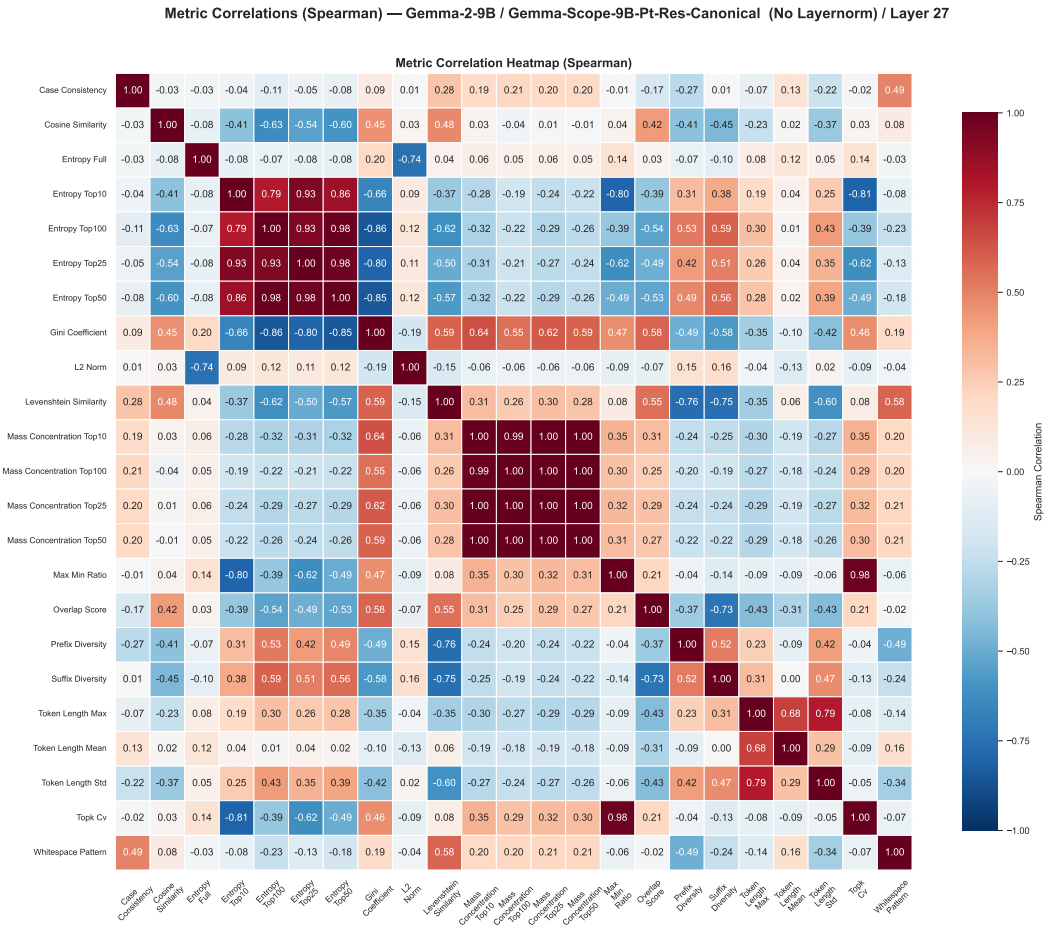


Figure 6: Spearman rank correlation (ρ) matrix for 23 candidate semantic metrics, calculated on all features from the Gemma-2-9B layer 27 SAE.

Mass Concentration measures the fraction of total positive attribution mass contained in the top-k tokens:

$$\text{Mass}_k = \frac{\sum_{i=1}^{k} s_i}{\sum_{j:s_j>0} s_j} \tag{16}$$

Higher values indicate that a small number of tokens account for most of the feature's impact.

Gini Coefficient quantifies inequality in the score distribution. For top-k scores sorted in ascending order $(s_{(1)}, \ldots, s_{(k)})$:

$$G_k = \frac{2\sum_{i=1}^{k} i \cdot s_{(i)}}{k\sum_{i=1}^{k} s_{(i)}} - \frac{k+1}{k} \tag{17}$$

Values range from 0 (perfect equality) to 1 (maximal inequality). Higher values indicate sparser, more concentrated attribution.

Coefficient of Variation (CV) measures relative dispersion of the top-k scores:

$$\text{CV}_k = \frac{\sigma(\{s_i | i \in \text{top-}k\})}{\mu(\{s_i | i \in \text{top-}k\})} \tag{18}$$

where σ and μ denote standard deviation and mean. Lower values indicate more uniform score magnitudes among top tokens.

Max/Min Ratio captures the dominance of the top-scoring token:

$$\text{Ratio}_k = \frac{s_1}{s_k} \tag{19}$$

where s_1 is the highest score and s_k is the k-th highest. Higher values indicate a single dominant token.

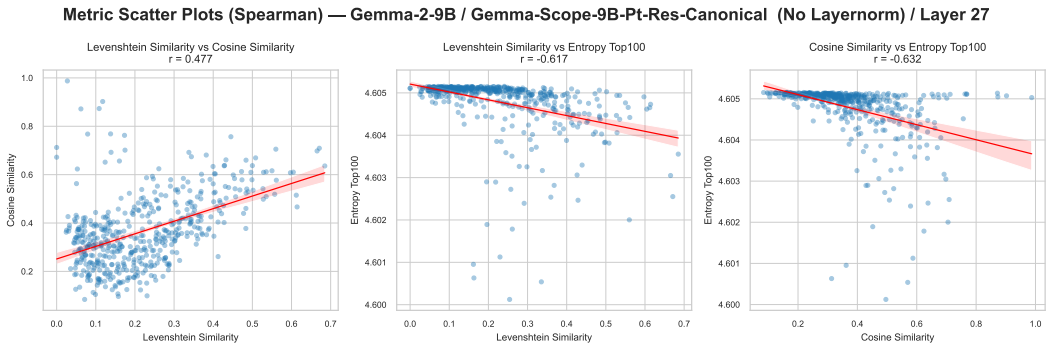


Figure 7: Pairwise scatter samples illustrating the relationships between the three selected semantic metrics for the Gemma-2-9B layer 27 SAE. The plots demonstrate that while the metrics are not entirely independent, their correlations are moderate. This supports their selection as a complementary set, where each metric provides a distinct perspective for evaluating the semantic coherence of a feature.

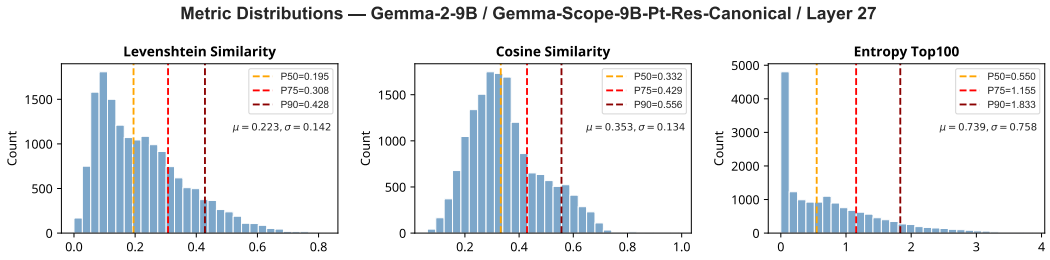


Figure 8: Distributions of the three primary semantic metrics—Levenshtein similarity (higher is better), cosine similarity (higher is better), and top-100 entropy (lower is better)—for all features in the Gemma-2-9B layer 27 SAE. The distributions are continuous and generally unimodal, lacking a distinct bimodal structure or natural cutoff point. This characteristic motivates our use of a percentile-based thresholding approach in Experiment 1, as it provides a consistent relative standard for what constitutes a "semantic" feature, rather than relying on an arbitrary absolute value. The distribution plot applies normalization (RMSNorm) because, without it, the entropy is highly concentrated near the maximum value with extreme outliers, making visualization difficult.

L2 Norm measures the overall magnitude of feature impact, normalized by vocabulary size:

$$\|s\|_{\text{norm}} = \frac{\|s\|_2}{\sqrt{V}} \tag{20}$$

This provides scale-independent comparison of attribution strength across features.

A.3 Morphological Pattern Metrics

Prefix/Suffix Diversity quantify the variety of token beginnings and endings. For prefix length ℓ (default 3):

$$\text{PrefixDiv}_k = \frac{|\{t[:\ell] : t \in \mathcal{T}_k\}|}{k} \tag{21}$$

and analogously for suffixes using $t[-\ell:]$. Lower diversity suggests morphological consistency (e.g., shared verb conjugations).

Token Length Statistics include mean, standard deviation, and maximum character length across top-k tokens. These capture whether features specialize to tokens of particular lengths.

Case Consistency measures the fraction of top-k tokens sharing the dominant case pattern (lowercase, uppercase, title case, or mixed):

$$\text{CaseConsist}_k = \frac{\max_{c \in \{\text{lower, upper, title, mixed}\}} |\{t \in \mathcal{T}_k : \text{case}(t) = c\}|}{k} \tag{22}$$

Whitespace Pattern computes the fraction of top-k tokens beginning with whitespace, capturing formatting conventions:

$$\text{Whitespace}_k = \frac{|\{t \in \mathcal{T}_k : t[0] \in \text{whitespace}\}|}{k} \tag{23}$$

To avoid redundancy and select a minimal set of metrics that provide complementary perspectives, we performed a correlation analysis. We computed the pairwise Spearman rank correlation coefficient (ρ) among all 23 metrics for the features in a representative SAE (Gemma-2-9B, layer 27). As illustrated in Figure 6, this analysis revealed distinct clusters of highly correlated metrics, indicating that many captured similar underlying properties.

B Validating the Semantic Population

This appendix evaluates the three-criteria semantic classifier introduced in Section 4 through three complementary checks: threshold sensitivity, LLM-judge agreement, and decoder-write interventions. The threshold sweep tests how the population size and depth profile depend on calibration. The judge reads each feature's top logit-lens tokens, blinded to label, to assess token-cluster coherence. The $\text{do}(z_i = 0)$ intervention tests whether the projection-based prediction corresponds to a downstream behavioral effect.

B.1 Threshold sensitivity

Setup. We introduce a joint strictness percentile $p \in \{25, 30, \ldots, 75\}$ and keep the calibration layer fixed for each (model, projection) cell, using the same layers as in Table 9. The Levenshtein-similarity and cosine-similarity cutoffs are set at percentile P_p of their respective calibration-layer distributions, and the top-100 entropy cutoff is set at P_{100-p}. Thus, increasing p makes all three criteria stricter, and $p = 50$ reproduces the paper's calibration. At every sweep point we apply the resulting cutoffs to every layer and count features passing all three criteria. Table 2 reports the arithmetic mean of these per-layer pass rates. For each cell, we also compute the Spearman correlation between the per-layer pass-rate profile at each p and its $p = 50$ baseline, summarizing the minimum within each tested percentile band in Table 1.

Results. The threshold sweep supports the architecture-dependent depth patterns identified in Experiment 1. Around the chosen calibration, Gemma retains its mid-depth decoder–unembedding troughs, and Llama retains the contrasting depth trends of its encoder–embedding and decoder–unembedding profiles. Across all six model–projection combinations, the minimum Spearman correlation with the $p = 50$ baseline is approximately 0.878 over $p \in [40, 60]$. The decoder–unembedding profiles remain highly correlated over the wider range $p \in [25, 75]$, with a minimum correlation of approximately 0.953 across the three models.

