DO FEATURES IN A SPARSE AUTOENCODER REALLY REPRESENT CLEAR SEMANTICS?

Anonymous authors
Paper under double-blind review

ABSTRACT

The sparse autoencoder (SAE) is a widely used mechanistic interpretability method that transforms dense LLM activations into sparse features. In this paper, we propose using interactions as a numerically verifiable metric to evaluate the semantic clarity of SAE features from four perspectives. (1) We observe that 19.2%–35.6% of the sampled SAE feature activations encode non-sparse interaction patterns, suggesting more diverse semantics. The average number of interactions per feature varies dramatically, and we find no clear correlation between the interaction number and the feature's activation frequency. (2) We find that SAE feature activations that encode fewer interactions tend to model simpler and more human-interpretable interactions. (3) 25.2%–41.9% of the feature activations encode interactions with severe positive-negative cancelling effects, indicating potentially less coherent semantics. (4) We find that 6.2%–19.5% of the SAE feature activations capture similar semantics across different contexts, and these activations also encode interactions with less positive-negative cancellation.

1 INTRODUCTION

Mechanistic interpretability is an established and rapidly expanding research area that aims to elucidate the internal mechanisms of neural networks through techniques such as logit lens and probes (Geva et al., 2022; Hewitt & Liang, 2019), sparse autoencoders (SAEs) (Huben et al., 2024; Templeton et al., 2024; Gao et al., 2025), and circuit analysis (Lindsey et al., 2025; Ameisen et al., 2025). Among these, the SAE is one of the most widely used methods and transforms dense LLM activations into sparse features. Each SAE feature is intended to represent clear semantics.

In SAE research, evaluating (Bricken et al., 2023; Gulko et al., 2025), questioning (Lu et al., 2025; Song et al., 2026), and enhancing (Gao et al., 2025; Rajamanoharan et al., 2024a;b) the semantic clarity of sparse features are persistent core tasks. Although the sparsity of SAE features is mathematically guaranteed, how to ensure their semantic clarity remains a non-trivial challenge. Most existing semantic analysis methods are largely empirical and cannot rigorously quantify all subtle input signals that affect an SAE feature. For instance, Bills et al. (2023) empirically determined the meaning of a feature by summarizing the meaning of input tokens that strongly activate it, while Bricken et al. (2023) interpreted an SAE feature by inspecting the feature's effects on the LLM's outputs. Quantitative tools for examining the semantic clarity of SAE features remain limited.

Interaction-based explanation. As a complement to previous empirical explorations, we propose using interactions as a numerically verifiable metric to analyze the semantic clarity of SAE features. This is because Ren et al. (2024a) have found and proven that a seemingly complex network function can generally be well approximated by a logical function consisting of AND-OR interactions between input tokens. The high morphological similarity between these two functions has led researchers (Ren et al., 2023a; Li & Zhang, 2023; Ren et al., 2024b) to regard the AND-OR interactions as faithful inference patterns of the network function.

Using interactions to evaluate the semantic clarity of SAE features. In this paper, we decompose the activation value of each SAE feature at a specific token position into a set of AND-OR interactions. As shown in Figure 1, each interaction corresponds to a phrase pattern composed of several tokens and contributes a numerical effect to the feature's activation value. We stress that, much like SAEs, interactions represent a method for explaining the internal logic in a DNN, rather than a perfect interpretable ground truth. Nevertheless, the advantage of interactions is that they are

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

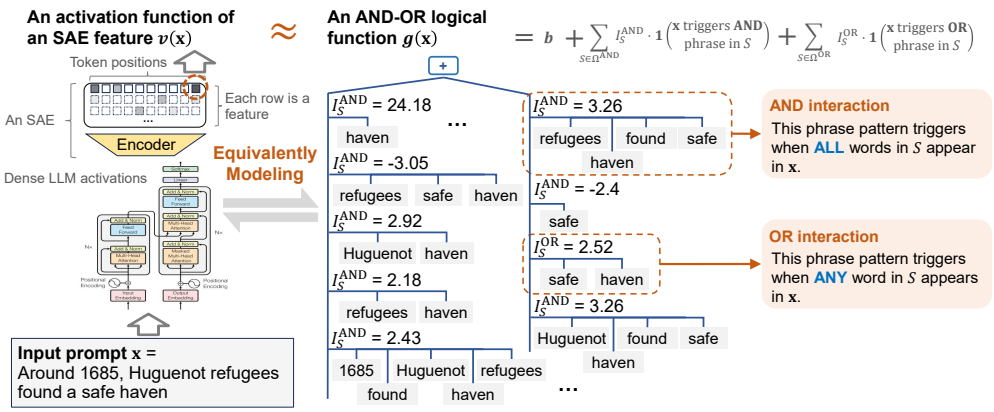


Figure 1: The activation value v of an SAE feature on a given input $\mathbf{x}$ can usually be well approximated by a logical function g composed of AND-OR interactions. Each interaction represents a non-linear phrase pattern, which is equivalently modeled by the SAE feature and contributes an effect to the feature's activation value. In this paper, we use interactions to explain the semantic clarity of SAE features. Appendix M provides more explanations for other SAE features.

inherently interpretable, and precisely decomposing a feature's function into interactions provides a verifiable metric to evaluate all subtle signals within the feature. Consequently, interactions offer four new diagnostic perspectives on SAE feature semantics.

• **Interaction number** measures the count of interactions used by an SAE feature activation during inference. Experiments show that 64.4%–80.8% of SAE feature activations encode ≤ 10 interactions, indicating concentrated semantics. In contrast, 5.9%–22.2% of feature activations encode $>$ 20 interactions, suggesting that these SAE feature activations are simultaneously affected by more diverse phrase patterns. Furthermore, the interaction number encoded by an SAE feature can vary dramatically from 1 to 156 when it activates at different token positions. We find no clear relationship between an SAE feature's activation frequency[1] and its average interaction number.

• **Interaction complexity** measures the complexity of the interactions encoded by an SAE feature activation. SAE feature activations encoding sparse interactions ($\leq$10) primarily rely on low-order interactions (orders 1–3). Conversely, feature activations encoding high-density interactions ($>$20) primarily rely on mid-order interactions (orders 4–6). The distributions show a positive trend between the number of interactions encoded by an SAE feature activation and the complexity of those interactions.

• **Positive-negative cancellation of interactions.** We introduce the Positive-Negative Cancellation Index (PNCI) to quantify how much positive and negative interaction effects cancel each other out. A higher PNCI indicates stronger mutual cancellation among interaction effects and may reflect potentially conflicting or less coherent semantics. Experiments show that 58.1%–74.8% of the sampled SAE feature activations achieve PNCI ≤ 0.5, suggesting that their interaction effects are less mutually cancelling. We also find that activation strength does not clearly correlate with the PNCI.

• **Semantic consistency of interactions.** We evaluate whether the same SAE feature consistently represents similar semantics across different positions. For each SAE feature, we identify its primary semantics as its most frequent (modal) semantic concept. We find that 6.2%–19.5% of the sampled activations successfully express the primary semantics of their corresponding SAE features. These activations also tend to exhibit less positive-negative cancellation (i.e., lower PNCI).

2 Related Work

Mechanistic interpretability studies the internal representations of neural networks to better understand how the networks produce their outputs. Within this area, probing methods (Hewitt & Liang, 2019; Belinkov, 2022; Ghandeharioun et al., 2024) typically define a semantic concept in

[1] An SAE feature's activation frequency refers to the fraction of tokens in the dataset that activate this feature.

advance and test whether this concept can be detected from LLM activations. Logit lens and its variants (Geva et al., 2022; Dar et al., 2023) unembed the intermediate representations through a vocabulary matrix, providing a way to inspect how LLMs refine their predictions across layers. Circuit analysis (Conmy et al., 2023; Lindsey et al., 2025; Ameisen et al., 2025) aims to identify subnetworks or computational pathways that are involved in specific model behaviors. However, these methods often suffer from the polysemanticity problem (Elhage et al., 2022), i.e., a single neuron in LLMs may simultaneously represent multiple unrelated semantic concepts, making it difficult to interpret and steer model behavior.

To this end, **sparse autoencoders (SAEs)** are proposed to transform dense LLM activations into a higher-dimensional, sparse feature space (Bricken et al., 2023; Templeton et al., 2024). Each sparse feature ***is expected to*** represent a clear semantic concept. Various SAE architectures have been proposed, which are detailed in the *Preliminaries* section. However, the sparse activation of an SAE feature does not inherently guarantee its semantic clarity. Previous studies primarily summarize the semantics of SAE features through inductive observations (Bills et al., 2023; Bricken et al., 2023). To achieve a quantifiable and verifiable evaluation, in this paper, we investigate the semantic clarity of SAE features by analyzing the interactions they encode.

Interaction-based explanation. Interaction-based explanation has emerged as a new method for disentangling primitive inference patterns from deep neural networks (DNNs). Numerous empirical works (Ren et al., 2023a; Li & Zhang, 2023) have found, and Ren et al. (2024a) have mathematically proved, that interactions possess the universal matching property and, under suitable conditions, a sparsity property. Together, these properties support faithful interaction-based explanations when their assumptions or empirical conditions are satisfied. Furthermore, Chen et al. (2024); Lou et al. (2026) have shown that the overall generalization power of DNNs can be explained as the ensemble effect of the generalization power of individual interactions. Regarding robustness, interactions have been employed to analyze adversarial robustness (Deng et al., 2024) and the robustness of Bayesian neural networks (Ren et al., 2023c). Moreover, interactions have been leveraged to explain representation bottlenecks (Deng et al., 2022) and to discover and explain the two-phase training dynamics of DNNs (Ren et al., 2024b).

3 Using Interactions to Explain Features

3.1 Preliminaries

Sparse autoencoder. Given an LLM, let $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]^\top$ denote the activation matrix of the l-th layer, where $\mathbf{a}_i \in \mathbf{R}^d$ is the activation vector at the i-th token position. An SAE is trained to map the dense activation vector $\mathbf{a}_i$ into higher-dimensional yet sparse features $\mathbf{h}_i \in \mathbf{R}^s$ ($s \gg d$), which are then used to reconstruct the original activation vector $\mathbf{a}_i$ as follows:

$$\begin{aligned} &\min_{\mathbf{W}_{\text{enc}}, \mathbf{b}_{\text{enc}}, \mathbf{W}_{\text{dec}}, \mathbf{b}_{\text{dec}}} \mathcal{L} = \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_2^2 + \lambda \cdot \text{sparsity}(\mathbf{h}_i), \\ &\text{with } \begin{cases} \mathbf{h}_i = \sigma(\mathbf{W}_{\text{enc}}\mathbf{a}_i + \mathbf{b}_{\text{enc}}) & \text{(Encoder)} \\ \hat{\mathbf{a}}_i = \mathbf{W}_{\text{dec}}\mathbf{h}_i + \mathbf{b}_{\text{dec}} & \text{(Decoder)} \end{cases}, \end{aligned} \tag{1}$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm and $\text{sparsity}(\cdot)$ denotes the sparsity penalty applied to the SAE features. $\lambda \geq 0$ is a scalar coefficient, and $\sigma(\cdot)$ is a non-linear activation function.

The above paradigm has different implementations in different studies. For example, Bricken et al. (2023) employed a ReLU activation function with an ℓ_1-norm sparsity penalty. TopK SAEs (Gao et al., 2025) and JumpReLU SAEs (Rajamanoharan et al., 2024b) adopted TopK and JumpReLU activation functions, respectively, both utilizing an ℓ_0 sparsity penalty (see Appendix C for details).

SAE features. Each dimension of the latent representation $\mathbf{h}_i$ is an SAE feature, which is supposed to capture a distinct semantic concept. An activated feature (i.e., $h_{ij} \neq 0$) signals that the corresponding semantic concept is detected by the LLM in the input prompt.

Using interactions to represent primitive inference patterns in a black-box model. Given an input prompt $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$ consisting of n input variables (tokens/words) indexed by $N = \{1, 2, \ldots, n\}$, let $v(\mathbf{x}) = \log \frac{p(x_{n+1}|\mathbf{x})}{1-p(x_{n+1}|\mathbf{x})}$ denote the log-odds (prediction score) of an LLM v generating the target token x_{n+1}. Ren et al. (2024a) prove that the target function $v(\mathbf{x})$ can be

effectively interpreted by a logical function g composed of AND-OR interactions. Specifically, the output values of $v(\mathbf{x})$ across all 2^n masked states[2] of the input $\mathbf{x}$ can be accurately matched by the logical function g:

$$\forall \mathbf{x}' \in \Psi,\ |g(\mathbf{x}') - v(\mathbf{x}')| < \epsilon,$$
$$g(\mathbf{x}') \overset{\text{def}}{=} \sum_{S\in\Omega^{\text{AND}}} \underbrace{I_S^{\text{AND}} \cdot \mathbf{1}(^{\mathbf{x}' \text{ triggers AND}}_{\text{phrase in } S})}_{\text{an AND interaction}} + \sum_{S\in\Omega^{\text{OR}}} \underbrace{I_S^{\text{OR}} \cdot \mathbf{1}(^{\mathbf{x}' \text{ triggers OR}}_{\text{phrase in } S})}_{\text{an OR interaction}} + b. \tag{2}$$

The above equation is also referred to as **the universal matching property** of interactions, where the small matching-error bound ϵ controls the numerical fidelity of the explanation. Here, $\Psi = \{\mathbf{x}_S \mid S \subseteq N\}$ denotes the set of all 2^n masked states[2] of $\mathbf{x}$, and $\mathbf{x}_S$ represents a specific state in which only the input variables (tokens/words) in subset S are retained (i.e., unmasked). Furthermore, Ω^{AND} and Ω^{OR} denote the sets of non-empty AND and OR interactions, respectively; the empty-set state contributes only through the scalar bias $b = v(\mathbf{x}_\emptyset)$. The parameters $I_S^{\text{AND}}, I_S^{\text{OR}}, b \in \mathbb{R}$ are learned following the method in (Li & Zhang, 2024)[3].

• The AND interaction $I_S^{\text{AND}} \cdot \mathbf{1}(^{\mathbf{x}' \text{ triggers AND}}_{\text{phrase in } S})$ represents an AND phrase pattern among the input variables in S. Its trigger function $\mathbf{1}(^{\mathbf{x}' \text{ triggers AND}}_{\text{phrase in } S}) \in \{0, 1\}$ outputs 1 if and only if all variables in S are present in the masked sample $\mathbf{x}'$. The numerical effect I_S^{AND} quantifies the contribution of this interaction to $v(\cdot)$: a positive I_S^{AND} increases $v(\mathbf{x}')$ by $|I_S^{\text{AND}}|$, whereas a negative one decreases $v(\mathbf{x}')$.

• Similarly, $I_S^{\text{OR}} \cdot \mathbf{1}(^{\mathbf{x}' \text{ triggers OR}}_{\text{phrase in } S})$ represents an OR phrase pattern among the input variables in S. Its trigger function outputs 1 if at least one variable in S appears in $\mathbf{x}'$, and outputs 0 otherwise.

The conciseness of interaction-based explanations is further motivated by **the sparsity property**. Prior work has repeatedly observed and theoretically characterized sparse salient interactions for suitable target functions (Ren et al., 2024a)[4]. Empirically, an LLM employs 50–150 interactions to compute the prediction score for a target token. Because the conditions underlying the formal sparsity result need not hold exactly for intermediate SAE activations, we treat sparsity as an empirical hypothesis in this setting and test it directly. Our experiments show that SAE feature activations with sparse interaction sets often use fewer than 10 interactions at a specific token position.

Together, the universal matching property and the sparsity property provides a numerically verifiable and concise interaction-based explanationion. A detailed discussion of the above two properties is provided in Appendix D, and a video demo is included in the supplementary material to help readers intuitively understand the interactions. Additionally, given the high computational complexity of interactions, we discuss their efficient computation in Appendix H.

3.2 Sparse Activations vs. Sparse Semantics

In the community of mechanistic interpretability, SAE methods enforce activation sparsity, and sparsely activated features are often assumed to represent clear semantics (Bricken et al., 2023; Huben et al., 2024). However, activation sparsity does not, in principle, strictly guarantee monosemanticity. How to identify the vague, entangled semantics encoded by SAE features therefore remains a major open challenge.

Therefore, we take a complementary perspective: in addition to activation-based analyses, we examine SAE features through the lens of interactions (i.e., phrase patterns) they encode. This is motivated by three facts: (1) each interaction corresponds to a distinct phrase pattern equivalently modeled by the SAE feature (see Figure 1). (2) Because of the proven universal matching property and sparsity property, interactions provide a precise and verifiable way to identify the primitive phrase patterns modeled by each SAE feature (see the Preliminaries section). (3) Interactions are

[2]Following Ren et al. (2023b), a masked state of an input sample $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$ is obtained by replacing the original values of a subset of input variables x_i with baseline values.

[3]The pseudocode for the interaction extraction algorithm is provided in Appendix G.

[4]An interaction is salient if its absolute effect exceeds a threshold τ. Typically, $\tau \in [0.02, 0.10] \times \mathbb{E}_{\mathbf{x}}|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$.

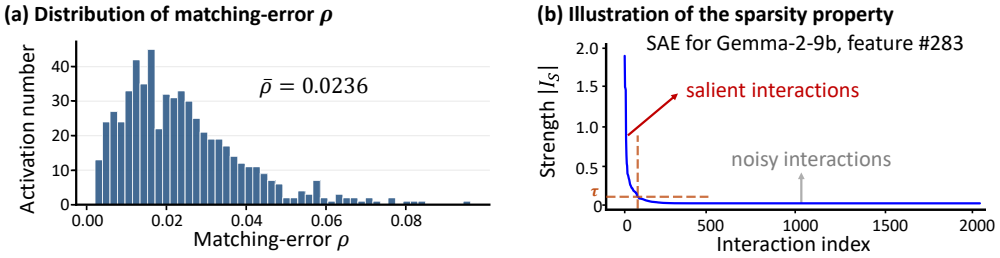


Figure 2: (a) The matching error ρ between the SAE feature activation v and the logical function g across different samples. (b) Interaction strengths for SAE feature #283 (Gemma-2-9B) on token position #30912, sorted in descending order.

transferable across similar samples (Li & Zhang, 2023) and across different LLMs: as found by Lou et al. (2026), about $2/3$ to $4/5$ of the interactions used to predict the next token are shared by many open-source LLMs[5]. The existence of many shared interactions suggests that interactions are not merely a trivial mathematical transformation of the input, but capture meaningful inference patterns shared across LLMs (Zhou et al., 2024).

Interpretation objective. In this paper, we propose to interpret each SAE feature using interactions. Formally, we define the activation value of the j-th SAE feature at the i-th token position of an input $\mathbf{x}$ as an **SAE feature activation** $h_{ij} \in \mathbb{R}$. We then define the function that computes this SAE feature activation as a target function, denoted by $v(\mathbf{x}) = h_{ij}$. For the threshold-based SAEs evaluated in our experiments, interaction extraction uses a continuous ReLU surrogate for the native activation operator; this substitution is used only for interaction computation and is described in Appendix I.4.

Visualizing interactions encoded by SAE features. We first visualize the interactions that contribute to the SAE feature activation. As shown in Figure 1, let us consider feature $j = 4699$ from an SAE trained on the residual stream at layer 20 of Gemma-2-9B. This feature activates at the last token of the input prompt. Its activation value, $v(\mathbf{x}) = 26.0$, can be decomposed into a small set of interactions. Each interaction corresponds to a distinct phrase pattern.

> The three most salient interactions are as follows: an AND interaction over $S_1 = \{\text{heaven}\}$, contributing $I_{S_1}^{\text{AND}} = +24.18$; an AND interaction over $S_2 = \{\text{refugee, found, safe, heaven}\}$, contributing $I_{S_2}^{\text{AND}} = +3.26$; and an OR interaction over $S_3 = \{\text{safe, heaven}\}$, contributing $I_{S_3}^{\text{OR}} = +2.52$. Specifically, the AND interaction over S_1 suggests that the word "heaven" alone dominates the activation value. The AND interaction over S_2 further captures a specific narrative of seeking refuge. The OR interaction over S_3 captures the shared meaning between "safe" and "heaven", indicating that the feature can also respond to general safety-related semantics. As shown above, the largest interactions encoded by a feature activation mainly focus on safety-related semantics. More examples are provided in the Appendix M.

Verifying the numerical fidelity of interaction-based explanations. The numerical fidelity of interaction-based explanations is theoretically guaranteed by the universal matching property (Ren et al., 2024a). Given an input $\mathbf{x}$, the logical function g composed of AND-OR interactions should accurately match the SAE feature activation v across all masked states[2] of $\mathbf{x}$, i.e., $\forall \mathbf{x}' \in \Psi$, $|v(\mathbf{x}') - g(\mathbf{x}')| < \epsilon$. To evaluate this matching fidelity, for each SAE feature activated at a specific token position, we measure the normalized matching error between the SAE feature activation v and the logical function g. The error is defined as:

$$\rho = \mathbb{E}_{\mathbf{x}' \in \Psi}\left[\frac{|g(\mathbf{x}') - v(\mathbf{x}')|}{|v(\mathbf{x})|}\right]. \tag{3}$$

We conduct experiments on an SAE trained on the residual stream at layer 20 of Gemma-2-9B (Lieberum et al., 2024); detailed experimental settings are provided in Appendix I. Figure 2(a) shows that the matching error ρ does not exceed 0.05 for most feature activations.

[5]These LLMs include Qwen, LLaMA, GLM, DeepSeek, Gemma, and others.

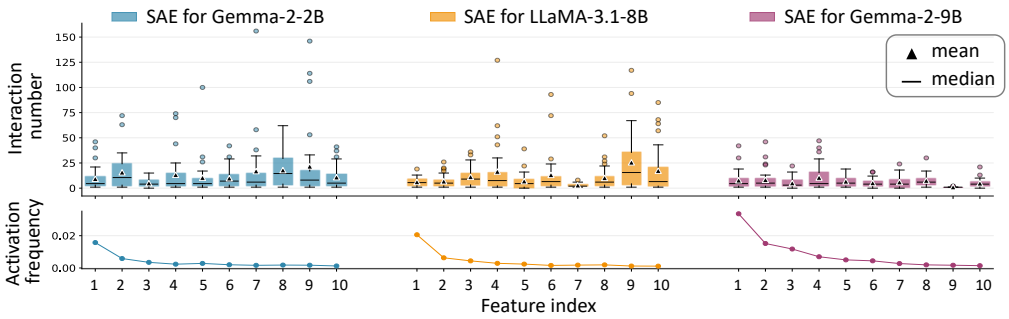


Figure 3: Distribution of interaction counts encoded by individual SAE features across different token positions. The x-axis represents different SAE features j (only 10 randomly selected features per SAE model are shown), sorted by their activation frequency[1]. The y-axis of the boxplot represents the number of interactions encoded by SAE feature j at token position i.

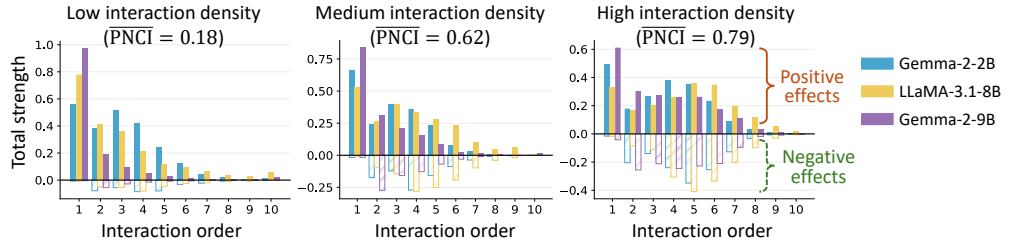


Figure 4: Histograms of the total interaction strength across different orders for each SAE feature activation group. Positive and negative strength are computed separately. Due to the substantial differences in interaction strength across SAE models, we normalize the ℓ_2-norm of the mean complexity vector $\bar{\mathbf{d}}$ within each group. $\overline{\text{PNCI}}$ denotes the mean PNCI of this group.

Excluding noisy patterns from evaluation. Although we do not rigorously prove that interaction sparsity still holds in the SAEs, we indeed empirically observe that the activation of an SAE feature is typically dominated by a few salient interactions[4]. Therefore, we evaluate the semantics of an SAE feature using only these salient interactions. In contrast, many other non-salient interactions are much weaker and mainly reflect minor noise inevitably captured by current SAE models. As shown in Figure 2(b), there is a substantial strength gap between salient and non-salient interactions.

3.3 Evaluating The Semantic Clarity of SAE Features

In this subsection, we analyze the semantic clarity of SAE features through the lens of interactions. We select the following three classic open-source SAEs: (i) a JumpReLU SAE (Lieberum et al., 2024) built upon the MLP output at layer 12 of Gemma-2-2B; (ii) a JumpReLU SAE (Lieberum et al., 2024) built upon the residual stream at layer 20 of Gemma-2-9B; (iii) an improved TopK SAE (He et al., 2024) built upon the residual stream at layer 15 of LLaMA-3.1-8B. Given each input prompt, we compute interactions among content words using a ReLU surrogate for the SAE activation function. Because of computational constraints, we analyze a sampled subset of SAE features and retained activations rather than the full dictionaries and activation streams. Please see Appendix I for detailed settings.

Experiment 1: analysis of the interaction number encoded by SAE feature activations. We take the number of interactions encoded by an SAE feature activation as the first metric for semantic clarity. A semantically clear feature is expected to encode only a small number of interactions with concentrated semantics. Conversely, encoding a large number of interactions indicates that the feature is likely activated by various phrase patterns representing diverse semantics.

SAE Model	Low interaction density group (encoding ≤10 interactions)	Medium interaction density group (encoding 11–20 interactions)	High interaction density group (encoding >20 interactions)
SAE for Gemma-2-2B	64.4%	17.9%	17.7%
SAE for LLaMA-3.1-8B	65.9%	11.9%	22.2%
SAE for Gemma-2-9B	80.8%	13.3%	5.9%

Table 1: Distribution of SAE feature activations based on interaction counts across different SAE models.

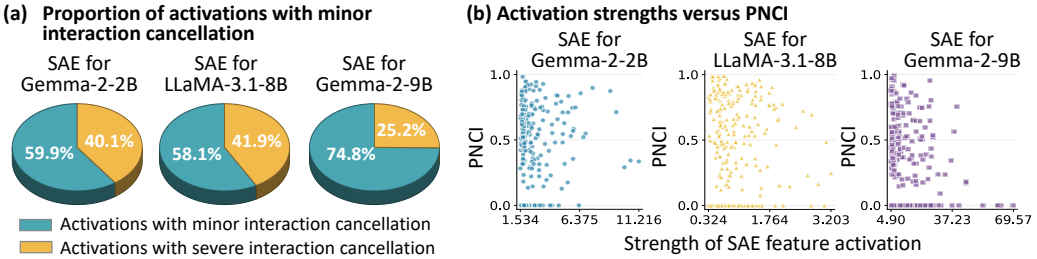


Figure 5: (a) Proportion of SAE feature activations that encode interactions with minor mutually cancelling effects (PNCI ≤ 0.5) versus those with severe mutually cancelling effects (PNCI > 0.5). (b) Scatter plot of SAE feature activation strength against the corresponding PNCI. Each point is a single SAE feature activation.

Specifically, according to Eq. equation 2, we enumerate all activated SAE features across different token positions, and extract a specific set of interactions by setting each individual SAE feature activation h_{ij} as the target function $v(\mathbf{x})$.

Table 1 reports the number of interactions encoded by different SAE feature activations. For clarity, we categorize these feature activations into three groups based on the number of interactions they encode. (1) **Low interaction density group** comprises feature activations with ≤ 10 interactions, accounting for 64.4%–80.8% of the sampled activations across different SAEs. (2) **Medium interaction density group** comprises feature activations with 11–20 interactions, accounting for 11.9%–17.9%. (3) **High interaction density group** comprises feature activations with > 20 interactions, accounting for 5.9%–22.2%. SAE feature activations with low interaction density rely on highly concise interactions during inference; those with high interaction density depend on a large number of interactions, indicating more diverse semantics.

Finding 1: *a single SAE feature can activate at multiple token positions, but the number of interactions it encodes varies widely.* Figure 3 illustrates the distribution of interaction numbers for the same SAE feature across different token positions i. When the same SAE feature activates at different positions under varying contexts, the number of interactions it encodes spans a wide range (from 1 to 156).

Finding 2: *the average number of interactions encoded by different SAE features varies significantly, with no clear monotonic relationship with their activation frequency*[1]. As shown in Figure 3, the average number of interactions differs substantially across SAE features, ranging from 1.1 to 25.6. Moreover, we find no clear relationship between the average interaction number and a feature's activation frequency.

Experiment 2: analysis of interaction complexity encoded by SAE feature activations. The complexity of an interaction is measured by its order, which is defined as the number of input tokens in the set S, i.e., $\text{order}(I_S^{\text{AND}}) = |S|$. We adopt interaction order as the second metric to evaluate semantic clarity. Low-order interactions involve only 1–3 tokens and are thus more human-interpretable. In contrast, high-order interactions that involve 8–10 tokens are typically opaque and hard to understand (Lou et al., 2026).

In Experiment 1, we have categorized SAE feature activations into three groups based on their interaction number. Here, we further compare the interaction complexity across these groups. The interaction complexity of an SAE feature activation is measured by the distribution of interaction

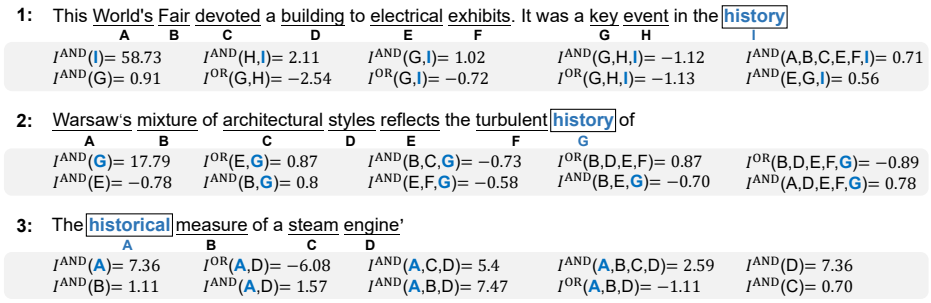


Figure 6: We show the top-ranked interactions extracted from three activations of the feature #4439 (SAE for Gemma-2-9B). The feature activates at the last token position of the prompt. Interactions are computed among *content words* (marked with an underline). These activations all represent the semantic concept of "history". Appendix N provides more examples.