B.2 LLM-judge agreement at scale

The three-criteria classifier in Section 4 is calibrated by taking each metric's 50th percentile over the feature population at one representative layer (the model's 2/3-depth for the decoder–unembedding pairing; see Table 9) and reusing those three numbers at every other layer. The thresholds are then validated by manual inspection of sampled features. To check that this inference-free decision rule scales beyond our manual inspection, we ask an external reader to classify the same features under the same definition.

Table 2: Threshold sensitivity of the three-criteria classifier. Entries are mean joint pass rates (%) across all layers. The joint strictness percentile p raises both similarity cutoffs and lowers the entropy cutoff; the bold row reproduces the paper's calibration.

	Decoder–unembedding			Encoder–embedding		
p	Gemma-2-2B	Gemma-2-9B	Llama-3.1-8B	Gemma-2-2B	Gemma-2-9B	Llama-3.1-8B
25	50.4	50.5	33.4	48.8	44.9	12.5
30	43.5	43.8	30.8	42.1	39.0	11.5
35	37.4	38.2	28.3	36.8	34.5	10.7
40	32.3	33.7	25.8	32.2	30.8	10.1
45	28.0	29.4	23.4	28.3	27.8	9.5
50	**24.0**	**25.6**	**20.8**	**24.7**	**25.2**	**8.9**
55	20.8	22.1	18.1	20.3	22.2	8.2
60	17.9	18.9	15.4	13.3	17.3	7.2
65	15.2	15.9	12.4	7.6	11.1	5.7
70	12.6	12.9	9.6	4.5	7.1	4.3
75	9.9	10.0	6.9	2.7	4.5	3.2

Setup. For each (model, projection, layer) cell we draw 1000 strictly-passing features (passing all three thresholds) and 1000 strictly-failing features (failing all three). We render each feature's top-5 logit-lens tokens with leading-space markers preserved, present them to Llama-3-70B-Instruct [20], and ask whether the five tokens collectively express one coherent meaning. The judge sees the token list only. Outputs are restricted to {yes, no}, with six fixed few-shot exemplars and temperature 0. We aggregate the per-feature verdicts to per-layer sensitivity, specificity, agreement, and Cohen's κ, and average across the eight sampled layers within each cell.

Result. Table 3 reports the six (model, projection) cells. Mean κ ranges from 0.80 to 0.87 across cells, with both Gemma sizes and both Llama projections falling inside that band. The overall aggregate sits at $\kappa = 0.83$. By the Landis–Koch bands, this is substantial-to-near-perfect agreement.

Two key takeaways emerge from these findings. First, the percentile-calibrated classifier proves to be effective: a capable LLM that reads only the same logit-lens tokens reproduce its decisions reliably at scale. Second, the agreement remains consistent across different model sizes and both projection methods, despite variations in the calibration layer and absolute thresholds across cells (see Table 9). This consistency suggests that the classifier can be extended to resource-constrained settings where running an LLM judge would be prohibitively expensive, such as when conducting comprehensive feature sweeps across every layer.

Table 3: LLM-judge agreement with the three-criteria semantic classifier. For each (model, projection) cell we sample 1000 strict-pass and 1000 strict-fail features at every listed layer (top-5 logit-lens tokens shown to the judge, blinded to label). Sensitivity, specificity, agreement, and Cohen's κ are computed per layer and averaged within the cell. Judge: Llama-3-70B-Instruct.

Projection	Model	Layers	Sensitivity	Specificity	Agreement	Cohen's κ
Decoder–Unembedding	Gemma-2-2B	2, 5, 8, 11, 14, 17, 20, 23	0.916	0.908	0.912	0.824
	Gemma-2-9B	5, 10, 15, 20, 25, 30, 35, 40	0.954	0.862	0.908	0.816
	Llama-3.1-8B	16, 18, 20, 22, 24, 26, 28, 30	0.889	0.925	0.907	0.814
Encoder–Embedding	Gemma-2-2B	2, 5, 8, 11, 14, 17, 20, 23	0.916	0.952	0.934	0.868
	Gemma-2-9B	5, 10, 15, 20, 25, 30, 35, 40	0.950	0.924	0.937	0.874
	Llama-3.1-8B	1, 2, 3, 4, 5, 6, 8, 10	0.801	0.995	0.898	0.796

B.3 Causal effect of the decoder write

The judge confirms that semantic features carry coherent token-level meaning at the projection level. We now investigate whether removing such a feature influences the model's decision on those tokens, using the intervention primitive defined in Equation (6).

Setup. For each sampled SAE layer, we draw 100 features that pass all three semantic criteria and have a decoder norm above the per-layer 5th percentile, ensuring the comparison is not dominated by

weak features. Each flagged feature f is paired with a *matched-norm baseline* f_b drawn from the same layer: f_b fails all three criteria (strict non-semantic), has decoder norm within $\pm 5\%$ of f's, and fires on the same prompt at the same position. The baseline therefore removes a non-semantic write of similar size to f's, on the same forward pass, at the same residual-stream position. Each flagged feature is tested on its top max-activating prompt from Neuronpedia [9], truncated at the position where the feature peaks, so the last sequence position is exactly where the feature fires. We then read the next-token logits at the feature's top-1 logit-lens token under a) the clean forward pass and b) the intervened forward pass with $\mathrm{do}(z = 0)$ applied to that one feature inside the SAE. The reported metrics are $|\Delta\mathrm{logit}|$ at the top-1 target and the full-vocabulary KL divergence between the clean and intervened next-token distributions.

Result. Table 4 shows the 16 sweeps. The flagged feature's removal moves the target token's logit by 0.8–7.1 (median, log-space units), against 0.02–0.12 for its matched-norm non-semantic neighbour at the same layer on the same prompt. The win-rate is the fraction of pairs in which the flagged feature produces a strictly larger $|\Delta\mathrm{logit}|$ than its baseline. It ranges from 0.88 to 1.00 across the 16 sweeps. The KL effect tracks $|\Delta\mathrm{logit}|$ but at much larger ratios: median full-vocabulary KL is 0.02–0.38 for flagged removals and $1{\times}10^{-4}$–$5{\times}10^{-3}$ for matched removals. The stark comparison clearly demonstrates the causal downstream effects of the features identified by our method on their semantic tokens.

Table 4: Intervention validation for Experiment 1. For each sampled SAE layer we draw 100 flagged features and pair each with a matched-norm baseline: a strict-non-semantic feature whose decoder norm is within $\pm 5\%$ of the flagged feature's and that fires on the same prompt. We apply $\mathrm{do}(z = 0)$ to each and read the change in the next-token logit at the flagged feature's top-1 logit-lens token. KL is over the full vocabulary. n_{m} is the number of pairs for which a matched baseline was available.

Model	Layer	n_{m}	Median $\lvert\Delta\mathrm{logit}\rvert$		Median KL		Win-rate
			semantic	matched	semantic	matched	$\lvert\Delta\rvert$
Gemma-2-9B	5	89	1.359	0.0391	0.0359	0.0001	0.92
	10	99	0.988	0.0156	0.0156	0.0001	0.96
	15	99	1.187	0.0312	0.0310	0.0001	0.92
	20	93	0.818	0.0391	0.0230	0.0002	0.97
	25	97	1.227	0.0469	0.0518	0.0003	0.97
	30	96	2.133	0.0469	0.1012	0.0003	0.95
	35	93	3.461	0.0469	0.2306	0.0003	1.00
	40	83	7.129	0.0312	0.3782	0.0005	1.00
Llama-3.1-8B	16	92	1.212	0.1094	0.0677	0.0035	0.92
	18	80	1.879	0.1055	0.1302	0.0023	0.96
	20	79	2.253	0.0781	0.1833	0.0034	0.97
	22	65	2.926	0.0938	0.3066	0.0032	0.97
	24	62	3.744	0.1035	0.1918	0.0048	0.95
	26	57	3.596	0.1250	0.2571	0.0023	0.96
	28	54	4.023	0.1035	0.2206	0.0019	0.94
	30	51	4.643	0.0938	0.1227	0.0017	0.88

C Feature-Level Attention Analysis

We now formalize the relationship between our weight-based feature-feature attention scores and the actual attention computations in the forward pass.

In our experiments, we apply normalization to individual feature contributions, following the convention used in prior work employing the logit lens. In contrast, during the model's actual forward pass, normalization operates on the full combination of features prior to attention computation. Applying normalization at the level of individual features is therefore an approximation, since normalization is a non-linear operation. For analytical clarity, the derivation below omits normalization.

C.1 SAE Reconstruction of the Residual Stream

In the actual forward pass, the residual stream at the end of layer L is:

$$\boldsymbol{x}_{\text{true}} = \boldsymbol{x}_{\text{SAE}} + \boldsymbol{x}_{\text{error}} \tag{24}$$

where $\boldsymbol{x}_{\text{SAE}}$ is the SAE reconstruction and $\boldsymbol{x}_{\text{error}}$ is the reconstruction error. Our analysis focuses on $\boldsymbol{x}_{\text{SAE}}$, which for well-trained SAEs captures the dominant systematic structure.

When features activate with strengths $z_1, z_2, \ldots, z_{d_{\text{sae}}}$, the SAE reconstruction is:

$$\boldsymbol{x}_{\text{SAE}} = \sum_{i=1}^{d_{\text{sae}}} z_i \boldsymbol{W}_{\text{dec},L}^{(i)} + \boldsymbol{b}_{\text{dec}} \tag{25}$$

C.2 Attention Computation in Layer $L+1$

For attention head h in layer $L+1$, under our approximation where normalization effects are set aside, the query and key projections are:

$$\boldsymbol{q}^{(h)} = \boldsymbol{x}_{\text{SAE}} \boldsymbol{W}_{Q,L+1}^{(h)} = \sum_{i=1}^{d_{\text{sae}}} z_i \boldsymbol{W}_{\text{dec},L}^{(i)} \boldsymbol{W}_{Q,L+1}^{(h)} + \boldsymbol{b}_{\text{dec}} \boldsymbol{W}_{Q,L+1}^{(h)} \tag{26}$$

$$\boldsymbol{k}^{(h)} = \boldsymbol{x}_{\text{SAE}} \boldsymbol{W}_{K,L+1}^{(h)} = \sum_{i=1}^{d_{\text{sae}}} z_i \boldsymbol{W}_{\text{dec},L}^{(i)} \boldsymbol{W}_{K,L+1}^{(h)} + \boldsymbol{b}_{\text{dec}} \boldsymbol{W}_{K,L+1}^{(h)} \tag{27}$$