SAE Model	Proportion of activations representing primary semantics	PNCI for activations representing primary semantics	PNCI for activations representing other semantics
SAE for Gemma-2-2B	6.2%	0.316 ± 0.305	0.373 ± 0.334
SAE for LLaMA-3.1-8B	13.0%	0.184 ± 0.228	0.420 ± 0.373
SAE for Gemma-2-9B	19.5%	0.124 ± 0.181	0.282 ± 0.312

Table 2: Comparison of mean PNCI between SAE feature activations that represent primary semantics and those that do not.

strength over different orders, denoted as $\mathbf{d} \in \mathbb{R}^{2n}$.

$$\begin{aligned} \mathbf{d} &= [d_+^{(1)}, \ldots, d_+^{(n)}, d_-^{(1)}, \ldots, d_-^{(n)}]^\top, \\ d_+^{(k)} &= \sum_{S \in \Omega^{\text{AND}}, |S|=k} \max(I_S^{\text{AND}}, 0) + \sum_{S \in \Omega^{\text{OR}}, |S|=k} \max(I_S^{\text{OR}}, 0), \\ d_-^{(k)} &= \sum_{S \in \Omega^{\text{AND}}, |S|=k} \min(I_S^{\text{AND}}, 0) + \sum_{S \in \Omega^{\text{OR}}, |S|=k} \min(I_S^{\text{OR}}, 0), \end{aligned} \tag{4}$$

where $d_+^{(k)}$ and $d_-^{(k)}$ represent the total strength of positive and negative k-th order interactions extracted from the SAE feature activation v, respectively.

Finding 3: *SAE feature activations encoding few interactions primarily model simpler interactions, whereas those encoding many interactions primarily model more complex interactions.* Figure 4 visualizes the mean distribution vector $\bar{\mathbf{d}}$ over all SAE features in each group. Results show that SAE feature activations in low interaction density group (encoding $\leq$ 10 interactions) primarily model low-order interactions (orders 1–3). In contrast, SAE feature activations in high interaction density group (encoding $>$ 20 interactions) not only model many first-order interactions, but also encode many mid-order interactions (orders 4–6). Across all three SAE models, we find that the interaction number is positively correlated with interaction complexity.

Experiment 3: towards the problem of mutually cancelling interactions. We find a potential problem that the positive and negative interaction effects encoded by a single SAE feature activation may cancel each other out. Ren et al. (2024b) previously found in LLMs that mutually cancelling interactions typically correspond to noise patterns. Our subsequent Experiment 4 further demonstrates that SAE features exhibiting severe interaction cancellation tend to represent less coherent semantics. To quantify this, we introduce the ***Positive-Negative Cancellation Index (PNCI)***:

$$\text{PNCI} = 1 - \frac{\left|\sum_{S \in \Omega^{\text{AND}}} I_S^{\text{AND}} + \sum_{S \in \Omega^{\text{OR}}} I_S^{\text{OR}}\right|}{\sum_{S \in \Omega^{\text{AND}}} \left|I_S^{\text{AND}}\right| + \sum_{S \in \Omega^{\text{OR}}} \left|I_S^{\text{OR}}\right|}. \tag{5}$$

A high PNCI means most of the interaction effects are canceled out. As shown in Figure 4, SAE feature activations in the low interaction density group predominantly encode positive interactions with minor cancellation (mean PNCI: 0.16–0.19). In contrast, feature activations in the high interaction density group encode numerous mid-order interactions (orders 4–6) with severe cancellation (mean PNCI: 0.74–0.85), indicating that these activations may capture less coherent semantics.

Furthermore, Figure 5 (a) shows the proportion of SAE feature activations with minor interaction cancellation (PNCI ≤ 0.5) and severe interaction cancellation (PNCI > 0.5). Across the three SAEs, 58.1%–74.8% of feature activations encode interactions with only minor cancellation, suggesting that they capture relatively reliable semantics. See Appendix E for further discussion of the PNCI.

Finding 4: *activation strength does not clearly correlate with the PNCI score.* Although high activation strengths are often considered as strong semantic signals, Figure 5 (b) shows that some strongly activated features still exhibit severe positive-negative cancellation (high PNCI). This is particularly evident in the SAEs for LLaMA-3.1-8B and Gemma-2-2B. This indicates that high activation strengths do not by themselves ensure reliable semantics.

Experiment 4: analyzing the semantic consistency of interactions when the same SAE feature is activated at different positions. If the interactions encoded by the same SAE feature consistently convey similar meanings regardless of where the feature activates, then this feature is considered to represent consistent semantics.

We aim to clarify all the semantics modeled by the same SAE feature when it activates at different token positions. To this end, for each activation of the given feature, we recruit seven human evaluators to independently annotate its underlying semantic concepts by summarizing the meaning of the interactions extracted from this activation. The profiles of the human evaluators and the operating protocols are provided in Appendix J.

> For example, Figure 6 shows a case where a specific SAE feature is activated at three token positions: *history*, *of*, and *'*. By analyzing interactions, the evaluators find that these three activations mainly encode interaction patterns involving *history*-related tokens. Thus, all these three activations model the concept of "history". Appendix N provides more examples of the same SAE feature exhibiting similar semantics across different activations.

Typically, an SAE feature does not always represent the same semantics across different activations. Among all the semantics modeled by a given feature, we define the most frequent one—i.e., its modal semantic concept—as its **primary semantics**. When a higher proportion of activations convey the primary semantics, it indicates greater semantic consistency of the SAE feature across different token positions. Table 2 shows that, across three SAE models, 6.2% to 19.5% of the sampled activations successfully represent the primary semantics of their corresponding SAE features. Moreover, we find that some SAE features consistently express primary semantics across over 80% of sampled activations. Detailed statistics for each SAE feature are provided in Appendix L.

Finding 5: *SAE feature activations that represent the primary semantics tend to have lower PNCI scores.* Table 2 compares the PNCI scores of feature activations that represent the primary semantics with those that do not. We find that feature activations representing the primary semantics generally have lower PNCI scores, implying that the primary semantics of an SAE feature is usually carried by activations with less mutually cancelling interaction effects.

4 Conclusion

In this paper, we use interactions as numerically verifiable metrics to analyse subtle signals within SAE features and thus evaluate their semantic clarity. (1) Regarding the number of interactions, we find that 19.2%–35.6% of the sampled SAE feature activations encode non-sparse interaction patterns, indicating mixed semantics. Moreover, the average number of interactions varies considerably across SAE features and shows no clear relationship with activation frequency. (2) Regarding interaction complexity, we find that if an SAE feature activation encodes high-density interactions, these interactions tend to represent complex semantics. (3) Regarding mutual cancellation of interactions, we find that 25.2%–41.9% of the sampled SAE feature activations encode interactions with severe mutually cancelling effects, which may indicate less coherent semantics. (4) Finally, regarding semantic consistency, we observe that 6.2%–19.5% of sampled feature activations represent their human-annotated primary semantics, and these activations also encode fewer mutually cancelling interactions. We admit that these findings are still preliminary, as they are established on a small sampled subset of features and activations; nevertheless, we hope they offer a new perspective on the semantic clarity of SAE features.

REFERENCES

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

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

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Landon Butler, Abhineet Agarwal, Justin Singh Kang, Yigit Efe Erginbas, Bin Yu, and Kannan Ramchandran. ProxySPEX: Inference-efficient interpretability via sparse feature interactions in LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=KI8qan2EA7.

Lu Chen, Siyu Lou, Benhao Huang, and Quanshi Zhang. Defining and extracting generalizable interaction primitives from dnns. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Representation Learning*, volume 2024, pp. 23780–23802, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/67a9b444cbcd647572c88194619f72d5-Paper-Conference.pdf.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 16318–16352. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16124–16170, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.893. URL https://aclanthology.org/2023.acl-long.893/.

Huiqi Deng, Qihan Ren, Hao Zhang, and Quanshi Zhang. DISCOVERING AND EXPLAINING THE REPRESENTATION BOTTLENECK OF DNNS. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=iRCUlgmdfHJ.

Huiqi Deng, Na Zou, Mengnan Du, Weifu Chen, Guocan Feng, Ziwei Yang, Zheyang Li, and Quanshi Zhang. Unifying fourteen post-hoc attribution methods with taylor interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4625–4640, 2024. doi: 10.1109/TPAMI.2024.3358410.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.3. URL https://aclanthology.org/2022.emnlp-main.3/.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models, 2024. URL https://arxiv.org/abs/2401.06102.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang

Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Alex Gulko, Yusen Peng, and Sachin Kumar. CE-bench: Towards a reliable contrastive evaluation benchmark of interpretability of sparse autoencoders. In Yonatan Belinkov, Aaron Mueller, Najoung Kim, Hosein Mohebbi, Hanjie Chen, Dana Arad, and Gabriele Sarti (eds.), *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 1–15, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-346-3. doi: 10.18653/v1/2025.blackboxnlp-1.1. URL https://aclanthology.org/2025.blackboxnlp-1.1/.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, et al. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*, 2024.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spacy: Industrial-strength natural language processing in python. 2020. doi: 10.5281/zenodo.1212303. URL https://doi.org/10.5281/zenodo.1212303.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Justin Singh Kang, Landon Butler, Abhineet Agarwal, Yigit Efe Erginbas, Ramtin Pedarsani, Bin Yu, and Kannan Ramchandran. SPEX: Scaling feature interaction explanations for LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=pRlKbAwczl.

Mingjie Li and Quanshi Zhang. Does a neural network really encode symbolic concepts? In *ICML*, pp. 20452–20469, 2023. URL https://proceedings.mlr.press/v202/li23at.html.

Mingjie Li and Quanshi Zhang. Technical note: Defining and quantifying and-or interactions for faithful and concise explanation of dnns, 2024. URL https://arxiv.org/abs/2304.13312.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024. URL https://arxiv.org/abs/2408.05147.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Siyu Lou, Yao Yan, Yuntian Chen, and Quanshi Zhang. Cross-llm consistency in inference: Evidence from shared interactions, 2026. URL https://arxiv.org/abs/2606.08129.

Yin Lu, Xuening Zhu, Tong He, and David Wipf. Sparse autoencoders, again? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=wxU2LuTE74.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving sparse decomposition of language model activations with gated sparse autoencoders. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 775–818. Curran Associates, Inc., 2024a. doi: 10.52202/079017-0024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/01772a8b0420baec00c4d59fe2fbace6-Paper-Conference.pdf.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders, 2024b. URL https://arxiv.org/abs/2407.14435.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016. URL https://arxiv.org/abs/1606.05250.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Defining and quantifying the emergence of sparse concepts in dnns. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20280–20289, 2023a. doi: 10.1109/CVPR52729.2023.01942.

Jie Ren, Zhanpeng Zhou, Qirui Chen, and Quanshi Zhang. Can we faithfully represent absence states to compute shapley values on a DNN? In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=YV8tP7bW6Kt.

Qihan Ren, Huiqi Deng, Yunuo Chen, Siyu Lou, and Quanshi Zhang. Bayesian neural networks avoid encoding complex and perturbation-sensitive concepts. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28889–28913. PMLR, 23–29 Jul 2023c. URL https://proceedings.mlr.press/v202/ren23a.html.

Qihan Ren, Jiayang Gao, Wen Shen, and Quanshi Zhang. Where we have arrived in proving the emergence of sparse interaction primitives in dnns. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Representation Learning*, volume 2024, pp. 50323–50356, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/db0ee27cb50dd9087b133f6e7d28a90e-Paper-Conference.pdf.

Qihan Ren, Junpeng Zhang, Yang Xu, Yue Xin, Dongrui Liu, and Quanshi Zhang. Towards the dynamics of a DNN learning symbolic interactions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=dIHXwKjXRE.

Xiangchen Song, Aashiq Muhamed, Yujia Zheng, Lingjing Kong, Zeyu Tang, Mona T. Diab, Virginia Smith, and Kun Zhang. Mechanistic interpretability should prioritize feature consistency in sparse autoencoders. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2172–2210, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.99. URL https://aclanthology.org/2026.acl-long.99/.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3319–3328. JMLR.org, 2017.

SymtrustAI Co., Ltd. SymTrustAI: The world's first verifiable mechanistic diagnosis platform for ai model, 2026. URL https://www.symtrustai.com/en/.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. Explaining generalization power of a dnn using interactive concepts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):17105–17113, Mar. 2024. doi: 10.1609/aaai.v38i15.29655. URL https://ojs.aaai.org/index.php/AAAI/article/view/29655.

A THE USE OF LARGE LANGUAGE MODELS (LLMS)

This paper used large language models only for language polishing. The authors first wrote an English draft of the paper themselves, then asked an LLM to polish the language, and finally checked the entire manuscript themselves.

B LIMITATIONS

The interaction-based explanation method employed in this paper suffers from high computational cost: we have to compute a distinct set of interactions for each SAE feature activation. As illustrated in Appendix G, for an input prompt with $n = 10$ input variables, the interaction extraction algorithm needs to optimize $2^{n+1} = 2048$ parameters. This computational burden restricts our experiments to using only a limited number of tokens as interaction players, and forces us to sample a relatively small set of SAE features and activations.

Here, we offer three remarks to contextualize this limitation:

(1) Our primary goal is not to propose a new mechanistic interpretability algorithm, but rather to evaluate the limitations of existing methods. Accordingly, instead of prioritizing computational efficiency, we focus on carefully verifying whether the semantics of SAE features are clear. Interactions provide a more fine-grained tool than simply analyzing activation strength (Bills et al., 2023), and represent a path that previous studies on SAE semantic clarity have not explored. We therefore hope this work serves as an inspiration for future research.

(2) In practical applications, we recommend that researchers first explore feature semantics using low-cost methods (e.g., OpenAI's automated interpretability (Bills et al., 2023)). Only when a specific feature is required for a downstream task should they apply interaction-based analysis for more detailed, quantitative verification, ensuring that its semantics are clear and reliable.

(3) In recent years, several techniques have been proposed for computing interactions more efficiently. We discuss them in the last paragraph of the Preliminaries section and in Appendix H.

C COMMON SAE ARCHITECTURES

In the Preliminaries section, an SAE is formulated as an encoder-decoder model that maps the dense LLM activation $\mathbf{a}_i \in \mathbf{R}^d$ to a sparse feature vector $\mathbf{h}_i \in \mathbf{R}^s$ ($s \gg d$), and reconstructs $\mathbf{a}_i$ as $\hat{\mathbf{a}}_i$. Different SAE architectures mainly differ in the choice of the non-linear activation function $\sigma(\cdot)$ and the sparsity penalty applied to $\mathbf{h}_i$. Below, we briefly review four common implementations under the same notation.

Standard SAE. The standard SAE (Bricken et al., 2023) uses the ReLU activation function in the encoder and imposes an ℓ_1-norm penalty on the feature activations:

$$\begin{aligned} &\min_{\mathbf{W}_{\text{enc}},\mathbf{b}_{\text{enc}},\mathbf{W}_{\text{dec}},\mathbf{b}_{\text{dec}}} \mathcal{L} = \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_2^2 + \lambda\|\mathbf{h}_i\|_1, \\ &\text{with } \begin{cases} \mathbf{h}_i = \mathrm{ReLU}(\mathbf{W}_{\text{enc}}\mathbf{a}_i + \mathbf{b}_{\text{enc}}), \\ \hat{\mathbf{a}}_i = \mathbf{W}_{\text{dec}}\mathbf{h}_i + \mathbf{b}_{\text{dec}}. \end{cases} \end{aligned} \tag{6}$$

The ℓ_1 penalty is a continuous relaxation of activation sparsity. However, because it penalizes the magnitude of activated features, it may introduce shrinkage in the feature activations.

TopK SAE. The TopK SAE (Gao et al., 2025) replaces ReLU with a TopK activation function, which keeps only the largest K positive pre-activations and sets all other dimensions to zero:

$$\begin{aligned} &\min_{\mathbf{W}_{\text{enc}},\mathbf{b}_{\text{enc}},\mathbf{W}_{\text{dec}},\mathbf{b}_{\text{dec}}} \mathcal{L} = \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_2^2, \\ &\text{with } \begin{cases} \mathbf{h}_i = \mathrm{TopK}(\mathbf{W}_{\text{enc}}\mathbf{a}_i + \mathbf{b}_{\text{enc}}; K), \\ \hat{\mathbf{a}}_i = \mathbf{W}_{\text{dec}}\mathbf{h}_i + \mathbf{b}_{\text{dec}}. \end{cases} \end{aligned} \tag{7}$$

This architecture directly controls the ℓ_0 sparsity level by fixing the number of activated features to at most K.

Gated SAE. The Gated SAE (Rajamanoharan et al., 2024a) separates the decision of whether a feature is activated from the estimation of its activation magnitude. Specifically, it uses a binary gating path to determine the active feature set, and a magnitude path to compute the corresponding

activation values:

$$\begin{aligned}
&\boldsymbol{\pi}_i = \mathbf{1}(\mathbf{W}_{\text{gate}}\mathbf{a}_i + \mathbf{b}_{\text{gate}} > 0)\,,\\
&\tilde{\mathbf{h}}_i = \mathrm{ReLU}(\mathbf{W}_{\text{mag}}\mathbf{a}_i + \mathbf{b}_{\text{mag}}),\\
&\mathbf{h}_i = \boldsymbol{\pi}_i \odot \tilde{\mathbf{h}}_i, \qquad \hat{\mathbf{a}}_i = \mathbf{W}_{\text{dec}}\mathbf{h}_i + \mathbf{b}_{\text{dec}},\\
&\mathcal{L} = \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_2^2 + \lambda\|\boldsymbol{\pi}_i\|_1.
\end{aligned} \tag{8}$$

By penalizing the gate $\boldsymbol{\pi}_i$ rather than the activation magnitude $\tilde{\mathbf{h}}_i$, this architecture encourages sparse feature selection while reducing shrinkage in the activated feature values.

JumpReLU SAE. The JumpReLU SAE (Rajamanoharan et al., 2024b) uses a thresholded ReLU activation function,

$$\mathrm{JumpReLU}_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{z} \odot \mathbf{1}(\mathbf{z} > \boldsymbol{\theta}), \tag{9}$$

where $\boldsymbol{\theta} \in \mathbf{R}^s$ denotes the learned threshold vector. The encoder-decoder form is

$$\begin{aligned}
&\min_{\mathbf{W}_{\text{enc}},\mathbf{b}_{\text{enc}},\mathbf{W}_{\text{dec}},\mathbf{b}_{\text{dec}},\boldsymbol{\theta}} \mathcal{L} = \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_2^2 + \lambda\|\mathbf{h}_i\|_0,\\
&\text{with } \begin{cases} \mathbf{h}_i = \mathrm{JumpReLU}_{\boldsymbol{\theta}}(\mathbf{W}_{\text{enc}}\mathbf{a}_i + \mathbf{b}_{\text{enc}}),\\ \hat{\mathbf{a}}_i = \mathbf{W}_{\text{dec}}\mathbf{h}_i + \mathbf{b}_{\text{dec}}. \end{cases}
\end{aligned} \tag{10}$$

The threshold $\boldsymbol{\theta}$ suppresses small positive pre-activations and allows the model to optimize an explicit ℓ_0 sparsity objective, commonly through gradient estimators for the discontinuous thresholding operation.

D Two Fundamental Properties of Interactions

Before introducing the two fundamental properties of interactions, we first provide a brief review of interaction-based explanation.

In prior work (Ren et al., 2023a), the objective is to explain a model's confidence in its prediction. Given an input sample $\mathbf{x} = [x_1, x_2, \cdots, x_n]^\top$ containing n variables (e.g., n image patches or tokens), whose indices are given by $N = \{1, 2, \cdots, n\}$, a DNN v predicts the probability $p(y = y^* \mid \mathbf{x})$ of classifying $\mathbf{x}$ to the ground-truth category y^*. The target score for explanation is typically defined as the log-odds of this probability:

$$v(\mathbf{x}) \overset{\text{def}}{=} \log \frac{p(y = y^*|\mathbf{x})}{1 - p(y = y^*|\mathbf{x})} \in \mathbb{R}. \tag{11}$$

Furthermore, the target $v(\mathbf{x})$ can be set to other scalar outputs without theoretical conflict, including regression scores or activation values of SAE features.

In order to explain the intricate inference logic of the DNN v, a logical function g composed of AND-OR interactions is constructed. For such an explanation to be faithful, it must satisfy two criteria. (i) **The fidelity criterion.** The logical function g must possess sufficient expressive capacity to accurately mimic the target function's outputs $v(\cdot)$ over a comprehensive set of input samples Ψ. (ii) **The conciseness criterion.** The function g should not be overly complex. It must use a relatively small set of interactions to explain the DNN's inference on a single input sample, making the explanation easy to understand. These two criteria can be formulated as follows:

$$\begin{aligned}
&\forall \mathbf{x}' \in \Psi,\ g(\mathbf{x}') = v(\mathbf{x}'),\\
&\text{subject to } \textit{complexity}(g) \le M,
\end{aligned} \tag{12}$$

where $M \in \mathbb{R}$ is an upper bound on the complexity of the logical function g.

(1) Specifically, the logical function g is represented as an AND-OR function.

$$\begin{aligned}
&\forall \mathbf{x}' \in \Psi,\\
&g(\mathbf{x}') \overset{\text{def}}{=} \sum_{S\in\Omega^{\text{AND}}} \underbrace{I_S^{\text{AND}} \cdot \mathbf{1}(\mathbf{x}' \text{ triggers AND relation in } S)}_{\text{an AND interaction}} +\\
&\qquad \sum_{S\in\Omega^{\text{OR}}} \underbrace{I_S^{\text{OR}} \cdot \mathbf{1}(\mathbf{x}' \text{ triggers OR relation in } S)}_{\text{an OR interaction}} + b,
\end{aligned} \tag{13}$$

where $\Omega^{\text{AND}} = \{S : S \subseteq N, S \neq \emptyset\}$ and $\Omega^{\text{OR}} = \{S : S \subseteq N, S \neq \emptyset\}$ represent the sets of AND and OR interactions, respectively, defined for the target function v on the input $\mathbf{x}$. $b \in \mathbb{R}$ is a scalar bias.

• **The AND interaction** is represented by the binary trigger function $\mathbf{1}(^{\mathbf{x}' \text{ triggers AND}}_{\text{relation in } S}) \in \{0, 1\}$, which indicates an AND relation (among the input variables in S) automatically modeled by the target function v. It outputs 1 if and only if all input variables in S co-appear in $\mathbf{x}'$. Removing/masking[6] any input variable in S will deactivate this interaction. $I_S^{\text{AND}} \in \mathbb{R}$ denotes the numerical effect exerted on $v(\mathbf{x}')$ when this AND interaction is triggered.

• **The OR interaction** is represented by the binary trigger function $\mathbf{1}(^{\mathbf{x}' \text{ triggers OR}}_{\text{relation in } S}) \in \{0, 1\}$. It outputs 1 if any input variable in the set S appears (not masked[6]) in $\mathbf{x}'$ and contributes $I_S^{\text{OR}} \in \mathbb{R}$ to $v(\mathbf{x}')$.

(2) The fidelity requirement is satisfied by the *universal matching property* in Theorem 1.

This property states that, with proper parameter assignment (interaction effects I_S^{AND} and I_S^{OR}), the logical function $g(\cdot)$ can exactly match the output of the DNN $v(\cdot)$ on every masked configuration[6] of the input sample $\mathbf{x}$. Here, the set of all masked samples is defined as $\Psi = \{\mathbf{x}_S \mid S \subseteq N\}$, where $\mathbf{x}_S$ denotes the sample in which variables in S retain their original values while all others ($N \setminus S$) are masked[6]. The set Ψ therefore exhaustively enumerates the 2^n masked samples.

Theorem 1 (Universal matching property, proven by Chen et al. (2024))**.** *Given a target function v and an input sample $\mathbf{x}$, set the scalar weights I_S^{AND} and I_S^{OR} in the logical function for all non-empty $S \subseteq N$ as $I_S^{\text{AND}} = \sum_{L \subseteq S}(-1)^{|S|-|L|} u_L^{\text{AND}}$ and $I_S^{\text{OR}} = -\sum_{L \subseteq S}(-1)^{|S|-|L|} u_{N \setminus L}^{\text{OR}}$. For all $L \subseteq N$, let $u_L^{\text{AND}} = 0.5 \cdot v(\mathbf{x}_L) + \gamma_L$ and $u_L^{\text{OR}} = 0.5 \cdot v(\mathbf{x}_L) - \gamma_L$, where $\{\gamma_L\}$ is a set of learnable parameters. Let $b = v(\mathbf{x}_\emptyset)$ and $\Psi = \{\mathbf{x}_S | S \subseteq N\}$. Then, we always have,*

$$\forall \mathbf{x}' \in \Psi, \quad g(\mathbf{x}') = v(\mathbf{x}'). \tag{14}$$

Since the AND-OR interactions are sensitive to small noise present in the DNN output $v(\cdot)$ (Zhou et al., 2024), Li & Zhang (2024) proposed introducing learnable parameters $\{\delta_L\}$ to mitigate its effect. Specifically, the computation of u_L^{AND} and u_L^{OR} in the above theorem is modified as follows: set $\delta_\emptyset = 0$ for the baseline state and, for non-empty $L \subseteq N$, let $u_L^{\text{AND}} = 0.5 \cdot (v(\mathbf{x}_L) - \delta_L) + \gamma_L$ and $u_L^{\text{OR}} = 0.5 \cdot (v(\mathbf{x}_L) - \delta_L) - \gamma_L$, with $|\delta_L| < \epsilon$. Under this baseline convention, they further proved that

$$\forall \mathbf{x}' \in \Psi, \ |g(\mathbf{x}') - v(\mathbf{x}')| < \epsilon, \tag{15}$$

where the small matching-error bound $\epsilon \in \mathbb{R}^+$ guarantees the fidelity of the explanation.

(3) The conciseness requirement is satisfied by the *sparsity property*. Following (Ren et al., 2024a), the complexity of the logical function g is defined as the total number of salient interactions it encodes. Under three common conditions (see Appendix F), prior work proves that interactions for a target function are inherently sparse. That is, only $\mathcal{O}(n^\xi/\tau)$ interactions have numerical effects greater than τ, which is significantly less than the total interaction number 2^n. Empirically, ξ is in the range of $[0.9, 1.2]$. We treat this result as theoretical motivation rather than an unconditional guarantee for intermediate SAE activations, and test whether the resulting sparsity pattern is observed empirically in our SAE experiments. Other interactions have negligible influence on the target function when these conditions are satisfied.

When the matching identity holds and sparse salient interactions are observed, a logical function built from just a few salient interactions can accurately mimic the DNN's outputs across the exponentially large set of masked inputs Ψ. In this setting, these interactions provide concise representations of the target function's primitive inference patterns.

Based on this, Li & Zhang (2024) have proposed an algorithm to simultaneously extract AND-OR interactions. See Appendix G for details.

[6]A masked state of the input sample $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$ is generated by substituting the original values of some variables x_i with the baseline value to remove their information. For Transformers, commonly used baseline values include zero, mean embeddings, PAD tokens, etc. See (Ren et al., 2023b) for detailed discussion.

E SATURATION OR REDUNDANCY DOES NOT INFLATE THE PNCI

One may be concerned that a negative interaction effect reflects saturation (i.e., redundancy between tokens) rather than genuinely conflicting semantics. For example, consider two tokens a and b with $v(\mathbf{x}_\emptyset) = 0$, $v(\mathbf{x}_{\{a\}}) = 10$, $v(\mathbf{x}_{\{b\}}) = 10$, and $v(\mathbf{x}_{\{a,b\}}) = 12$. A pure-AND (Shapley-style) decomposition yields the interaction effect $I_{\{a,b\}} = 12 - 10 - 10 + 0 = -8$, which appears to indicate strong positive-negative cancellation, even though a and b may simply encode redundant, similar meanings.

This concern is resolved by the AND-OR decomposition in Eq. equation 13. Owing to the learnable parameters $\{\gamma_L\}$ in Theorem 1, the decomposition admits free reparameterization between the AND and OR channels, which allows redundant (sub-additive) relations to be absorbed by positive OR effects instead of negative AND effects. For the above example, the AND-OR decomposition can represent the same function as a single OR interaction $I^{\text{OR}}_{\{a,b\}} = +10$ plus a small AND interaction $I^{\text{AND}}_{\{a,b\}} = +2$, with all singleton effects equal to zero. This assignment exactly reproduces $v(\cdot)$ on all four masked configurations, and it yields $\text{PNCI} = 1 - 12/12 = 0$ (Eq. equation 5), correctly indicating that there is no cancellation.

More generally, any saturating (sub-additive) relation with $\max\big(v(\mathbf{x}_{\{a\}}), v(\mathbf{x}_{\{b\}})\big) \le v(\mathbf{x}_{\{a,b\}}) \le v(\mathbf{x}_{\{a\}}) + v(\mathbf{x}_{\{b\}})$ admits a fully non-negative AND-OR representation[7], so saturation alone yields a near-zero PNCI. In contrast, genuine suppression—where the co-occurrence of two tokens contributes less than either token alone, i.e., $v(\mathbf{x}_{\{a,b\}}) < \max\big(v(\mathbf{x}_{\{a\}}), v(\mathbf{x}_{\{b\}})\big)$—cannot be reproduced by any non-negative AND-OR assignment, and therefore necessarily introduces negative effects and raises the PNCI. Hence, under the AND-OR decomposition, the PNCI separates conflicting or suppressive patterns from redundant ones, while redundancy is instead reflected by the strength of OR interactions. For instance, in Figure 1 of the main paper, the redundant relation between *safe* and *heaven* is encoded as a positive OR interaction ($I^{\text{OR}}_{\{safe,heaven\}} = +2.52$), not as a negative effect.