Defining the per-feature query and key vectors:

$$\boldsymbol{q}_i^{(h)} = \boldsymbol{W}_{\text{dec},L}^{(i)} \boldsymbol{W}_{Q,L+1}^{(h)}, \quad \boldsymbol{k}_i^{(h)} = \boldsymbol{W}_{\text{dec},L}^{(i)} \boldsymbol{W}_{K,L+1}^{(h)} \tag{28}$$

and the bias contributions:

$$\boldsymbol{q}_b^{(h)} = \boldsymbol{b}_{\text{dec}} \boldsymbol{W}_{Q,L+1}^{(h)}, \quad \boldsymbol{k}_b^{(h)} = \boldsymbol{b}_{\text{dec}} \boldsymbol{W}_{K,L+1}^{(h)} \tag{29}$$

we can rewrite the projections as:

$$\boldsymbol{q}^{(h)} = \sum_{i=1}^{d_{\text{sae}}} z_i \boldsymbol{q}_i^{(h)} + \boldsymbol{q}_b^{(h)}, \quad \boldsymbol{k}^{(h)} = \sum_{j=1}^{d_{\text{sae}}} z_j \boldsymbol{k}_j^{(h)} + \boldsymbol{k}_b^{(h)} \tag{30}$$

C.3 Decomposition of Attention Scores

The pre-softmax attention score in head h is the scaled dot product:

$$s^{(h)} = \frac{\boldsymbol{q}^{(h)} \cdot \boldsymbol{k}^{(h)}}{\sqrt{d_{\text{head}}}} \tag{31}$$

$$= \frac{1}{\sqrt{d_{\text{head}}}} \left(\sum_{i=1}^{d_{\text{sae}}} \sum_{j=1}^{d_{\text{sae}}} z_i z_j (\boldsymbol{q}_i^{(h)} \cdot \boldsymbol{k}_j^{(h)}) + \sum_{i=1}^{d_{\text{sae}}} z_i (\boldsymbol{q}_i^{(h)} \cdot \boldsymbol{k}_b^{(h)}) \right.$$
$$\left. + \sum_{j=1}^{d_{\text{sae}}} z_j (\boldsymbol{q}_b^{(h)} \cdot \boldsymbol{k}_j^{(h)}) + \boldsymbol{q}_b^{(h)} \cdot \boldsymbol{k}_b^{(h)} \right) \tag{32}$$

The most interesting term is the double sum over feature pairs. Defining the per-feature pairwise attention potential:

$$s_{i,j}^{(h)} = \frac{\boldsymbol{q}_i^{(h)} \cdot \boldsymbol{k}_j^{(h)}}{\sqrt{d_{\text{head}}}} \tag{33}$$

the forward pass attention score decomposes as:

$$s^{(h)} = \sum_{i=1}^{d_{\text{sae}}} \sum_{j=1}^{d_{\text{sae}}} z_i z_j \, s_{i,j}^{(h)} + (\text{bias terms}) \tag{34}$$

This decomposition shows that the pre-softmax matrix $\boldsymbol{S}^{(h)} = [s_{i,j}^{(h)}]$ captures the weight-based structural potentials that, when weighted by actual feature activations $z_i z_j$, determine the attention scores in the forward pass (up to the normalization approximation discussed above).

D Framework clarifications

This appendix expands on the three modeling choices flagged at the end of Section 3: per-feature normalization, the encoder–decoder asymmetry that motivates our paired projections, and the decoder-side scope of the do-intervention.

D.1 Per-feature normalization preserves rank ordering

The projections in Equations (3) to (5) apply $\mathrm{FinalNorm}$ or $\mathrm{AttnNorm}$ to a single decoder row in isolation, whereas in the model's forward pass these layer normalizations act on the full residual-stream vector at a token position. Because the in-context norm depends on every other component present in the stream, per-feature normalization is necessarily an approximation. Quantifying this approximation against the in-context norm directly would require running inference across a corpus and tracking feature co-activation, which moves outside the weight-based premise of the framework. We instead address the concern that motivates the question: whether the approximation distorts the *ordering* of features, since rank ordering is what every claim in the paper relies on.

We compare the per-feature normalized attention scores against an unnormalized baseline across 128 heads spanning 8 evenly sampled layers (layers 5–40) of Gemma-2-9B. The mean Spearman rank correlation between the two is $0.902 \leq \bar{\rho} \leq 0.985$ for queries and $0.876 \leq \bar{\rho} \leq 0.980$ for keys (Figure 10); the per-head distributions sit tightly above 0.85 at every layer for both query (row-sum) and key (column-sum) aggregates. All architectural findings reported in Sections 5 and 6 are qualitatively identical under both normalization schemes. The per-feature approximation therefore preserves the essential rank structure that our analyses exploit.

D.2 Encoder–decoder asymmetry justifies the paired projections

A natural concern about Equations (2) and (3) is that, in a model with tied embeddings, $\boldsymbol{W}_{\text{enc}}^{(\cdot,i)}\boldsymbol{W}_{\text{E}}^{\top}$ and $\boldsymbol{W}_{\text{dec}}^{(i)}\boldsymbol{W}_{\text{U}}$ would carry the same information whenever the encoder column and decoder row of feature i are themselves aligned, making the paired analysis redundant. Whether this redundancy holds is therefore an empirical question about the SAE itself, not about the host model's weight tying.

We measure the per-feature cosine similarity between $\boldsymbol{W}_{\text{enc}}^{(\cdot,i)}$ and $\boldsymbol{W}_{\text{dec}}^{(i)}$ across layers of the Gemma Scope SAEs on Gemma-2-9B and the Llama Scope SAEs on Llama-3.1-8B (Figure 11). On Gemma-2-9B the layer-wise mean is approximately 0.8, with early layers dropping to ≈ 0.55. Llama-3.1-8B sits substantially lower across the entire depth, with mean per-feature cosine well below the Gemma values, so the encoder and decoder vectors of the same Llama Scope feature are even further from being interchangeable. This bimodality between the encoder and decoder of a feature has been documented independently [23]. The two sides therefore carry different signal in both architectures, and Section 4 confirms that each projection surfaces a population the other does not.

D.3 The do-intervention is decoder-side by construction

The intervention $\mathrm{do}(z_i = 0)$ in Equation (6) zeros the SAE feature and re-decodes, equivalently subtracting $z_i \boldsymbol{W}_{\text{dec},L}^{(i)}$ from the residual stream. What it removes is the decoder write, so it is a counterfactual test of claims that route through the decoder row: the decoder–unembedding projection (Equation (3)) and the decoder–QK projections (Equations (4) and (5)). Sections 4 and 5 use it for exactly these claims.

The encoder–embedding projection in Equation (2) is structurally different: it predicts *which inputs activate feature i*, not what feature i does once active. A read-side prediction of this form is supported by the structural projection together with the activation evidence on which it relies, but it is not the kind of claim that $\mathrm{do}(z_i = 0)$ can validate, since zeroing the activation removes the write rather than testing the read. We therefore do not attempt to force the encoder side through the same do-operation, and we treat the encoder–embedding result in Section 4 as a structural prediction with the LLM-judge agreement check serving as its independent confirmation.

E Rationale for Attention Specialization Metrics

The decision to analyze computational features from both a query-based and a key-based perspective is motivated by the inherent asymmetry of the attention mechanism. The row-wise application of the softmax function imparts fundamentally different properties and, consequently, different computational roles to features acting as queries versus those acting as keys.

E.1 The Asymmetry of Query and Key

The attention score from a query feature i to a key feature j is calculated as part of a distribution over all possible keys: $\boldsymbol{a}_i^{(h)} = \text{softmax}(\boldsymbol{s}_i^{(h)})$. This means that for a given query i, all keys $j \in \{1, \ldots, d_{\text{sae}}\}$ are in direct competition to receive its attention.

Query-based (Outgoing) Perspective A query feature must allocate a fixed budget of attention (summing to 1.0) across all possible keys. To achieve a high score for a specific key, it only needs to have a pre-softmax alignment with that key that is significantly higher than its alignment with any other key. This identifies features that are specialized "lookers" or "actors," designed to perform a targeted information lookup.

Key-based (Incoming) Perspective A key feature's incoming attention is the sum of attention it receives from many independent, row-wise softmax operations. There is no column-wise competition. For a key to be considered "dominant," it must consistently win the attention competition across many different queries. This identifies features that represent "information hubs": concepts that are so useful that multiple computational circuits attend to them.

E.2 Justification of Metric Choices

This asymmetry necessitates different metrics for each perspective:

For query specialists: We use *sum* of top-k > 0.95.

This metric is ideal for identifying "specialist lookers." It asks: "Does this feature concentrate nearly all of its attention budget on a very small set of targets?" This is a very intuitive and powerful definition of a computationally specialized feature. It directly identifies features that participate in targeted circuits, capturing the essence of "doing something specific" in attention.

For key hubs: We use *mean* of top-k > 0.25.

This is a much stricter criterion designed to identify "information hubs." A simple sum could be inflated by one query giving 0.95 attention and nine others giving almost none. The mean, however, requires that a feature be consistently and substantially important to multiple independent queries (e.g., each of its top-10 admirers dedicates, on average, >25% of their attention to it). This filters for features that are important to the model's computations, for which a lower pass rate is expected and informative.

We chose $k = 10$ primarily as a compromise between retaining sufficient information and managing the computational budget required for all features across all layers and models, since each column or row stores d_{sae} scores for both pre- and post-softmax matrices during exploration.

Given that these thresholds may be somewhat arbitrary, we also provide several alternative values for τ_Q and τ_K for reference in Table 5 and Table 6, respectively.

Table 5: Average percentage of features qualifying as query specialists across all layers for a range of thresholds (τ_Q). The criterion requires the sum of a feature's top-10 post-softmax attention scores to exceed the threshold. The highlighted column corresponds to our primary choice.

Threshold τ_Q	0.95	0.90	0.75	0.50	0.25
Gemma-2-2B	39.54%	43.79%	50.94%	59.29%	68.98%
Gemma-2-9B	53.82%	58.49%	65.60%	72.84%	79.73%
Llama-3.1-8B	9.75%	11.71%	15.62%	21.14%	28.83%

Table 6: Average percentage of features qualifying as key hubs across all layers for a range of thresholds (τ_K). The criterion requires the mean of a feature's top-10 incoming post-softmax attention scores to exceed the threshold. The highlighted column corresponds to our primary choice.

Threshold τ_K	0.50	0.25	0.10	0.05	0.01
Gemma-2-2B	1.49%	2.90%	8.88%	14.16%	25.82%
Gemma-2-9B	3.16%	6.15%	17.52%	27.31%	43.59%
Llama-3.1-8B	0.42%	0.85%	2.15%	3.57%	8.03%

F Validating Computational Specialization: Intervention

This appendix tests, with the same $\mathrm{do}(z_i = 0)$ primitive used in Section B.3, whether features flagged as query specialists or key hubs in Section 5 actually shape the QK computation of the next attention layer. The question is mirrored to each role: for a query specialist at layer L firing at position p, removing its decoder write should perturb the post-softmax attention row at $(L+1, h^\star, p)$; for a key hub at p, removal should reduce the attention that later queries ($q > p$) place on p at $(L+1, h^\star)$. The structurally predicted head $h^\star$ is the head that exceeds the role's threshold; ties broken by the role-specific score.