In practice, the interaction extraction algorithm (Li & Zhang, 2024) learns the parameters $\{\gamma_L\}$ under a sparsity objective, so redundant effects tend to be assigned to the OR channel rather than being split into cancelling AND effects. This is why we read the PNCI in Experiment 3 as a measure of genuine positive-negative cancellation, while regarding strong OR interactions as an indicator of synonymous or redundant realizations of similar semantics.

F COMMON CONDITIONS FOR THE SPARSITY PROPERTY OF INTERACTIONS

Ren et al. (2024a) proved that, under three common conditions, a trained DNN usually encodes very sparse interactions. These conditions were formulated for output confidence functions and are not assumed to hold exactly for intermediate SAE activations; in this paper, they serve as motivating background, while the persistence of sparse interactions for SAE features is assessed empirically. Given an input $\mathbf{x} \in \mathbb{R}^n$ and a DNN, let $c(\mathbf{x})$ denote the confidence score of the DNN output on $\mathbf{x}$. We compute the average classification confidence $C_m(\mathbf{x}) = \mathbb{E}_{T\subseteq N:|T|=n-m}[c(\mathbf{x}_T)]$ on masked inputs with m input variables being masked. The three conditions are as follows. (i) The average classification confidence C_m monotonically decreases when more input variables are masked, i.e., if $m > m'$, then $C_m < C_{m'}$; (ii) The average inference confidence does not decrease faster than a polynomial function of the ratio of masked input variables; (iii) The high-order derivatives of the network output with respect to the input are all zero.

G PROCEDURE OF THE INTERACTION EXTRACTION ALGORITHM

An algorithm to simultaneously extract AND-OR interactions (I^{AND}_S and I^{OR}_S) is proposed by Li & Zhang (2024) and has been employed by many studies (Ren et al., 2024b; Lou et al., 2026).

[7] Assuming $0 \le v(\mathbf{x}_{\{a\}}) \le v(\mathbf{x}_{\{b\}})$, set $I^{\text{OR}}_{\{a,b\}} = v(\mathbf{x}_{\{a\}})$, $I^{\text{OR}}_{\{b\}} = v(\mathbf{x}_{\{b\}}) - v(\mathbf{x}_{\{a\}})$, and $I^{\text{AND}}_{\{a,b\}} = v(\mathbf{x}_{\{a,b\}}) - v(\mathbf{x}_{\{b\}})$; all three effects are non-negative, and they exactly reproduce $v(\cdot)$ on all four masked configurations.

For any masked state $\mathbf{x}_L$, this algorithm aims to learn a decomposition $v(\mathbf{x}_L) = u_L^{\text{AND}} + u_L^{\text{OR}}$ that makes the extracted interactions sparsest. The component u_L^{AND} is explained by AND interactions and u_L^{OR} is explained by OR interactions. Specifically, the decomposition takes the form $u_L^{\text{AND}} = 0.5 \cdot v(\mathbf{x}_L) + \gamma_L$ and $u_L^{\text{OR}} = 0.5 \cdot v(\mathbf{x}_L) - \gamma_L$, where $\{\gamma_L \mid L \subseteq N\}$ is a set of learnable parameters. In this way, the AND-OR interactions can be computed by applying Theorem 1 to these components, i.e., for every non-empty $S \subseteq N$, $I_S^{\text{AND}} = \sum_{L \subseteq S} (-1)^{|S|-|L|} u_L^{\text{AND}}$ and $I_S^{\text{OR}} = -\sum_{L \subseteq S} (-1)^{|S|-|L|} u_{N \setminus L}^{\text{OR}}$.

Sparsity optimization. To learn a decomposition that makes the extracted interactions sparsest, the parameter set $\{\gamma_L \mid L \subseteq N\}$ is optimized using the following LASSO-like loss function. The empty set is reserved for the baseline bias and is not treated as an interaction.

$$\min_{\{\gamma_L\}} \sum_{\substack{S \subseteq N \\ S \neq \emptyset}} \left(|I_S^{\text{AND}}| + |I_S^{\text{OR}}| \right). \tag{16}$$

Noise mitigation. Interactions, especially high-order ones, are found to be sensitive to small noises in the model's outputs (Zhou et al., 2024), thus Li & Zhang (2024) further propose to learn to remove these small noises. Specifically, the above decomposition is revised to $u_L^{\text{AND}} = 0.5(v(\mathbf{x}_L) - \delta_L) + \gamma_L$ and $u_L^{\text{OR}} = 0.5(v(\mathbf{x}_L) - \delta_L) - \gamma_L$ for non-empty L, while $\delta_\emptyset = 0$. A tiny noise bound $[-\epsilon, \epsilon]$ is employed to constrain the non-baseline parameters $\{\delta_L\}$, where typically $\epsilon = 0.02 \cdot \mathbf{E}_{\mathbf{x}}|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$. Then the parameter sets $\{\delta_L \mid L \subseteq N, L \neq \emptyset\}$ and $\{\gamma_L \mid L \subseteq N\}$ are optimized using the following LASSO-like loss function.

$$\min_{\{\delta_L\},\{\gamma_L\}} \sum_{\substack{S \subseteq N \\ S \neq \emptyset}} \left(|I_S^{\text{AND}}| + |I_S^{\text{OR}}| \right). \tag{17}$$

Pseudocode for computing I_S^{AND} and I_S^{OR} is provided in Algorithm 1. Parameter settings for the experiments in this paper are provided in Appendix I.5. Partially open-source code and its descriptions are provided in the supplementary material.

H Methods to Lower the Computational Cost of Interactions

H.1 Efficient computation of interactions

The computational complexity of computing Harsanyi interactions scales exponentially with the number of input variables n. To make interaction-based explanations feasible for real-world applications, existing studies reduce the computational cost primarily from two perspectives: developing approximation algorithms and reducing the number of input variables through engineering techniques.

Approximation algorithms. Approximation algorithms have been proposed to efficiently compute Harsanyi interactions (Kang et al., 2025; Butler et al., 2026). These algorithms enable interaction-based explanations to be widely adopted in various industrial applications (SymtrustAI Co., Ltd., 2026).

Reducing the number of input variables. Engineering techniques reduce the computational cost by decreasing the number of input variables n. For image classification tasks, previous studies divide an input image into different patches and treat each patch as a single input variable (Li & Zhang, 2023). For natural language processing (NLP) tasks, treating all tokens in an input prompt as individual variables is unnecessary. Instead, individual words, phrases, or sentence fragments can be considered as single input variables. Attribution methods can guide the selection of these input variables. Methods such as integrated gradients (Sundararajan et al., 2017) and SHAP (Lundberg & Lee, 2017) have shown that a small number of input variables is often sufficient to cover the majority of attributions that influence the model's output. By selecting important words, phrases, or sentences based on these methods, the number of input variables can be substantially reduced. Deng et al. (2022) have adopted this kind of input variable selection method in their study.

Implementation in this paper. Specifically, this work employs a straightforward yet effective approach to selecting input variables for sparse autoencoder (SAE) features. Up to 10 nearest content

Algorithm 1 Compute AND and OR Interactions and Select Salient Ones

Require: Network function v, input sample $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$, set of indices $N = \{1, 2, \ldots, n\}$, small noise bound ϵ, salient threshold τ, number of training epochs E.

Ensure: AND interactions I_S^{AND}, OR interactions I_S^{OR}, the significant interaction sets $\Omega_{\text{sparse}}^{\text{AND}}$ and $\Omega_{\text{sparse}}^{\text{OR}}$, and the bias b in the logical model.

Initialize learnable parameters $\{\gamma_L\}$ for all $L \subseteq N$ and $\{\delta_L\}$ for all non-empty $L \subseteq N$; set $\delta_\emptyset = 0$.
Compute all masked samples $\mathbf{x}_L$ and their static outputs $v(\mathbf{x}_L)$ for $L \subseteq N$, including the baseline $v(\mathbf{x}_\emptyset)$.
Set $u_\emptyset^{\text{AND}} \leftarrow 0.5\, v(\mathbf{x}_\emptyset) + \gamma_\emptyset$ and $u_\emptyset^{\text{OR}} \leftarrow 0.5\, v(\mathbf{x}_\emptyset) - \gamma_\emptyset$.
for $e = 1$ **to** E **do**
  for all non-empty subset $L \subseteq N$ **do**
    Reuse the precomputed output $v(\mathbf{x}_L)$.
    Compute noise term δ_L constrained in $[-\epsilon, \epsilon]$.
    Decompose $v(\mathbf{x}_L)$ into AND and OR components:
    $u_L^{\text{AND}} \leftarrow 0.5 \cdot (v(\mathbf{x}_L) - \delta_L) + \gamma_L$
    $u_L^{\text{OR}} \leftarrow 0.5 \cdot (v(\mathbf{x}_L) - \delta_L) - \gamma_L$
  end for
  for all non-empty subset $S \subseteq N$ **do**
    Compute AND interaction I_S^{AND}:

$$I_S^{\text{AND}} \leftarrow \sum_{L \subseteq S} (-1)^{|S|-|L|} u_L^{\text{AND}}$$

    Compute OR interaction I_S^{OR}:

$$I_S^{\text{OR}} \leftarrow -\sum_{L \subseteq S} (-1)^{|S|-|L|} u_{N \setminus L}^{\text{OR}}$$

  end for
  Merge first-order effects: for $|S| = 1$, set $I_S^{\text{AND}} \leftarrow I_S^{\text{AND}} + I_S^{\text{OR}}$ and $I_S^{\text{OR}} \leftarrow 0$.
  Compute current loss $\mathcal{L} \leftarrow \sum_{\substack{S \subseteq N \\ S \neq \emptyset}} \left(|I_S^{\text{AND}}| + |I_S^{\text{OR}}| \right)$.
  Optimize parameters $\{\gamma_L\}$ and $\{\delta_L\}$ to minimize $\mathcal{L}$.
end for
Let bias $b = v(\mathbf{x}_\emptyset)$.
Select significant AND interactions: $\Omega_{\text{sparse}}^{\text{AND}} \leftarrow \{S \subseteq N, S \neq \emptyset : |I_S^{\text{AND}}| > \tau\}$
Select significant OR interactions: $\Omega_{\text{sparse}}^{\text{OR}} \leftarrow \{S \subseteq N, S \neq \emptyset : |I_S^{\text{OR}}| > \tau\}$
return I_S^{AND}, I_S^{OR}, $\Omega_{\text{sparse}}^{\text{AND}}$, $\Omega_{\text{sparse}}^{\text{OR}}$, and b.

words preceding the token position where an SAE feature activates are selected as the input variables. To verify the feature's dependency on these variables during sampling, we ensure that the sampled SAE feature activations do not activate when all selected input variables are masked. This procedure helps ensure that the selected variables account for the feature activation.

I Detailed Experimental Settings

In Appendix K, we analyze the robustness of our main findings to various hyperparameters.

I.1 Selection of SAEs

We focus on SAEs trained on the MLP output and residual stream of the middle layers of LLMs, as many studies (Gulko et al., 2025; Templeton et al., 2024) suggest that SAEs built upon these locations tend to be more interpretable.

Our experiments are conducted on three open-source SAEs, including: (i) a JumpReLU SAE from the Gemma-Scope series (Lieberum et al., 2024), built upon the MLP output at layer 12 of Gemma-2-2B (Team et al., 2024), with a dictionary size of 65K and an average L_0 sparsity of 96; (ii) a JumpReLU SAE from the Gemma-Scope series, built upon the residual stream at layer 20 of Gemma-2-9B, with 16K features and average L_0 sparsity of 68; (iii) an improved TopK SAE from the LLaMA-Scope series (He et al., 2024), built upon the residual stream at layer 15 of LLaMA-3.1-8B (Grattafiori et al., 2024), with 32K features and $K = 50$ sparsity.

Artifact licenses. Gemma models are used under the Gemma Terms of Use; LLaMA model is used under the Llama 3.1 Community License Agreement; Gemma-Scope is used under CC BY 4.0; LLaMA-Scope is used under Apache 2.0. All artifacts are used consistently with their intended purposes. Gemma and LLaMA models are used for interpretability research, which aligns with their intended use as open models for research. Gemma-Scope and Llama-Scope are specifically designed for interpretability studies.

I.2 Details about the Dataset

We extract sentences from the SQuAD v1.1 (Rajpurkar et al., 2016) to construct our experimental dataset in the following manner.

(1) We select the first 1,000 passages from the SQuAD v1.1 development set. For each passage, the text is first normalized (e.g., converting curly quotes to straight quotes, em/en dashes to hyphens, etc.) and then segmented into subsentences. Starting from the first subsentence, words are sequentially collected until 13 content words are obtained. A content word is defined as any word that is not a stop word (specifically, we use stop words in spaCy (Honnibal et al., 2020)), punctuation, or whitespace. Compound expressions, such as hyphenated or slash-separated terms (e.g., "EOR-LOR", "APOLLO/Stature") and units like temperature notations (e.g., "25 degrees C"), are merged into single words provided no spaces occur between their components.

(2) We aim to extract two prompts from each passage. After extracting the first prompt (ending at the 13th content word), the second prompt begins from the next subsentence following the last one used in the first extraction. If the remaining subsentences contain insufficient content words to form a complete prompt of 13 content words, only one prompt is extracted from that passage.

(3) For each input prompt, following Ren et al. (2023b), we regard all content words as valid input variables. That is, during interaction calculation, only these valid input variables are masked, while other "background" words remain unchanged. Due to the high computational cost of calculating interactions among more than 10 input variables, we only compute interactions among at most the 10 input variables closest to the token position where the SAE feature is triggered. Thus, the maximum number of input variables in the set N is 10.

Ultimately, we compile a dataset comprising 1864 prompts, about 56,000 tokens in total. On average, each prompt contains 23 words, among which 13 are content words.

I.3 Selection of SAE Features and Corresponding Activations.

We select a total of 21 features per SAE. 20 of these are sampled uniformly across the activation frequency interval of $[0.1\%, 10\%]$, where activation frequency is defined as the proportion of token positions that cause the given feature to activate. This frequency range is chosen because features with excessively high activation frequencies tend to exhibit no clear semantics, whereas those with excessively low frequencies provide too few samples for reliable experimental analysis. An additional feature is randomly selected from the narrow range of $[9\%, 11\%]$ to include a representative high-frequency feature.

To ensure that the semantics of a feature are fully captured by the selected input variables used for interaction computation, we only retain feature activations that are non-zero when all tokens are present ($v(\mathbf{x}_N) \neq 0$) but become zero when all inputs are masked ($v(\mathbf{x}_\emptyset) = 0$). Since computing interactions for all activations of an SAE feature is infeasible, we sample 30 feature activations. Specifically, we keep the top 5 activations with the largest unmasked outputs $v(\mathbf{x}_N)$. From the remainder, we uniformly sample 25 according to their activation values. This approach preserves both high-response instances and a representative distribution of activation values while remaining computationally tractable.

I.4 Adapting SAE Feature Activation Functions for Interaction Extraction

We aim to explain the scalar activation of an SAE feature at a given token position, which we call the SAE feature activation and denote by $v(\cdot) \in \mathbb{R}$. To compute interactions, we mask the input prompt $\mathbf{x}$ to obtain a set of masked states Ψ, and collect $v(\mathbf{x}')$ for every masked state $\mathbf{x}' \in \Psi$.

In a standard SAE, this feature activation value is obtained by applying a nonlinear activation function σ to the pre-activation value, i.e., $v(\mathbf{x}') = \sigma(\text{pre-activation})$. Commonly used σ functions include threshold-based operators such as JumpReLU or TopK, which forcibly set small positive pre-activations to zero. This sudden drop to zero introduces output noise to $v(\mathbf{x}')$, disrupts the sparsity of interactions, and leads to the disproportionate amplification of noisy interactions. Indeed, the sensitivity of interaction computation to output noise has already been highlighted by (Zhou et al., 2024) in experiments interpreting LLMs.

To avoid the small output noise introduced by threshold-based activation functions, we replace σ with a standard ReLU when defining $v(\cdot)$, i.e., $v(\mathbf{x}') = \mathrm{ReLU}(\text{pre-activation})$. We choose ReLU for two main reasons: (1) it preserves the sign of the pre-activation: negative values indicate misalignment with the feature direction and are set to zero, whereas positive values are kept unchanged; and (2) it restores small positive activations that would otherwise be thresholded to zero, thereby eliminating the noise they introduce and stabilizing the interaction computation. This substitution is applied only after the SAE has been trained, during interaction extraction; it does not alter the SAE feature dictionary or the feature extraction process itself, but reuses information in the pre-activation that native thresholding suppresses. Indeed, this substitution does not change the semantics learned by the SAE: much like PCA, the SAE encoder essentially learns a direction vector that represents certain semantics from the LLM activation vectors, and the pre-activation is exactly the projection of an LLM activation vector onto this direction, i.e., the true triggering strength of the semantics on the given input. Threshold-based operators such as TopK or JumpReLU only sparsify the features extracted on each sample, while the learned direction vector is what actually carries the feature's semantics. Therefore, replacing them with ReLU instead restores the true triggering degree of the feature on any input, rather than regarding a weakly triggered feature as not triggered; forcibly setting such weak responses to zero during interaction extraction introduces unreliable noise into the analysis, as verified by the comparison below.

To validate the effectiveness of the above substitution, we compare the experimental results on the SAE for Gemma-2-9B before and after substituting the activation function. Table 3 demonstrates that replacing the activation function with ReLU dramatically reduces the interaction number. Furthermore, Figure 7 reveals that the threshold-based activation function σ increases the number of complex interactions (orders ≥ 4). These complex interactions exhibit a spindle-shaped distribution and suffer from severe positive-negative cancellation, representing typical noise patterns (Ren et al., 2024b).

Activation setting	Low interaction density group (encoding ≤10 interactions)	Medium interaction density group (encoding 11–20 interactions)	High interaction density group (encoding ≥21 interactions)
SAE with ReLU function	80.8%	13.3%	5.9%
SAE with σ function	57.0%	11.0%	32.0%

Table 3: Distribution of SAE feature activations grouped by interaction counts for the Gemma-2-9B SAE with threshold-based activation function σ and with ReLU activation function.

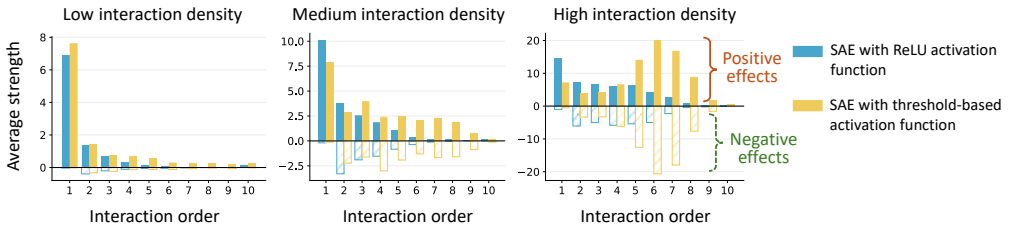


Figure 7: Histograms of the total interaction strength across different orders for each SAE feature activation group. Positive and negative strength are computed separately.

Please note that this substitution is used solely for interaction computation and the numerical verification of interaction-based explanations. It is not applied to other experimental data, such as the statistics of SAE activation values and frequencies.

I.5 Parameter Settings for Interaction Extraction

A detailed procedure of the interaction extraction algorithm is provided in Appendix G; here we only provide the parameter settings. We provide a further discussion on the following interaction extraction parameters in Appendix K.

(i) The matching-error bound ϵ is set to $0.04 \cdot \mathbf{E}_v|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$ for all SAEs.

(ii) The salience threshold τ is set to $2 \cdot \epsilon$ to select salient interactions. All other interactions are treated as noise patterns and thus excluded from our analysis.

(iii) For most activation functions, the training loss converges after 40,000 steps, at which point the interactions remain stable with negligible change. Given this convergence, repeated optimization runs or averaging over multiple trials are unnecessary.

(iv) Since first-order AND and OR interactions share identical triggering conditions, we merge the effect of each first-order OR interaction into its corresponding AND interaction, i.e., for all $S \subseteq N$ with $|S| = 1$, we set $I_S^{\text{AND}} \leftarrow I_S^{\text{AND}} + I_S^{\text{OR}}$.

(v) The scalar bias b is not a hyperparameter; it is simply the feature's response on the fully masked input, i.e., $b = v(\mathbf{x}_\emptyset)$, which is excluded from the interaction set $\Omega^{\text{AND}} \cup \Omega^{\text{OR}}$ and thus affects neither the interaction counts nor the PNCI. Moreover, τ is normalized per activation by $\mathbf{E}_v|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$, and our activation selection (Appendix I.3) retains only activations with $v(\mathbf{x}_\emptyset) = 0$, so $b = 0$ holds for all analyzed activations.

I.6 Model Size and Computational Budget

Model names specify their parameter sizes (e.g., Gemma-2-2B: 2B parameters). Our largest SAE has 300M parameters, negligible compared to the base models. Experiments were conducted on NVIDIA A800 GPUs. Inference over all $\mathbf{x}' \in \Psi$ ($\leq$1024 samples) is computationally trivial, and 40,000-step optimization per SAE feature activation completes in about 40 seconds. Interaction optimization itself requires minimal GPU resources (memory, utilization, power), thus enabling parallel computation across many different feature activations.

J Details about Semantic Consistency Analysis

J.1 The Profiles of Human Evaluators.

To examine whether the interactions encoded by a given SAE feature consistently convey similar meanings regardless of its activation context, we recruit seven paid undergraduate evaluators. All evaluators are fluent in English and are provided only with a basic explanation of sparse autoencoders and interactions. We deliberately employ non-expert annotators because clear, interpretable semantics should be comprehensible to most laypeople, not solely to domain experts. In fact, OpenAI also relies primarily on non-expert evaluators when investigating the semantics modeled by neurons (Bills et al., 2023).

Given that the annotation task focuses on summarizing the semantics of word groups, we do not collect detailed demographic information. Although we do not conduct a formal consent procedure, participation is entirely voluntary, and evaluators are explicitly informed that their annotations will be used in research publications. All evaluators receive monetary compensation commensurate with local standards.

J.2 Instructions for Human Evaluators

We summarize the key instructions provided to the evaluators as follows.

(1) **Background.** We begin by introducing sparse autoencoders (SAEs) and their intended purpose. To illustrate the semantics captured by an individual SAE feature, we explain how we extract phrase patterns (termed *interactions*) from input prompts that most strongly drive the SAE feature's activation.

(2) **Explanation of interactions.** We clarify that AND interactions represent conjunctive relationships among words and should be interpreted as a cohesive semantic unit (e.g., $I^{\text{AND}}_{\{armed,rebellion\}}$ in Figure 16). OR interactions represent disjunctive relationships, where each word contributes independently to the activation (e.g., $I^{\text{OR}}_{\{safe,heaven\}}$ in Figure 1 of the main paper). Consequently, tokens with similar meanings are merged into a single OR interaction at extraction time, so different lexical realizations of the same concept do not inflate the interaction count. Accordingly, in the annotation task, evaluators are instructed to attribute interactions with different surface forms but the same meaning to the same semantic concept.

(3) **Annotation protocols.** Evaluators are instructed to prioritize interactions with large absolute strength (i.e., large $|I_S^{\text{AND}}|$ and $|I_S^{\text{OR}}|$), as these exert the greatest influence on the activation of the SAE feature. We further clarify that, although a feature's activation may be governed by dozens of interactions, these interactions typically concentrate on a small set of recurring words (see Figure 8). Consequently, in practice, evaluators only need to examine words that appear frequently among the most influential interactions. This procedure streamlines the annotation workflow and helps mitigate subjective bias.

(4) **Example.** To illustrate the annotation protocol, we guide the evaluators through the example presented in Figure 6 of the main paper. For the first feature activation examined, the ten highest-impact interactions revolve around words such as "key," "event," and "history," whereas other words appear only sporadically. Its semantics can thus be summarized as pertaining to "history" or "important events." Applying the same procedure to the second and third feature activations reveals that they all capture concepts related to "history." Thus, these three activations can be viewed as modeling the concept of "history". Appendix N provides more examples of the same SAE feature exhibiting similar semantics across different activations.

All evaluators are required to annotate the entire dataset. Throughout the task, they work independently and are instructed not to discuss their annotations with one another.

J.3 The Procedure of Semantic Consistency Evaluation.

If the interactions encoded by an SAE feature consistently convey similar meanings regardless of where the feature activates, the feature is considered to represent consistent semantics. However, an SAE feature often represents different semantics at different token positions. To quantify the

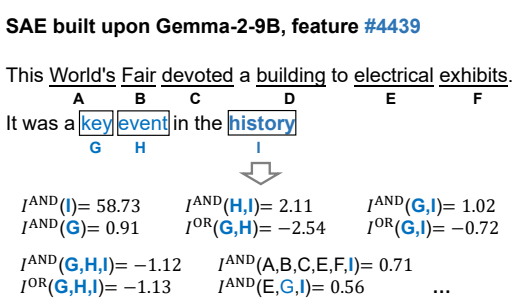


Figure 8: Different interactions extracted from one feature activation typically correspond to phrases composed of highly overlapping tokens/words, exhibiting relatively concentrated semantics that are easy to analyze.

Setting	Low interaction density (≤10 interactions)	Medium interaction density (11–20 interactions)	High interaction density (>20 interactions)
original	80.8%	13.3%	5.9%
average baseline	65.9%	14.8%	19.3%
smaller matching-error bound	76.3%	14.6%	9.1%
smaller salient threshold	57.0%	20.0%	23.0%

Table 4: Distribution of SAE feature activations by interaction count under four interaction computation settings. All results are obtained from the Gemma-2-9B SAE.

semantic consistency of a feature across different token positions, we identify the most frequent semantics—i.e., its modal semantic concept—as its primary semantics. We then use the proportion of activations that successfully express this primary semantics to quantify the SAE feature's semantic consistency.

(1) For each activation of a given SAE feature, we provide evaluators with the input prompt, mark the positions of the input variables, and list the extracted interactions (in a format similar to Figure 6 in the main paper, where interactions are shown as $I^{\text{AND}}(\text{armed}, \text{rebellion}) = 0.47$). Since a single activation can represent multiple semantic meanings, we ask evaluators to identify at most three of the most distinct ones. After annotating all activations of an SAE feature, we require the evaluators to report the primary semantics (i.e., the most frequent semantic concept shared across different activations), along with the set of feature activations that represent this primary semantics. If no shared semantics can be identified, evaluators are permitted to report an empty set.

(2) We then aggregate the feedback from the seven evaluators based on a majority-rule principle: a valid primary semantic concept must be agreed upon by more than half of the evaluators. This evaluator-level majority rule is distinct from the activation-level SCATP, which measures the proportion of activations assigned to the resulting modal concept. In cases of severe disagreement, the feature is considered to have no primary semantics. Specifically, for each SAE feature: (i) We discard any proposed primary semantics and the corresponding activation set that distinctly differ from those identified by the other evaluators. If fewer than four sets remain, we conclude that the feature lacks primary semantics. (ii) Using the remaining activation sets, we include an activation in the primary-semantics activation set $\mathcal{P}$ if it is selected by at least two evaluators.

(3) The primary semantics are determined by the authors after reviewing all evaluators' feedback. To further quantify how consistently a feature represents its primary semantics, we introduce the Semantic Consistency Across Token Positions (SCATP): SCATP $= \frac{|\mathcal{P}|}{|\mathcal{V}|} \times 100\%$ where $|\mathcal{V}|$ denotes the total number of activations for this feature. A higher SCATP indicates that the SAE feature exhibits greater semantic consistency across different token positions.

Appendix L reports the SCATP and primary semantics for all sampled SAE features. We do not report the primary semantics for SAE features with an SCATP $< 10\%$, as their semantics are highly inconsistent across different contexts, making them lack a common semantic meaning.

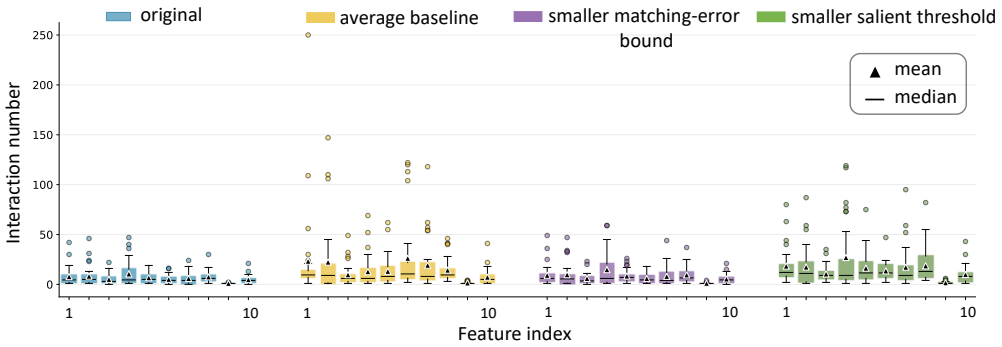


Figure 9: Distribution of interaction numbers encoded by ten randomly selected SAE features under four interaction computation settings for the Gemma-2-9B SAE. Within each setting, features are ordered by activation frequency, and each boxplot summarizes activations at different token positions.

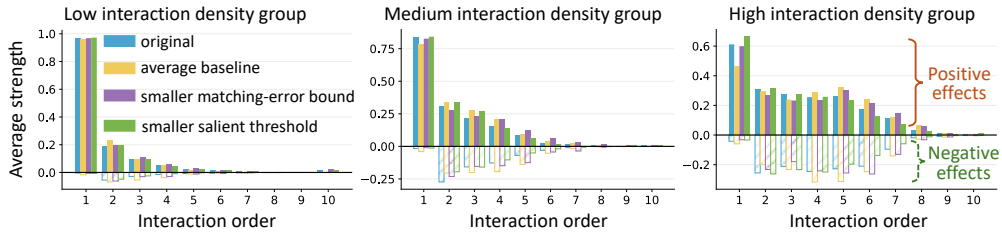


Figure 10: Interaction complexity distributions under four interaction computation settings for the Gemma-2-9B SAE. The low, medium, and high interaction density groups contain feature activations encoding ≤ 10, 11–20, and > 20 interactions, respectively. Positive and negative interaction strengths are computed separately, and each group-level mean complexity vector $\bar{\mathbf{d}}$ is normalized to unit ℓ_2-norm. $\overline{\text{PNCI}}$ denotes the mean PNCI of the corresponding group.