The two roles have asymmetric prompt requirements that follow directly from where they sit in the attention computation. A query specialist participates in attention as soon as it fires: its decoder write enters the residual stream at position p, and at the very next layer that residual is read as a query at the same position p, so the attention behaviour we want to inspect already exists at the firing token. We therefore truncate the prompt at p and feed the model the prefix that ends exactly where the feature peaks, which is also the convention used for the previous intervention experiment (Section B.3). A key hub, by contrast, only matters for queries that come *after* the firing position: its decoder write puts a key vector at position p that subsequent positions can attend back to. Truncating at p would leave no later queries to read the hub. We therefore keep the prompt un-truncated for key hubs so the firing position p sits mid-sequence and the queries at $q > p$ are present to react to the intervention.

F.1 Setup

Sampling. For each (model, role, layer) cell we draw 100 flagged features and pair each with a *matched-norm baseline*: a strict-non-specialist feature whose decoder norm is within $\pm 5\%$ of the flagged feature's and that fires on the same prompt at the same position. *Strict-non-specialist* means failing both criteria $\sum_{\text{top-10}} \boldsymbol{A}^{(h)}_{i,\cdot} > \tau_Q$ and $\frac{1}{10}\sum_{\text{top-10}} \boldsymbol{A}^{(h)}_{\cdot,j} > \tau_K$ at *every* head.

Prompt construction. For each flagged feature we fetch its top max-activating prompt from Neuronpedia [9]. For query specialists the prompt is truncated at the firing position so the last token fed to the model is exactly the firing token. For key hubs the prompt is kept un-truncated so the firing position p sits mid-sequence and later queries can attend back to it. The flagged head $h^\star$ is the argmax over heads of the role-specific specialist score on the flagged feature.

Metric. For query specialists we report the KL divergence between the clean and intervened post-softmax attention rows at $(L+1, h^\star, p)$ over keys $j \in [0, p]$. For key hubs we report the change in mean incoming attention from later queries, $\Delta = \frac{1}{|Q|}\sum_{q \in Q}\big(\boldsymbol{A}^{(h^\star)}_{q,p,\text{clean}} - \boldsymbol{A}^{(h^\star)}_{q,p,\text{intervened}}\big)$ with $Q = \{q : q > p\}$. KL is non-negative; Δ is signed, and a positive value matches the structural prediction that the hub was a target the queries concentrated on.

We sample 8 SAE layers per model, evenly spaced across depth (Gemma-2-9B: $\{5, 10, \ldots, 40\}$; Llama-3.1-8B: $\{16, 18, \ldots, 30\}$, plus $\{10, 14\}$ for query specialists where the flagged pool remains populated). Llama is constrained to $L \geq 14$ for key hubs because at $L < 13$ the hub pool is too thin to support $n = 100$ with matched-norm baselines.

F.2 Query specialists: causal effect on the attention row

Table 7 reports the 18 sweeps (8 layers in Gemma plus 10 in Llama). Removing a flagged query specialist shifts its attention row by a median KL of 0.010–0.246 across all sweeps, against 4×10^{-5}–1.7×10^{-3} for the matched-norm baseline firing on the same prompt. The flagged-to-matched ratio (per-pair median, not the ratio of the two median columns) is $43\times$ to $2161\times$ across cells, with a typical value near $100\times$. The win-rate, the fraction of pairs in which the flagged feature produces a strictly larger KL than its matched baseline, is 0.93–1.00 in every cell, and reaches 1.00 in 6 of the 18.

The KL effect is robust across depth in both models. Gemma's median KL is largest at the shallowest sampled layer and stays in $[0.010, 0.056]$ for $L \geq 10$. Llama's median KL is consistently in $[0.012, 0.060]$ across $L \in [14, 30]$ with a single early peak at $L = 10$ (0.181).

Table 7: Intervention validation for Experiment 2, query specialists. For each sampled SAE layer we draw 100 flagged query specialists and pair each with a matched-norm baseline: a strict-non-specialist feature whose decoder norm is within $\pm5\%$ of the flagged feature's and that fires on the same prompt at the same position. We apply $\mathrm{do}(z = 0)$ to each and read the KL divergence between clean and intervened post-softmax attention rows of the flagged head at the firing position. n_m is the number of pairs for which a matched baseline was available.

Model	Layer	n_m	Median KL		Win-rate
			flagged	matched	flagged $>$ matched
Gemma-2-9B	5	96	0.2456	8.21×10^{-5}	1.00
	10	99	0.05559	5.67×10^{-5}	0.98
	15	100	0.05594	8.50×10^{-5}	1.00
	20	95	0.04033	2.94×10^{-4}	0.99
	25	96	0.03372	1.35×10^{-4}	1.00
	30	98	0.02398	6.36×10^{-5}	1.00
	35	99	0.01032	4.30×10^{-5}	1.00
	40	99	0.02766	4.85×10^{-5}	0.98
Llama-3.1-8B	10	95	0.1811	1.68×10^{-3}	0.99
	14	90	0.01206	1.72×10^{-4}	1.00
	16	96	0.04495	5.81×10^{-4}	0.97
	18	98	0.04454	5.44×10^{-4}	0.97
	20	99	0.05926	6.59×10^{-4}	0.97
	22	95	0.03351	4.84×10^{-4}	0.97
	24	95	0.01196	2.53×10^{-4}	0.97
	26	96	0.01617	5.20×10^{-4}	0.99
	28	89	0.02982	5.23×10^{-4}	0.93
	30	90	0.01197	2.27×10^{-4}	0.98

F.3 Key hubs: causal effect on incoming attention

Table 8 reports the results for key hubs. The headline magnitude is $|\Delta|$, the absolute change in mean incoming attention from later queries. Median $|\Delta|$ for flagged hubs is $\sim10^{-3}$ across cells (Gemma: 8.4×10^{-4}–2.5×10^{-3}; Llama: 1.1×10^{-4}–2.9×10^{-3}), against $\sim10^{-5}$ for most matched baselines. These magnitudes are intrinsically small in absolute terms because the metric averages over many later queries (median $|Q| \in [50, 80]$). The cross-row scale is therefore not directly comparable to the query KL; the relevant comparison is the ratio against the matched-norm baseline. That ratio (again, per-pair median) is $9\times$ to $37\times$ across all 17 sweeps, and the win-rate against matched is 0.88–0.98 in every cell.

Taken together, query and key interventions deliver a causal counterpart to the projection-based criteria of Section 5. A feature flagged by the post-softmax thresholds is, with very high consistency, the feature that actually steers the next layer's attention at the position where it fires; a non-specialist of equal decoder norm firing on the same prompt produces an effect one to three orders of magnitude smaller.

Table 8: Intervention validation for Experiment 2, key hubs. For each sampled SAE layer we draw 100 flagged key hubs and pair each with a matched-norm baseline. We apply $\mathrm{do}(z=0)$ at the hub's firing position and measure the change in mean incoming attention from later queries ($q > p$) at the flagged head. Magnitude columns report $|\Delta|$ of the drop.

Model	Layer	n_m	Median |drop| flagged	Median |drop| matched	Win-rate flagged > matched
Gemma-2-9B	5	98	1.705×10^{-3}	1.02×10^{-4}	0.94
	10	93	2.504×10^{-3}	6.22×10^{-5}	0.98
	15	99	1.058×10^{-3}	4.21×10^{-5}	0.97
	20	93	2.026×10^{-3}	2.33×10^{-4}	0.94
	25	92	1.570×10^{-3}	1.29×10^{-4}	0.92
	30	93	1.048×10^{-3}	4.61×10^{-5}	0.90
	35	89	8.436×10^{-4}	2.77×10^{-5}	0.98
	40	98	1.097×10^{-3}	4.08×10^{-5}	0.98
Llama-3.1-8B	14	92	1.101×10^{-4}	8.61×10^{-6}	0.92
	16	98	2.447×10^{-3}	1.76×10^{-4}	0.92
	18	96	7.716×10^{-4}	6.07×10^{-5}	0.95
	20	94	2.654×10^{-3}	1.95×10^{-4}	0.88
	22	93	2.352×10^{-3}	1.08×10^{-4}	0.97
	24	96	1.437×10^{-3}	1.44×10^{-4}	0.95
	26	96	2.852×10^{-3}	9.16×10^{-5}	0.92
	28	89	1.118×10^{-3}	6.96×10^{-5}	0.91
	30	93	1.146×10^{-3}	7.81×10^{-5}	0.94

G Analysis of Lower Computational Feature Pass Rates in Llama-3.1-8B

A notable finding from our computational analysis is the lower pass rate for both query-specialists and key-hubs in Llama-3.1-8B compared to the Gemma-2 models (e.g., 9.75% vs. 53.82% for query-specialists with $\tau_Q = 0.95$). We hypothesize this discrepancy arises from several interacting factors related to model and SAE architecture, as well as the nature of our out-of-context methodology.

G.1 SAE dictionary size

The most direct architectural difference is the SAE dictionary size: 32,768 for Llama-3.1-8B versus 16,384 for Gemma-2-9B. This has two potential consequences: A larger dictionary size naturally leads to a lower percentage pass rate, even if the absolute number of specialized features for a given function were similar across models.

G.2 Feature Splitting

More importantly, a larger dictionary may encourage the SAE to learn a more distributed or compositional code [24]. Instead of representing a concept with a single feature, it might reconstruct the same activation using a linear combination of several more granular features. Our methodology, which evaluates each feature's decoder vector in isolation, would register a diluted signal in such cases, as no single feature vector carries the full computational weight of the concept. Consequently, fewer individual features would meet the dominance thresholds.

We can formalize how feature splitting would lead to lower measured scores for individual features. For simplicity, we ignore normalizations and reconstruction coefficients.

Let $\boldsymbol{x} \in \mathbb{R}^{d_{\text{model}}}$ be a target vector in the residual stream that represents a specific computational role. An ideal SAE might learn a single feature, say feature k, to represent this vector. In this scenario, the feature's decoder vector is proportional to the target vector:

$$\boldsymbol{W}_{\text{dec}}^{(k)} = \boldsymbol{x} \tag{35}$$

Our methodology measures this feature's participation in a QK circuit by projecting it. For example, its query vector for head h would be:

$$\boldsymbol{q}_k^{(h)} = \boldsymbol{W}_{\text{dec}}^{(k)} \boldsymbol{W}_Q^{(h)} = \boldsymbol{x} \boldsymbol{W}_Q^{(h)} \tag{36}$$

The resulting pre-softmax attention scores are directly proportional to the full magnitude of $\boldsymbol{x}$.

Now, consider an SAE with a larger dictionary. It might learn to represent the same vector $\boldsymbol{x}$ using a linear combination of two or more features. Let's say features m and n combine to form $\boldsymbol{x}$:

$$\boldsymbol{x} = \boldsymbol{W}_{\text{dec}}^{(m)} + \boldsymbol{W}_{\text{dec}}^{(n)} \tag{37}$$

A simple solution for the SAE optimization is to distribute the representation, for instance:

$$\boldsymbol{W}_{\text{dec}}^{(m)} = \alpha \boldsymbol{x} \quad \text{and} \quad \boldsymbol{W}_{\text{dec}}^{(n)} = (1-\alpha)\boldsymbol{x} \tag{38}$$

for some fraction $\alpha \in (0, 1)$.

Our methodology analyzes each feature in isolation. When we evaluate feature m, its query vector is:

$$\boldsymbol{q}_m^{(h)} = \boldsymbol{W}_{\text{dec}}^{(m)} \boldsymbol{W}_Q^{(h)} = (\alpha \boldsymbol{x}) \boldsymbol{W}_Q^{(h)} = \alpha (\boldsymbol{x} \boldsymbol{W}_Q^{(h)}) = \alpha \boldsymbol{q}_k^{(h)} \tag{39}$$

The query vector for the split feature m is a scaled-down version of the query vector for the ideal single feature k. Consequently, its pre-softmax attention score with any key feature j is also scaled down:

$$s_{m,j}^{(h)} = \frac{\boldsymbol{q}_m^{(h)} \cdot \boldsymbol{k}_j^{(h)}}{\sqrt{d_{\text{head}}}} = \alpha \left(\frac{\boldsymbol{q}_k^{(h)} \cdot \boldsymbol{k}_j^{(h)}}{\sqrt{d_{\text{head}}}} \right) = \alpha s_{k,j}^{(h)} \tag{40}$$

The measured score for the individual split feature is only a fraction of the score it would have if it represented the concept alone. This makes it significantly less likely for any single feature to pass a fixed, high dominance threshold like τ_q or τ_k.

G.3 Strictness of out-of-context competition

Our out-of-context framework is inherently strict, as it simulates attention competition across the entire SAE dictionary without the benefit of contextual sparsity from a real forward pass. In an actual inference step, only a few hundred features might be active, and the softmax competition would occur among this small set. Our method pits one feature against all others in the dictionary. This effect is likely magnified for Llama-3.1's larger dictionary; the bar for a single feature to dominate a competition against 32,767 others is substantially higher than against 16,383, potentially contributing to its lower pass rate.

G.4 Focus on intra-layer interactions

Our analysis focuses exclusively on intra-layer attention, where features from layer L interact with the QK circuits of the immediately subsequent layer, $L + 1$. It is plausible that some features are specialized for longer-range, inter-layer interactions, for example, by being preserved in the residual stream to be primarily used by attention heads in layers $L + 2$ or beyond. Such long-range specialists would not be captured by our current methodology. While this is a limitation common to our analysis of all models, architectural differences could lead to Llama relying more heavily on such mechanisms, which would not be reflected in our reported scores.

H Additional Results

This appendix presents additional results that were omitted from the main text for brevity.

Table 9: Thresholds for semantic feature classification, set at the 50th percentile of scores from a representative layer. The arrows indicate the desired direction for each metric.

Analysis Type	Model	Layer	Levenshtein Sim. ↑	Cosine Sim. ↑	Top-100 Entropy ↓
Decoder-Unembedding	Gemma-2-2B	16	0.157385	0.292363	0.612010
	Gemma-2-9B	27	0.194815	0.332252	0.550316
	Llama-3.1-8B	20	0.154048	0.092486	2.911260
Encoder-Embedding	Gemma-2-2B	3	0.149182	0.282485	4.604819
	Gemma-2-9B	6	0.153151	0.348118	4.604956
	Llama-3.1-8B	3	0.143704	0.028160	4.597589

LLM-judge prompt (top-k=5)

System:
You are a careful linguistic analyst. You will be given a list of tokens (words, subwords, or symbols) drawn from a language model's vocabulary. Decide whether the tokens collectively point to one coherent meaning.
A coherent meaning includes:
- one word and its morphological variants (e.g., 'run', 'runs', 'running');
- the same word across casing or scripts ('Cat', 'CAT', 'cat');
- the same word across languages ('cat', 'gato', 'chat', 'U+732B');
- a tight semantic family (e.g., kitchen appliances, planets, basketball terms).
It is NOT coherent when the list is dominated by:
- random unrelated words across topics;
- generic high-frequency vocabulary ('the', 'and', 'a');
- punctuation, code symbols, or fragments with no shared meaning;
- tokens that only share a few letters but mean different things.
Answer only with 'yes' or 'no'.

User:
Tokens: `" Pull", "Pull", " PULL", " pull", " pulled"`
Do these tokens collectively express one coherent meaning? Answer 'yes' or 'no'.

Assistant:
yes

User:
Tokens: `" Alaska", "Alaska", " Alaskan", " Anchorage", " Juneau"`
Do these tokens collectively express one coherent meaning? Answer 'yes' or 'no'.

Assistant:
yes

User:
Tokens: `" red", "red", " blue", " green", " yellow"`
Do these tokens collectively express one coherent meaning? Answer 'yes' or 'no'.

Assistant:
yes

User:
Tokens: `" the", "-", " and", ".", " a"`
Do these tokens collectively express one coherent meaning? Answer 'yes' or 'no'.

Assistant:
no

User:
Tokens: `"keur", "éu", " protoimpl", "StoreMessageInfo", "neo"`
Do these tokens collectively express one coherent meaning? Answer 'yes' or 'no'.

Assistant:
no

User:
Tokens: `" auf", " Bedeutung", " carrots", " ngày", " ☘"`
Do these tokens collectively express one coherent meaning? Answer 'yes' or 'no'.

Assistant:
no

User:
Tokens: `"<token 1>", "<token 2>", "<token 3>", "<token 4>", "<token 5>"`
Do these tokens collectively express one coherent meaning? Answer 'yes' or 'no'.

Figure 9: LLM-judge prompt for the semanticity check. The system message defines what counts as a coherent token cluster, six few-shot exemplars (three positive, three negative) anchor the decision boundary, and the final user turn presents the feature's top-5 logit-lens tokens for verdict. The judge is restricted to {yes, no} at temperature 0. Tokens are rendered with their leading-space markers preserved so the judge sees the same surface form as the model's vocabulary.

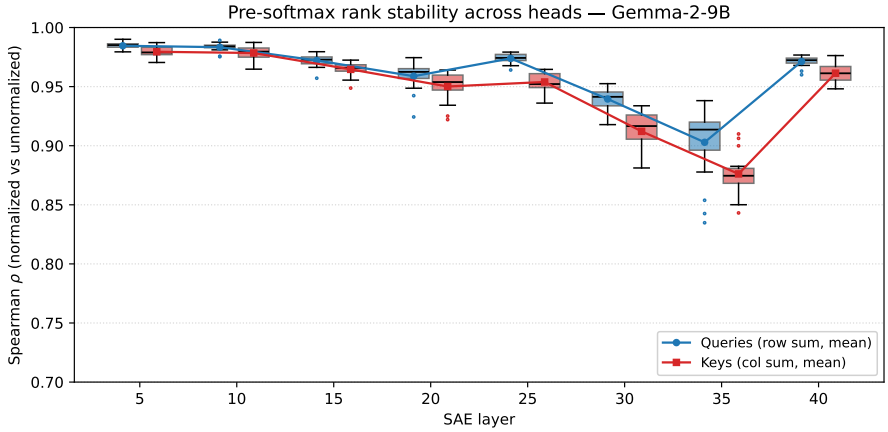


Figure 10: Per-layer Spearman rank correlation between per-feature-normalized and unnormalized pre-softmax aggregate scores, across all attention heads in 8 sampled layers of Gemma-2-9B. Boxes show the per-head distribution; lines join the per-layer means. Both query (row-sum) and key (column-sum) aggregates remain above $\bar{\rho} \approx 0.88$ at every depth, confirming that the per-feature normalization preserves the rank ordering on which our claims depend.

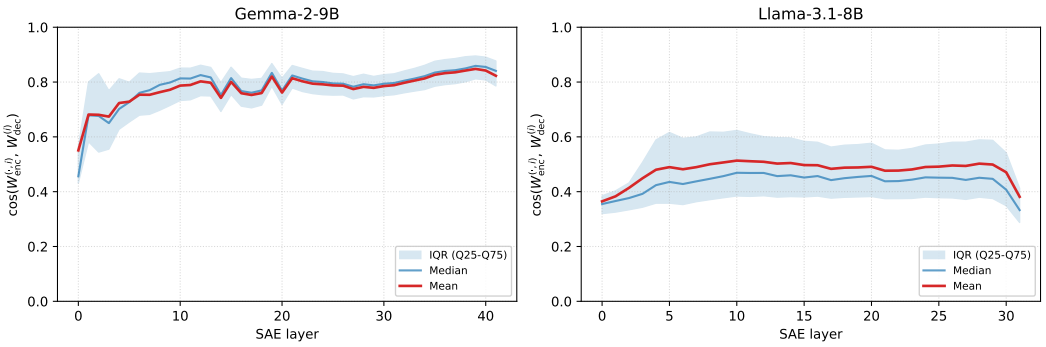


Figure 11: Per-layer cosine similarity between the encoder column and decoder row of the same SAE feature, for the Gemma Scope SAEs on Gemma-2-9B (left) and the Llama Scope SAEs on Llama-3.1-8B (right). Solid lines are the per-layer mean (red) and median (blue) across features; the shaded band is the inter-quartile range. Gemma sits around 0.8 on average with a dip in early layers; Llama sits substantially lower throughout, making the paired encoder–embedding and decoder–unembedding analyses non-redundant by an even wider margin.

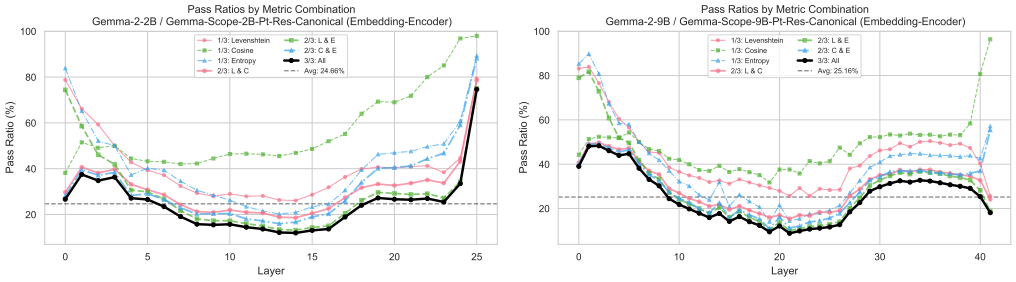


Figure 12: Supplementary results for Experiment 1 on Gemma-2-2B and 9B, using encoder–embedding alignment. Semantic features display a U-shaped distribution, with an average joint pass rate of 24.66% and 25.16%, respectively.

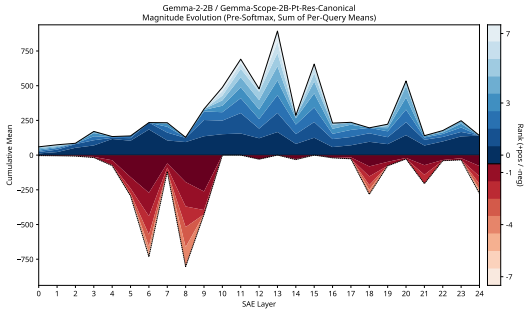


Figure 13: Main results for Experiment 2 on Gemma-2-2B: Magnitude of attention participation across layers, with the total height at each layer representing the aggregate pre-softmax score. These scores are calculated as the mean across all keys for each query feature, then summed for each head. The stacking direction of the head values is determined based on whether they are positive or negative. Each color represents a rank of magnitude, rather than a specific head index, across all layers.