K Sensitivity Analysis of Main Findings to Hyperparameters

In this section, we analyze how hyperparameters influence the main conclusions of the paper. The hyperparameters can be broadly categorized into two classes: (1) Interaction computation hyperparameters, which include the mask baseline selection, the matching-error bound, and the salient threshold (detailed parameter descriptions are provided in Appendix I.5). (2) Semantic analysis hyperparameters, which include the partitioning of interaction density groups in Experiment 1, and the criteria for classifying minor and severe interaction cancellations in Experiment 3.

K.1 Hyperparameters for Interaction Computation

First, we discuss the rationale behind the selection of three hyperparameters for interaction computation and how they are adjusted for sensitivity analysis. Our sensitivity analysis is mainly conducted on the SAE applied to Gemma-2-9B, following the experimental settings in Appendix I.

Selection of the masking baseline. Masking is a commonly used strategy in attribution methods such as the Shapley value: a variable is erased so that its influence on the output can be measured. As described in Appendix I.2, the input variables in N are the at most 10 content words closest to the token position where the feature is triggered, i.e., $n = |N| \leq 10$, and the 2^n masked states are generated by replacing every subset of these variables with the baseline value while keeping all other words in the prompt unchanged. Consequently, $\mathbf{x}_\emptyset$ is not an empty input, but a complete sequence in which only the selected content words are masked, and the interaction decomposition is therefore conditioned on the retained context.

The choice of the baseline value used for masking can considerably affect which interactions are extracted. According to the optimal-masking theory (Ren et al., 2023b), improper baseline values

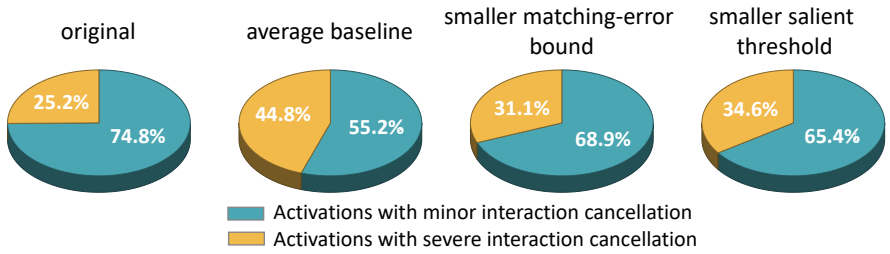


Figure 11: Proportions of SAE feature activations with minor interaction cancellation (PNCI ≤ 0.5) and severe interaction cancellation (PNCI > 0.5) under four interaction computation settings for the Gemma-2-9B SAE.

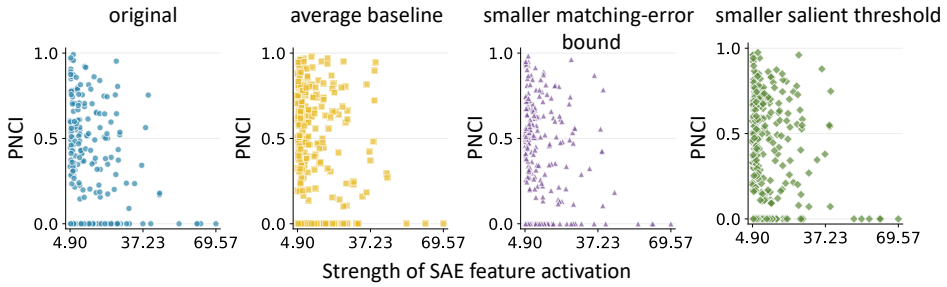


Figure 12: Scatter plots of SAE feature activation strength against the corresponding PNCI under four interaction computation settings for the Gemma-2-9B SAE. Each point represents one SAE feature activation.

make interactions capture spurious patterns. For example, consider a target function that contains only a single interaction, $v(\mathbf{x}) = w(x_p - \delta_p)(x_q - \delta_q)$, where the two variables take values $x_p = 3$ and $x_q = 4$, and $\delta_p = 2$ and $\delta_q = 3$ are the optimal baseline values for masking them. When the two variables are masked with exactly δ_p and δ_q, the function outputs zero whenever at least one variable is masked, i.e., on $\mathbf{x}_{\{p\}}$, $\mathbf{x}_{\{q\}}$, and $\mathbf{x}_\emptyset$, so the entire function is perfectly explained by a single AND interaction $I^{\text{AND}}_{\{p,q\}} = w$. In contrast, when both variables are masked with an incorrect value (e.g., 1), the four evaluations become $v(\mathbf{x}_{\{p,q\}}) = w$, $v(\mathbf{x}_{\{p\}}) = -2w$, $v(\mathbf{x}_{\{q\}}) = -w$, and $v(\mathbf{x}_\emptyset) = 2w$, yielding a non-zero bias $b = v(\mathbf{x}_\emptyset) = 2w$ plus three spurious AND interactions $I^{\text{AND}}_{\{p\}} = -4w$, $I^{\text{AND}}_{\{q\}} = -3w$, and $I^{\text{AND}}_{\{p,q\}} = 6w$ (computed as Harsanyi dividends under a pure-AND decomposition, i.e., $2^2 = 4$ non-zero effects in total), none of which correspond to any real structure of the function. Therefore, to ensure the reliability of our analysis, in the main text, we set the masking baseline following this theory. This aims to eliminate the semantic information of the original token as much as possible in the context of interactions without introducing extraneous semantics, thereby yielding sparser extracted interactions.

Although alternative values such as the mean or zero theoretically fail to meet this requirement, to conduct sensitivity analysis, we replace the masking baseline with the mean embedding.

Matching-error bound ϵ. As shown in Equation (2) of the main text and Equation (15) in the Appendix, decreasing ϵ reduces the fitting error but also leads to less sparse interactions. We decrease ϵ from $0.04 \cdot \mathbf{E}_v|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$ to $0.02 \cdot \mathbf{E}_v|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$. Furthermore, since $\tau = 2 \cdot \epsilon$ is established in the main text, to strictly control variables, we maintain the absolute value of τ fixed at $0.08 \cdot \mathbf{E}_v|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$ when adjusting ϵ.

Salient threshold τ. Increasing τ filters out interactions with smaller numerical utility (e.g., $|I^{\text{AND}}_S|$ or $|I^{\text{OR}}_S|$), thereby reducing the total number of extracted interactions. In the smaller-threshold setting, we reduce τ from the default $2\epsilon = 0.08 \cdot \mathbf{E}_v|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$ to $\tau = \epsilon = 0.04 \cdot \mathbf{E}_v|v(\mathbf{x}) - v(\mathbf{x}_\emptyset)|$, while keeping ϵ at its default value.

Setting	PNCI for activations representing primary semantics	PNCI for activations representing other semantics
original	0.124 ± 0.181	0.282 ± 0.312
average baseline	0.229 ± 0.277	0.452 ± 0.341
smaller matching-error bound	0.155 ± 0.206	0.335 ± 0.326
smaller salient threshold	0.177 ± 0.207	0.374 ± 0.315

Table 5: Mean PNCI for SAE feature activations representing primary semantics and other semantics under four interaction computation settings.

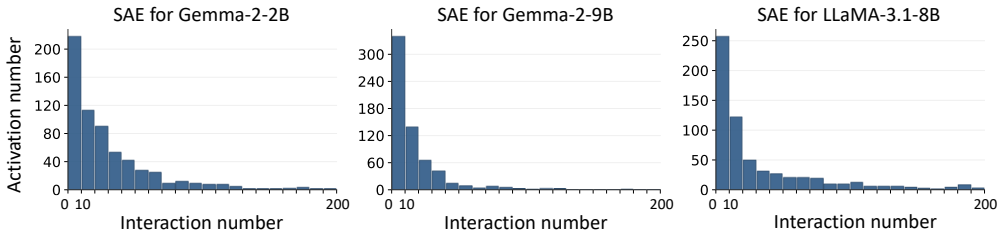


Figure 13: Distributions of interaction counts encoded by SAE feature activations across the three SAE models. For visual clarity, feature activations encoding more than 200 interactions are omitted; these activations account for less than 3% of all analyzed feature activations.

We next examine the effect of these choices on the four experiments in order. Unless otherwise stated, the sensitivity analysis uses the SAE applied to Gemma-2-9B. Because re-annotating all feature activations under every setting is costly, Experiment 4 reuses the primary semantics identified by the human evaluators in the main experiments.

Experiment 1 (interaction number). Table 4 and Figure 9 show that changing the interaction computation parameters changes the absolute interaction-count distribution but not its main pattern. The low interaction density group remains the largest group under all four settings, accounting for 57.0%–80.8% of the feature activations. The average baseline can introduce noisy patterns, inflating the proportion of activations in the high interaction density group from 5.9% to 19.3%, while decreasing the matching-error bound or the salient threshold retains more interactions. Thus all three changes harm interaction sparsity. The boxplots in Figure 9 reveal that the average number of interactions per feature varies significantly, and that there is no clear correlation between the number of interactions and the feature's activation frequency. These observations confirm that Findings 1 and 2 remain valid.

Experiment 2 (interaction complexity). Figure 10 shows that the complexity distribution is stable across all four settings. We observe that across all four settings, SAE feature activations in the low interaction density group (encoding ≤ 10 interactions) primarily model low-order interactions (orders 1–3). In contrast, SAE feature activations in the high interaction density group (encoding $>$ 20 interactions) not only model many first-order interactions but also encode numerous mid-order interactions (orders 4–6). This indicates that SAE feature activations encoding fewer interactions tend to model simpler interactions, confirming that Finding 3 remains valid.

Experiment 3 (positive-negative cancellation). Figure 11 shows that 55.2%–74.8% of feature activations encode interactions with minor cancellation effects. The smaller matching-error bound causes the largest variation in the proportion of activations with minor interaction cancellation, while the average baseline and the smaller salient threshold cause smaller shifts. Although these changes incorporate different amounts of non-salient noisy interactions into the evaluation, activations with minor interaction cancellation consistently constitute the majority. Furthermore, Figure 12 shows that regardless of how the interaction extraction algorithm's settings are adjusted, there remains no discernible relationship between PNCI and activation strength. In summary, Finding 4 remains valid.

Experiment 4 (semantic consistency). Due to cost constraints, we only recruited human evaluators to assess the semantic consistency of interactions under the original setting. For the other settings, we directly reuse the primary semantics analyzed from the original setting. It should be noted that

SAE Model	Low interaction density group (encoding $\leq$10 interactions)	Medium interaction density group (encoding 11–20 interactions)	High interaction density group (encoding >20 interactions)
SAE for Gemma-2-2B	77.14%	13.97%	8.89%
SAE for LLaMA-3.1-8B	73.81%	10.63%	15.56%
SAE for Gemma-2-9B	89.37%	7.78%	2.86%

Table 6: Distribution of SAE feature activations based on interaction counts across different SAE models under the revised grouping thresholds.

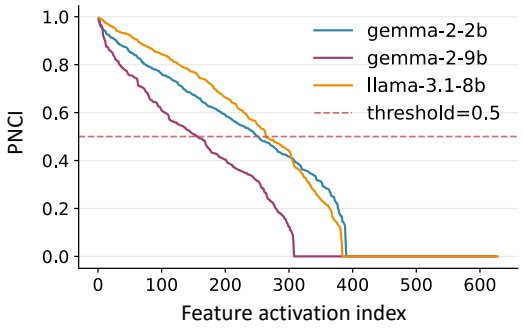


Figure 14: Ranked distributions of PNCI values for feature activations from the three SAE models. For each model, feature activations are sorted in descending order of PNCI. A feature activation is classified as exhibiting minor interaction cancellation when its PNCI does not exceed the selected threshold. The dashed line marks the default threshold of $\mathrm{PNCI} = 0.5$.

the average embedding baseline is theoretically unfaithful (Ren et al., 2023b), making it unreasonable to evaluate semantic consistency based on interactions calculated using this baseline. As for varying the matching-error bound and salient threshold, these modifications primarily affect small interactions, whereas our semantic consistency analysis focuses exclusively on top-ranked interactions. Table 5 compares the mean PNCI of feature activations representing primary semantics with those not representing primary semantics. As shown, the PNCI of feature activations representing primary semantics is consistently lower on average than that of those not representing primary semantics. This supports the robustness of Finding 5 under these settings.

K.2 Hyperparameters for Semantic Analysis

Criteria for interaction density grouping. In the main text, we classify feature activations into three groups based on the number of encoded interactions: low interaction density ($\leq$ 10 interactions), medium interaction density (11–20 interactions), and high interaction density ($>$ 20 interactions). This categorization is primarily based on the observed distribution of interaction counts shown in Figure 13, where most SAE feature activations encode only a few interactions, while a small fraction encodes a substantial number.

This grouping criterion primarily affects the conclusions of Experiments 1 and 2. To verify the robustness of our findings, we adjusted the partitioning thresholds to: low interaction density ($\leq$ 15 interactions), medium interaction density (16–30 interactions), and high interaction density ($>$ 30 interactions). After re-calculating the proportions of each group and plotting the interaction order distribution based on the new criteria, Table 6, the low interaction density group remains the largest for all three SAE models. The results in Figure 15 reveal that the positive correlation between the number of interactions and interaction complexity remains significant. Furthermore, the substantial positive-negative cancellation phenomenon still persists in high interaction density.

Thus, the core conclusions of Experiments 1 and 2 remain valid under the new grouping criteria. Moreover, the conclusions of Experiments 3 and 4 are unaffected by this parameter.

Criteria for classifying degrees of interaction cancellation. In the main text, we use $\mathrm{PNCI} \leq 0.5$ to classify feature activations as having minor or severe interaction cancellation. To evaluate the impact of this threshold, we computed the PNCI for all feature activations across the three SAE models discussed in the main text and plotted their distributions in descending order. Figure 14 shows that

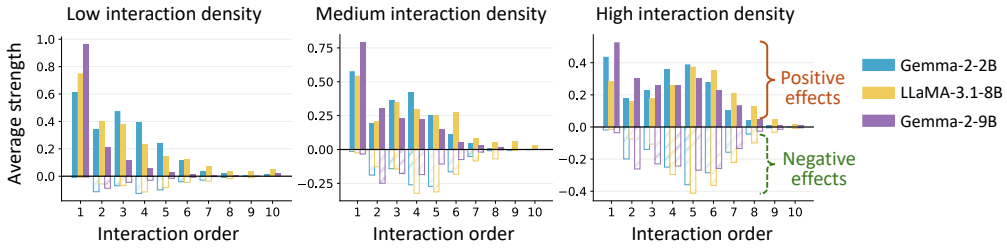


Figure 15: Interaction-complexity distributions under the revised interaction-density thresholds. The low, medium, and high interaction density groups contain feature activations encoding ≤ 15, 16–30, and > 30 interactions, respectively. For each SAE model and interaction-density group, the figure shows the positive and negative interaction strengths across interaction orders. To compare the distributions despite differences in interaction-strength scale across SAE models, each group-level mean complexity vector $\bar{\mathbf{d}}$ is normalized to unit ℓ_2-norm.

a large number of feature activations exhibit a PNCI of 0, which occurs because certain feature activations exclusively model positive interactions. Further observation indicates that regardless of how the PNCI threshold is configured, the SAE applied to Gemma-2-9B consistently exhibits the highest proportion of feature activations with minor interaction cancellation.