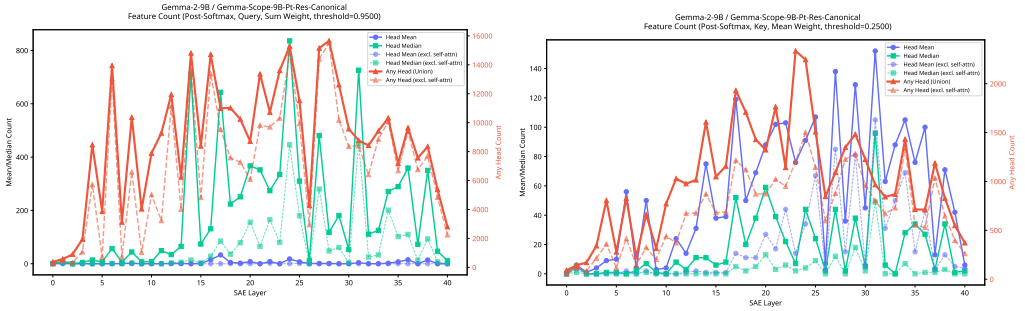


Figure 14: Feature count trends across Gemma-2-9B layers showing features with top-10 query-based post-softmax attention weight sum above 0.95 (left) and key-based attention weight mean above 0.25 (right). Note that there are two y-axes of different magnitudes: the left y-axis is for the mean and median counts across heads, while the right y-axis is for the union across all heads ("any head"). Solid lines include weights of self-attention; dashed lines exclude self-attention.

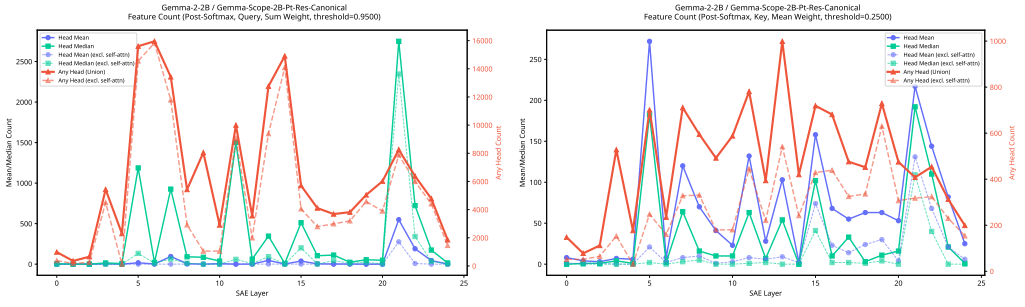


Figure 15: Feature count trends across Gemma-2-2B layers showing features with top-10 query-based post-softmax attention weight sum above 0.95 (left) and key-based attention weight mean above 0.25 (right). Note that there are two y-axes of different magnitudes: the left y-axis is for the mean and median counts across heads, while the right y-axis is for the union across all heads ("any head"). Solid lines include weights of self-attention; dashed lines exclude self-attention.

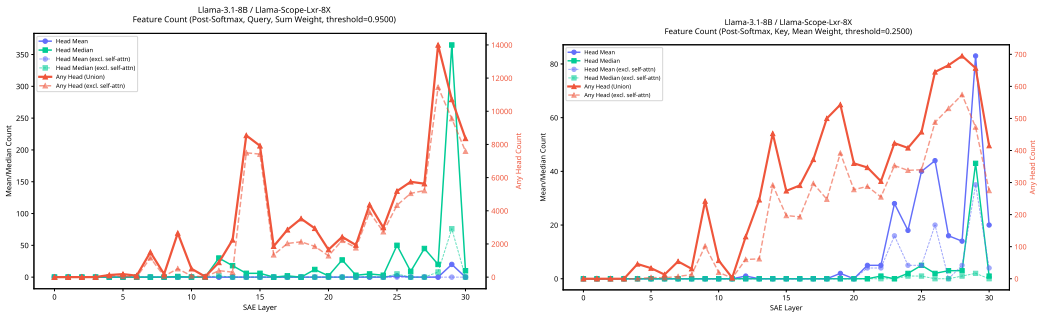
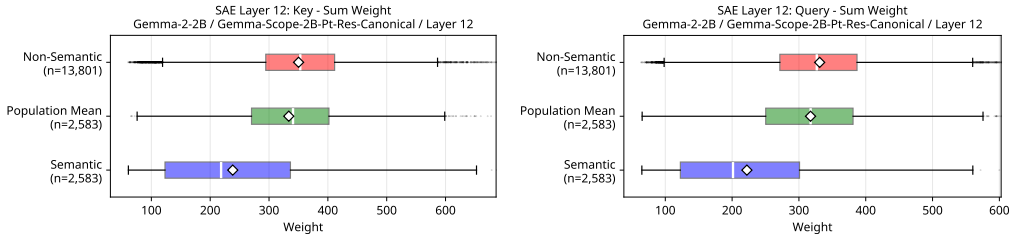
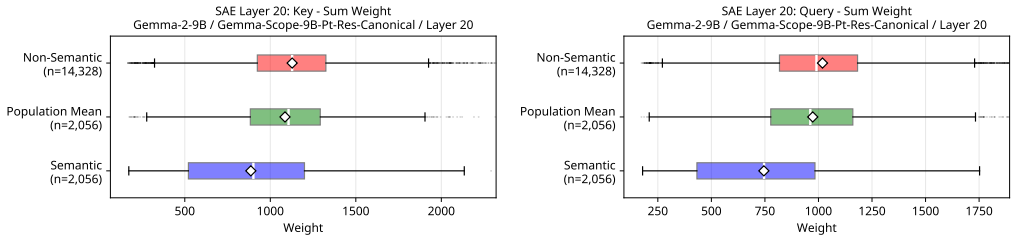


Figure 16: Feature count trends across Llama-3.1-8B layers showing features with top-10 query-based post-softmax attention weight sum above 0.95 (left) and key-based attention weight mean above 0.25 (right). Note that there are two y-axes of different magnitudes: the left y-axis is for the mean and median counts across heads, while the right y-axis is for the union across all heads ("any head"). Solid lines include weights of self-attention; dashed lines exclude self-attention.



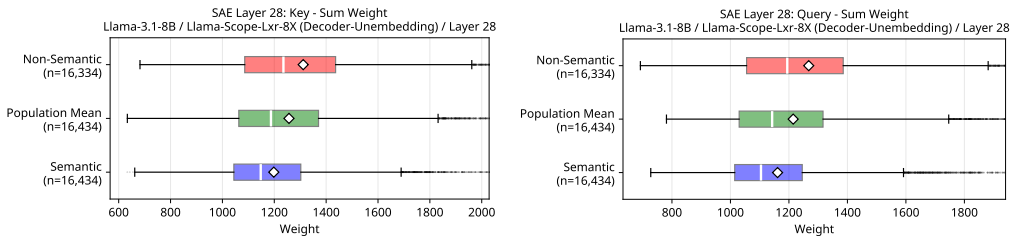
(a) Key-based. (b) Query-based.

Figure 17: Attention weight distributions in Gemma-2-2B layer 12. White diamonds mark means and white lines mark medians.



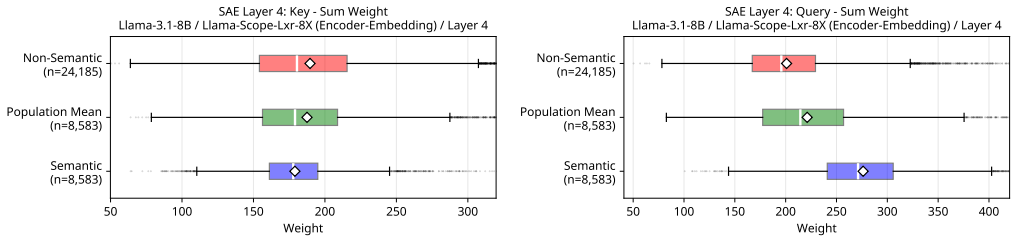
(a) Key-based. (b) Query-based.

Figure 18: Attention weight distributions in Gemma-2-9B layer 20. White diamonds mark means and white lines mark medians.



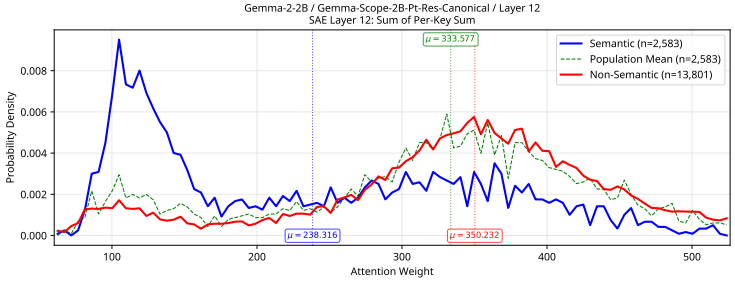
(a) Key-based. (b) Query-based.

Figure 19: Attention weight distributions in Llama-3.1-8B layer 28. White diamonds mark means and white lines mark medians.

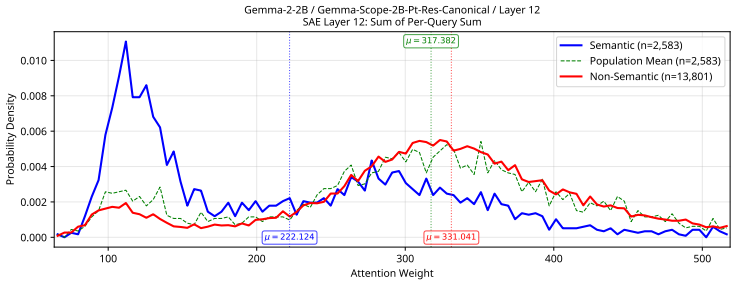


(a) Key-based. (b) Query-based.

Figure 20: Attention weight distributions in Llama-3.1-8B layer 4. White diamonds mark means and white lines mark medians.

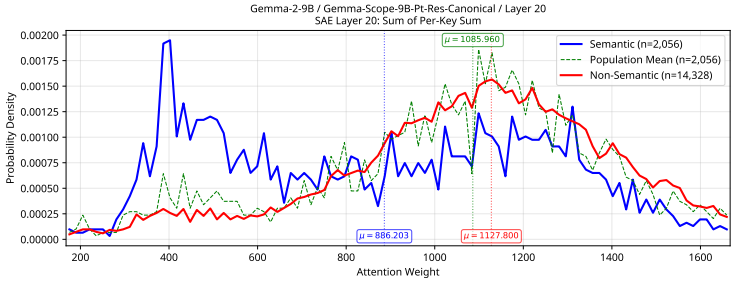


(a) Key-based.

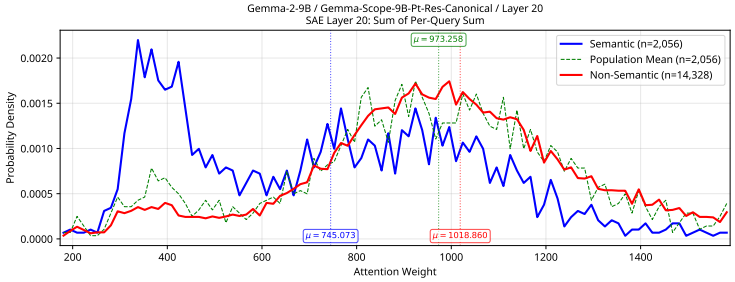


(b) Query-based.

Figure 21: Probability densities of attention weights in Gemma-2-2B layer 12. Dotted lines mark means.

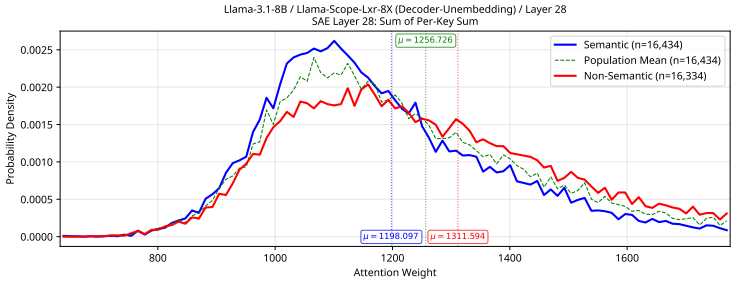


(a) Key-based.

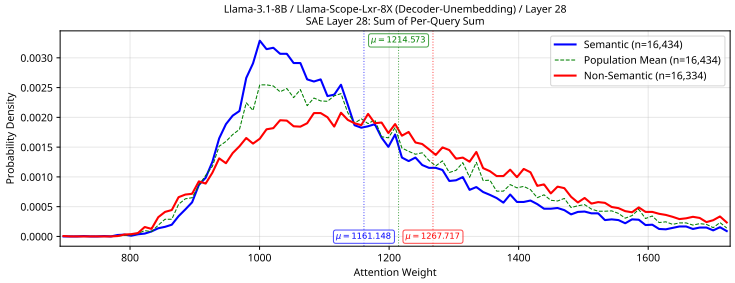


(b) Query-based.

Figure 22: Probability densities of attention weights in Gemma-2-9B layer 20. Dotted lines mark means.

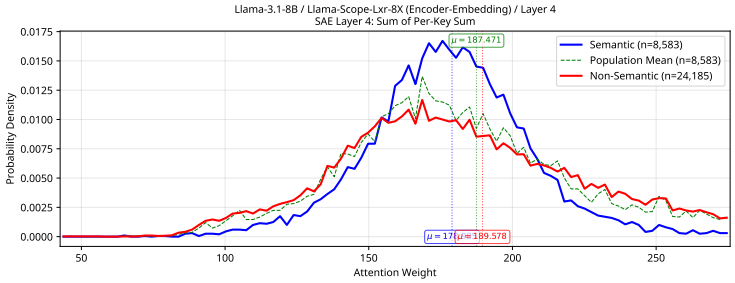


(a) Key-based.

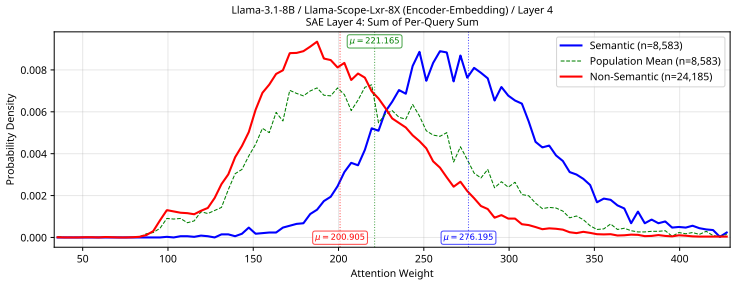


(b) Query-based.

Figure 23: Probability densities of attention weights in Llama-3.1-8B layer 28. Dotted lines mark means.

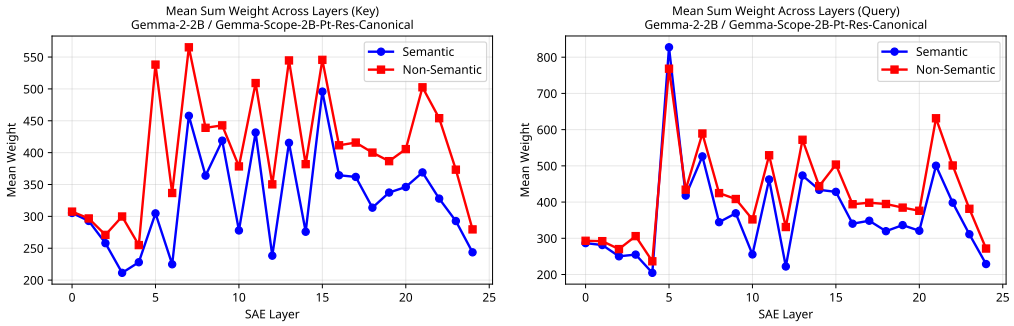


(a) Key-based.



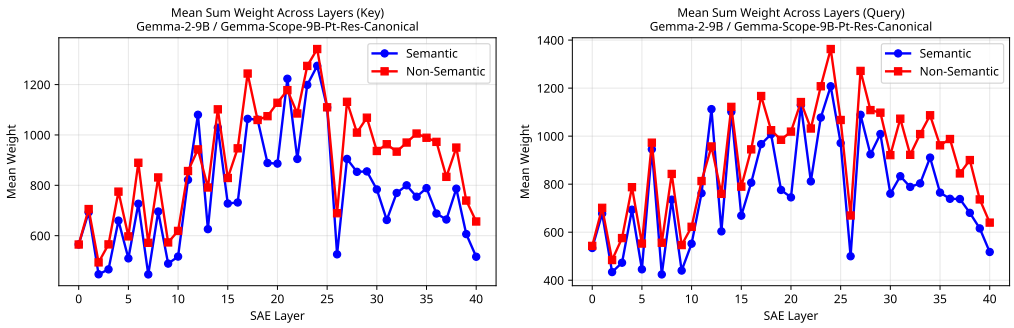
(b) Query-based.

Figure 24: Probability densities of attention weights in Llama-3.1-8B layer 4. Dotted lines mark means.



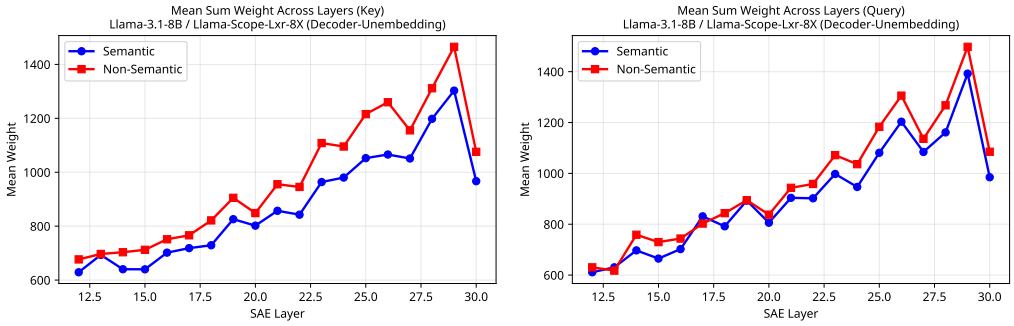
(a) Key-based. (b) Query-based.

Figure 25: Mean attention weights across layers in Gemma-2-2B.



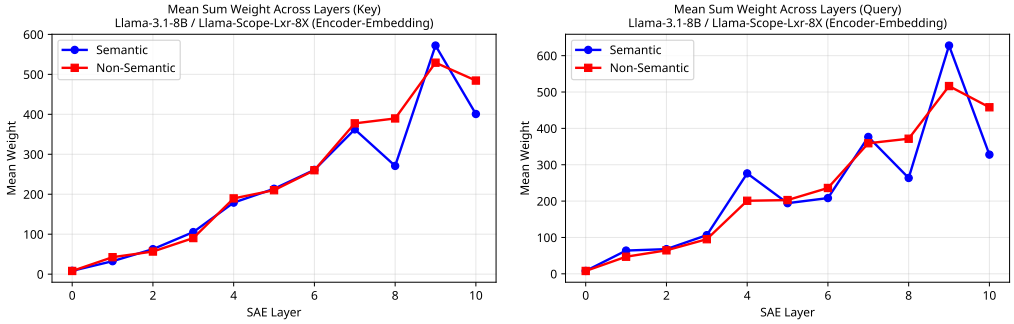
(a) Key-based. (b) Query-based.

Figure 26: Mean attention weights across layers in Gemma-2-9B.



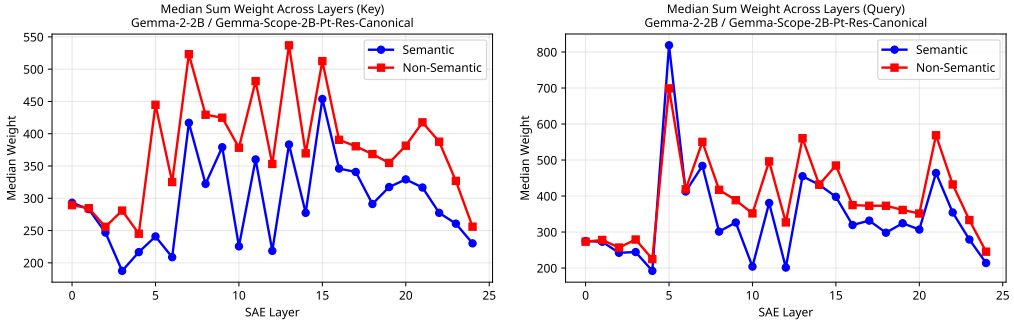
(a) Key-based. (b) Query-based.

Figure 27: Mean attention weights across layers in Llama-3.1-8B, decoder–unembedding phase (layers 12–31).



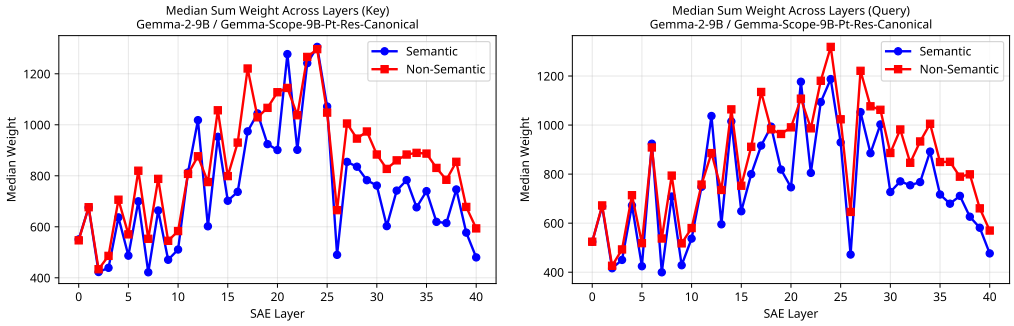
(a) Key-based. (b) Query-based.

Figure 28: Mean attention weights across layers in Llama-3.1-8B, encoder–embedding phase (layers 0–10).



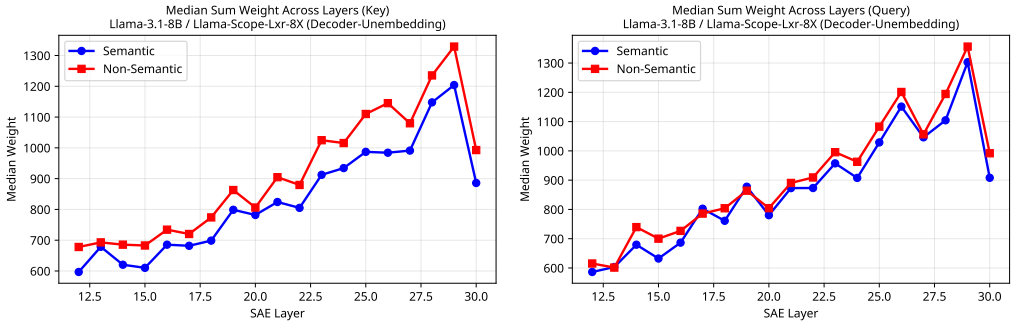
(a) Key-based. (b) Query-based.

Figure 29: Median attention weights across layers in Gemma-2-2B.



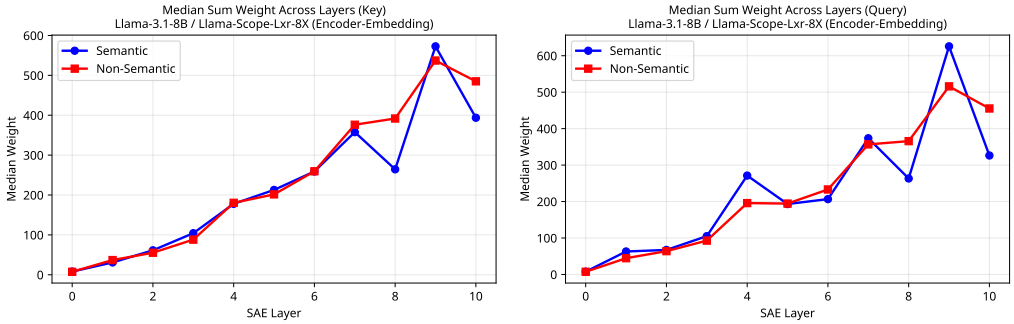
(a) Key-based. (b) Query-based.

Figure 30: Median attention weights across layers in Gemma-2-9B.



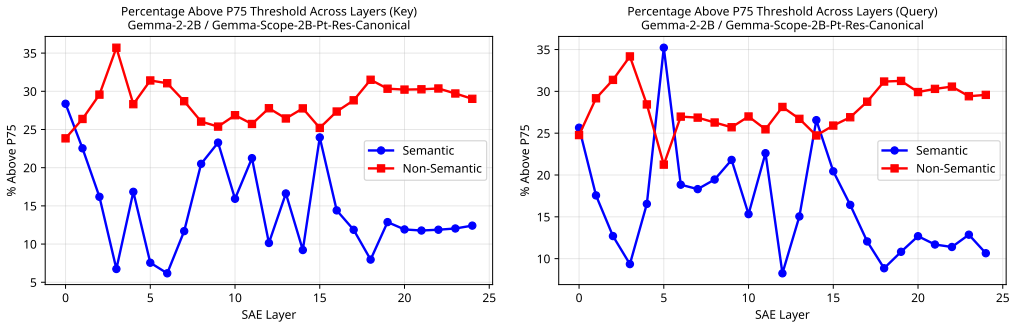
(a) Key-based. (b) Query-based.

Figure 31: Median attention weights across layers in Llama-3.1-8B, decoder–unembedding phase (layers 12–31).



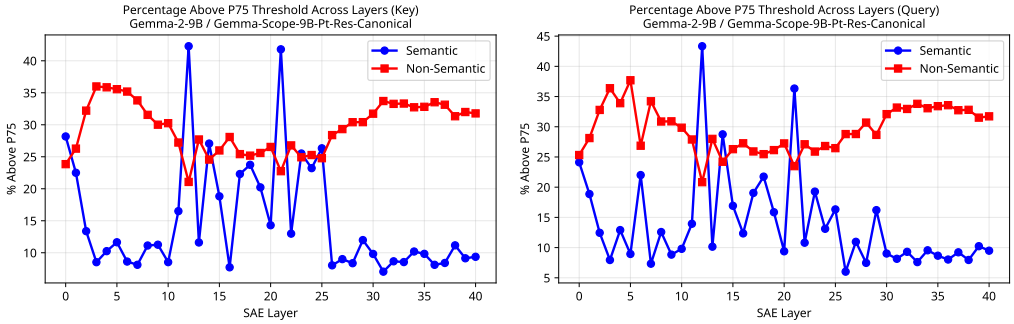
(a) Key-based. (b) Query-based.

Figure 32: Median attention weights across layers in Llama-3.1-8B, encoder–embedding phase (layers 0–10).



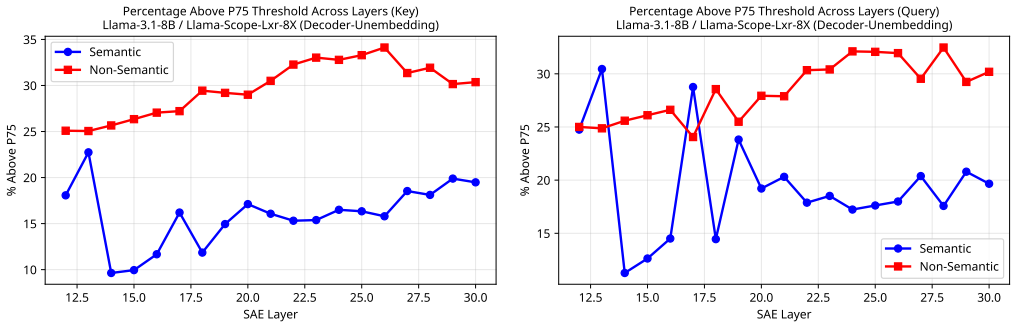
(a) Key-based. (b) Query-based.

Figure 33: Percentage of features exceeding global P_{75} threshold across layers in Gemma-2-2B.



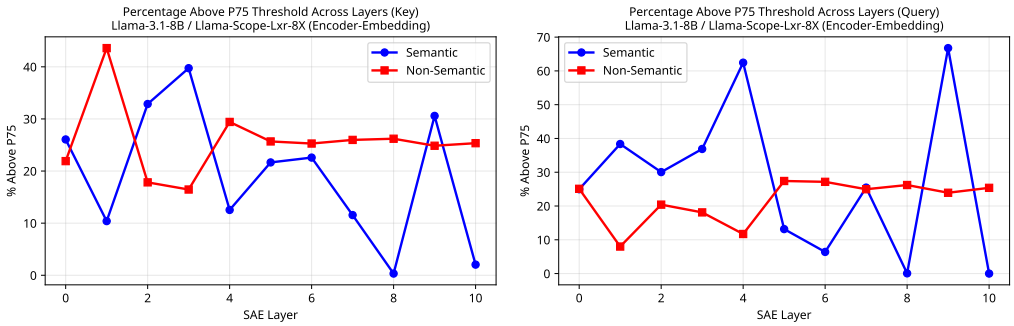
(a) Key-based. (b) Query-based.

Figure 34: Percentage of features exceeding global P_{75} threshold across layers in Gemma-2-9B.



(a) Key-based. (b) Query-based.

Figure 35: Percentage of features exceeding global P_{75} threshold across layers in Llama-3.1-8B, decoder–unembedding phase (layers 12–31).



(a) Key-based.

(b) Query-based.

Figure 36: Percentage of features exceeding global P_{75} threshold across layers in Llama-3.1-8B, encoder–embedding phase (layers 0–10).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state three contributions (a weight-based interpretation framework, an investigation of semantic and computational feature roles, and three depth-dependent architectural signatures) and these are exactly what Sections 3 to 6 establish. Scope is limited to two model families (Gemma-2 and Llama-3.1) and to attention layers, both stated in the text.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 3 flags three modeling choices (per-feature normalization, encoder–decoder asymmetry, and the decoder-side scope of the do-intervention) and Section D expands each with empirical evidence on its impact. The framework is also restricted to attention participation in the next layer; MLP and longer-range routing are explicitly out of scope.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not state formal theorems. The only derivation, which expresses the forward-pass attention score as a bilinear form $z_i z_j s_{i,j}^{(h)}$ and motivates the weight-based score matrix, is given step by step in Section C alongside its assumptions.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All weight-based projections, thresholds (τ_Q, τ_K, k, calibration layers in Table 9), and the do-intervention recipe are stated in the main text. The base models (Gemma-2-2B/9B, Llama-3.1-8B), the SAE suites (Gemma Scope, Llama Scope), and the LLM judge (Llama-3-70B-Instruct) are publicly available, and the validation appendices (Sections B and F) describe the prompts, sample sizes, and matched-baseline construction used.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper commits to publicly releasing the code required to reproduce all experiments. All data sources (base models, SAE suites, judge model) are already openly available; no new dataset is introduced.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We train no new models; all settings are evaluation-time. The three semantic thresholds and their calibration layers are reported in Section 4 and Table 9; the attention thresholds $\tau_Q = 0.95$, $\tau_K = 0.25$, and $k = 10$ are specified in Section 5 with sensitivity analysis in Section E; the intervention sample sizes, baseline-matching procedure, and judge-prompt setup are in Sections B and F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Validation experiments report Cohen's κ with per-cell ranges across the LLM-judge cells (Section 4), per-pair win-rates and median $|\Delta\text{logit}|$ ratios across many independent feature pairs (Sections 4 and 5), and Spearman rank correlations across heads and layers for the normalization sensitivity check (Section D). These quantify variability across the relevant population of features rather than across reruns, since the projections are deterministic given fixed weights.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The weight-based pipeline performs only matrix products against pre-trained model and SAE weights, so the dominant cost is loading the largest checkpoint into memory; a single GPU with at least 48 GB VRAM suffices for Gemma-2-9B and Llama-3.1-8B, and the full set of projections runs in hours rather than days. The intervention and LLM-judge experiments require additional inference passes through the host model and Llama-3-70B-Instruct, which we ran on a single multi-GPU node; sample sizes are reported in Sections B and F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work analyzes already-released open-weight models and SAEs, introduces no new datasets, involves no human subjects, and contributes interpretability tooling for understanding existing language models.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper develops foundational interpretability methodology for existing language models. The positive impact (better mechanistic understanding supports alignment and oversight work) is indirect, and we are not aware of a direct path from a weight-based feature description to negative downstream applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper releases analysis code only; it does not release new models, data, or scraped content with a high risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets (Gemma-2 and Llama-3.1 base models, the Gemma Scope and Llama Scope SAE suites, and Llama-3-70B-Instruct as the judge) are cited are used within their respective public licenses. No proprietary or scraped data are used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The only new asset is the analysis code, which will be released publicly with documentation covering the four projections, the do-intervention primitive, and scripts that reproduce every figure and table.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper involves no crowdsourcing or human-subject research; threshold calibration uses author inspection of weight-derived statistics, and the only judge is a language model.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper involves no human-subject research, so IRB review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of our core methods. The framework is weight-based and uses no LLM in its inner loop. The Llama-3-70B-Instruct judge in Section B.2 serves only as an external sanity check on our inference-free classifier and is not part of the proposed methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.