This suggests that although the threshold setting alters the specific proportional values, the core trend regarding the inter-model differences in Experiment 3 remains unchanged. Meanwhile, the conclusions of Experiments 1, 2, and 4 are also unaffected by this parameter.

Summary. In summary, although the specific metric values are influenced by hyperparameters, the trends identified in the main findings of the paper remain robust to these hyperparameter choices.

L Detailed Statistics of Each SAE Feature

For the three SAEs selected for our experiment, we report a series of statistics for each sampled SAE feature across all sampled activations. The rows in each table are sorted by activation frequency in descending order.

(1) **Feature Index (Feat. Idx.).** The encoder of an SAE typically has 16k–131k output dimensions, each of which is referred to as an SAE feature. "Feature Index" identifies the corresponding encoder output dimension.

(2) **Activation Frequency (Act. Freq.).** Given a dataset, an SAE feature may activate at different token positions. "Act. Freq." is the percentage of all token positions at which the feature activates.

(3) **Mean Interaction Number (Mean Num.).** For each sampled feature activation, we extract a set of salient interactions. "Mean Num." is the mean number of these interactions over all sampled activations of a feature.

(4) **Proportion of Semantically Sparse Activations (Sparse Prop.).** "Sparse Prop." is the percentage of sampled activations in the semantically sparse activation set (≤ 10 interactions). The proportion in the non-sparse activation set is its complement.

(5) **Average Interaction Complexity (Avg. Comp.).** Interaction complexity is measured by the intensity-weighted average interaction order: interaction orders receive weights proportional to their total absolute interaction strength, so a higher value indicates that more strength is carried by higher-order interactions. "Avg. Comp. Sparse" and "Avg. Comp. Dense" report the mean interaction complexity in the semantically sparse and non-sparse activation sets, respectively.

(6) **Mean Positive-Negative Cancellation Index (Mean PNCI).** PNCI measures the extent to which positive and negative interaction effects cancel each other; a higher PNCI indicates more severe cancellation. "Mean PNCI Sparse" and "Mean PNCI Dense" report the mean PNCI over the semantically sparse and non-sparse activation sets, respectively.

Feat. Idx.	Act. Freq.	Mean Num.	Sparse Prop.	Avg. Comp Sparse	Avg. Comp Dense	Mean PNCI Sparse	Mean PNCI Dense	SCATP	Primary semantics
9159	10.927%	11.17	53.33%	3.31	3.01	0.27	0.64	0.00%	–
45458	1.818%	9.30	70.00%	3.39	3.11	0.13	0.71	23.33%	geographic names or region-related terms
58358	0.688%	15.73	50.00%	3.01	4.48	0.19	0.73	20.00%	concepts related to the Black Death
54501	0.478%	12.90	73.33%	3.09	3.80	0.22	0.73	0.00%	–
18846	0.375%	4.97	93.33%	3.89	2.96	0.21	0.39	0.00%	–
41382	0.354%	15.77	43.33%	3.22	4.05	0.23	0.65	0.00%	–
63423	0.305%	13.27	66.67%	2.81	3.91	0.18	0.75	10.00%	terms related to race and ethnicity
44734	0.284%	10.13	63.33%	2.82	3.97	0.28	0.66	0.00%	–
45299	0.254%	10.40	76.67%	2.43	3.75	0.19	0.68	0.00%	–
5806	0.247%	10.10	70.00%	3.92	3.18	0.26	0.65	10.00%	–
3312	0.219%	9.83	70.00%	3.91	4.18	0.19	0.67	0.00%	–
28374	0.219%	3.33	93.33%	2.92	1.40	0.06	0.32	13.33%	terms related to building floors
147	0.196%	16.93	66.67%	4.25	4.34	0.23	0.77	0.00%	–
18178	0.182%	13.10	56.67%	3.72	3.99	0.23	0.65	0.00%	–
44904	0.177%	17.90	46.67%	2.71	3.53	0.19	0.74	30.00%	problems in algorithms
37667	0.161%	13.97	60.00%	3.27	3.40	0.21	0.70	0.00%	–
17271	0.158%	21.60	56.67%	2.62	3.63	0.16	0.71	10.00%	terms describing size
5567	0.149%	10.53	60.00%	2.59	4.04	0.28	0.59	0.00%	–
49877	0.138%	10.87	63.33%	3.83	4.61	0.13	0.76	13.33%	the Super Bowl
24738	0.124%	10.67	60.00%	3.72	3.82	0.22	0.62	0.00%	–
53388	0.112%	16.60	60.00%	3.18	4.65	0.06	0.74	0.00%	–

Table 7: Detailed statistics for sampled features of the SAE built upon Gemma-2-2B. Rows are sorted by decreasing activation frequency. For the meanings of the table header abbreviations, please see Appendix L.

(7) **SCATP and Primary Semantics.** Following Appendix J.3, the most frequently annotated semantic concept for a feature is defined as its primary semantics. SCATP is the percentage of sampled activations which represent the primary semantics, and "Primary Semantics" reports the concept when one was identified. A dash indicates that no primary semantics was identified or that the relevant statistic is unavailable.

A special case. Function words were not included as interaction players in our analysis. Nevertheless, feature #3077 in the SAE built upon Gemma-2-9B consistently activates on function words such as "and," "or," and "for." For most sampled activations, the output with all interaction players masked, $v(\mathbf{x}_\emptyset)$, is close to the output for the unmasked input, $v(\mathbf{x})$. This suggests that the feature stably activates on function words and is only minimally influenced by preceding content words. Because its function-word semantics cannot be evaluated under the same annotation protocol, we do not report a primary-semantics proportion for this feature.

Feature #15783 in the same SAE also activates on function words (e.g., "of," "to," and "and"). Unlike feature #3077, however, its activation on a function word is strongly influenced by academic terms in the preceding context, such as "science," "analysis," or "seminal paper."

M Interaction-based Explanation for More SAE Features

In Figure 16 and Figure 17, we further provide several examples of how the activation values of SAE features can be decomposed into a set of concise interactions. Due to space constraints, we only present the top-ranked interactions. Note that the SAE features activate at the last token of the input prompt. Interactions are computed among content words marked with an underline (see Appendix I.2 for the definition of content words).

N More Examples of Semantically Consistent Feature Activation Groups

As described in Experiment 4 of the main paper, human evaluators were asked to provide several semantic concepts for a given SAE feature. This allowed us to identify groups of feature activations sharing similar semantic concepts. Figure 18 presents three groups of feature activations sharing similar semantic concepts. Each group of activations belongs to the same SAE feature. Please note

Feat. Idx.	Act. Freq.	Mean Num.	Sparse Prop.	Avg. Comp Sparse	Avg. Comp Dense	Mean PNCI Sparse	Mean PNCI Dense	SCATP	Primary semantics
6299	9.095%	3.03	100.00%	1.50	–	0.11	–	0.00%	–
8232	3.851%	7.80	76.67%	1.87	2.57	0.24	0.44	23.33%	terms describing city size or administrative divisions
3553	1.324%	8.27	76.67%	1.97	3.01	0.19	0.74	0.00%	–
2961	1.105%	16.83	70.00%	2.24	4.13	0.18	0.71	0.00%	–
4699	0.939%	5.07	90.00%	1.87	2.14	0.14	0.53	10.00%	terms related to refugees and asylum
12538	0.846%	1.73	100.00%	1.06	–	0.03	–	90.00%	words beginning with the letter "e"
12070	0.739%	10.50	70.00%	1.90	3.21	0.17	0.73	16.67%	words related to competition
10378	0.622%	4.53	86.67%	1.98	1.85	0.20	0.39	26.67%	terms expressing merging or combining
8798	0.578%	6.70	76.67%	2.27	2.56	0.19	0.57	10.00%	broadcast a program or game
1507	0.503%	6.83	70.00%	2.25	2.46	0.21	0.59	30.00%	expressions introducing examples
3077	0.438%	5.13	86.67%	1.31	2.20	0.25	0.65	–[a]	function words[a]
10595	0.375%	10.83	53.33%	2.66	2.95	0.19	0.65	0.00%	–
13037	0.319%	6.00	86.67%	3.20	2.66	0.17	0.62	0.00%	–
15783	0.301%	3.37	93.33%	1.25	2.06	0.09	0.59	26.67%	phrases related to academia
11777	0.245%	7.53	76.67%	1.99	2.63	0.28	0.66	0.00%	–
666	0.233%	9.50	66.67%	3.20	2.86	0.25	0.63	16.67%	education-related concepts
4439	0.186%	1.07	100.00%	1.02	–	0.01	–	86.67%	concepts related to history
12314	0.172%	13.50	56.67%	4.48	3.26	0.10	0.71	0.00%	–
283	0.149%	4.80	93.33%	1.50	3.20	0.19	0.73	30.00%	expressions indicating discovery
9208	0.128%	5.07	86.67%	2.34	3.11	0.08	0.71	13.33%	–
318	0.105%	6.90	80.00%	2.46	3.41	0.12	0.71	30.00%	terms related to the census

[a] See the special case discussion in Appendix L.

Table 8: Detailed statistics for sampled features of the SAE built upon Gemma-2-9B. Rows are sorted by decreasing activation frequency. For the meanings of the table header abbreviations, please see Appendix L.

Feat. Idx.	Act. Freq.	Mean Num.	Sparse Prop.	Avg. Comp Sparse	Avg. Comp Dense	Mean PNCI Sparse	Mean PNCI Dense	SCATP	Primary semantics
17670	9.917%	19.10	66.67%	3.56	4.83	0.16	0.86	0.00%	–
18710	1.973%	6.13	86.67%	2.42	2.81	0.18	0.47	0.00%	–
5717	0.673%	6.80	83.33%	1.93	3.21	0.27	0.67	40.00%	people's names
2515	0.579%	3.00	100.00%	1.46	–	0.10	–	63.33%	tokens preceded by a word beginning with the letter "m"
11728	0.459%	11.20	63.33%	3.40	4.71	0.28	0.73	0.00%	–
15825	0.349%	6.13	93.33%	4.35	5.48	0.17	0.60	0.00%	–
15177	0.325%	16.47	60.00%	3.65	4.59	0.24	0.70	0.00%	–
22974	0.276%	28.30	30.00%	3.50	4.48	0.22	0.80	0.00%	–
8678	0.247%	6.97	80.00%	3.62	4.19	0.19	0.69	20.00%	words expressing inclusion
12673	0.217%	29.03	33.33%	3.17	5.32	0.20	0.84	0.00%	–
3118	0.207%	13.17	73.33%	3.43	4.42	0.28	0.81	13.33%	terms related to the engine safety devices
14471	0.186%	18.73	70.00%	1.95	4.54	0.15	0.80	23.33%	phrases expressing multiple elements organized together to form a whole
24305	0.172%	41.17	23.33%	3.21	5.10	0.05	0.81	0.00%	–
8035	0.162%	2.90	100.00%	1.90	–	0.06	–	26.67%	words related to early or basic stages
10685	0.151%	10.20	66.67%	2.79	4.07	0.21	0.64	26.67%	terms related to power and authority
8403	0.137%	21.27	53.33%	3.44	4.50	0.21	0.77	0.00%	–
25197	0.130%	25.60	40.00%	2.17	4.71	0.15	0.86	33.33%	phrases containing numbers
2372	0.120%	19.87	63.33%	3.02	4.60	0.28	0.82	26.67%	words related to salary and income
10782	0.115%	17.33	63.33%	3.83	5.19	0.23	0.86	0.00%	–
5961	0.111%	20.60	46.67%	3.20	4.64	0.29	0.76	0.00%	–
10363	0.101%	5.43	86.67%	5.69	4.64	0.09	0.72	0.00%	–

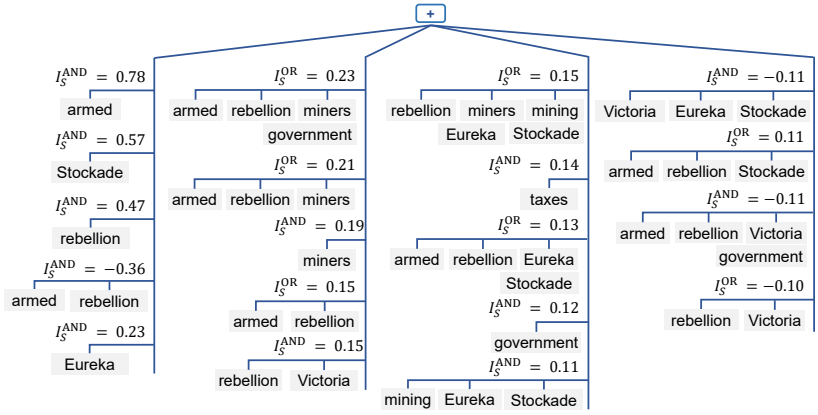
Table 9: Detailed statistics for sampled features of the SAE built upon LLaMA-3.1-8B. Rows are sorted by decreasing activation frequency. For the meanings of the table header abbreviations, please see Appendix L.

that only three feature activations are displayed per group; in practice, the number of activations modeling similar semantics may be larger.

SAE built upon MLP output of Gemma-2-2B, feature #58358

prompt In 1854 at Ballarat there was an armed rebellion against the government of Victoria by miners protesting against mining taxes (the "Eureka Stockade"). This was

AND-OR logical function $g(\mathbf{x})$



SAE built upon residual stream of Gemma-2-9B, feature #8232

prompt Downtown San Diego[a] is the central business[a] district of San Diego[b], though the city is filled with business[b] districts

AND-OR logical function $g(\mathbf{x})$

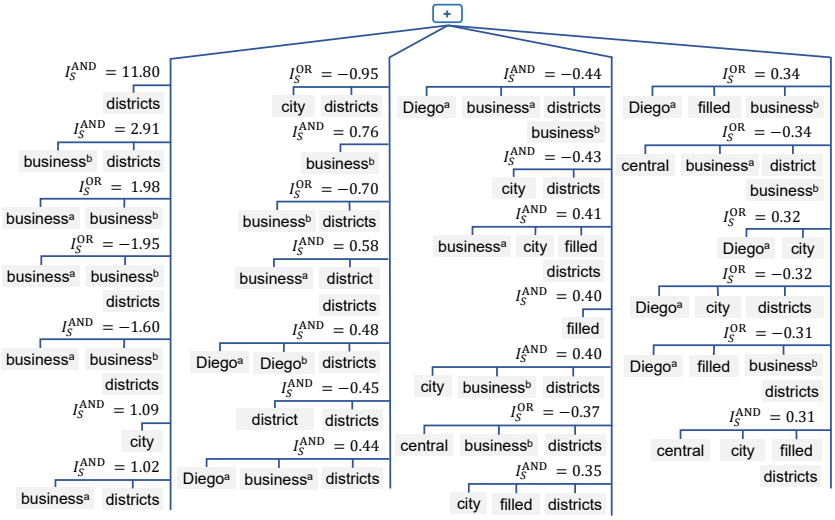


Figure 16: Interaction-based explanations for two SAE feature activations. These SAE features activate at the position of the last token of the prompt. Interactions are computed among content words marked with an underline.

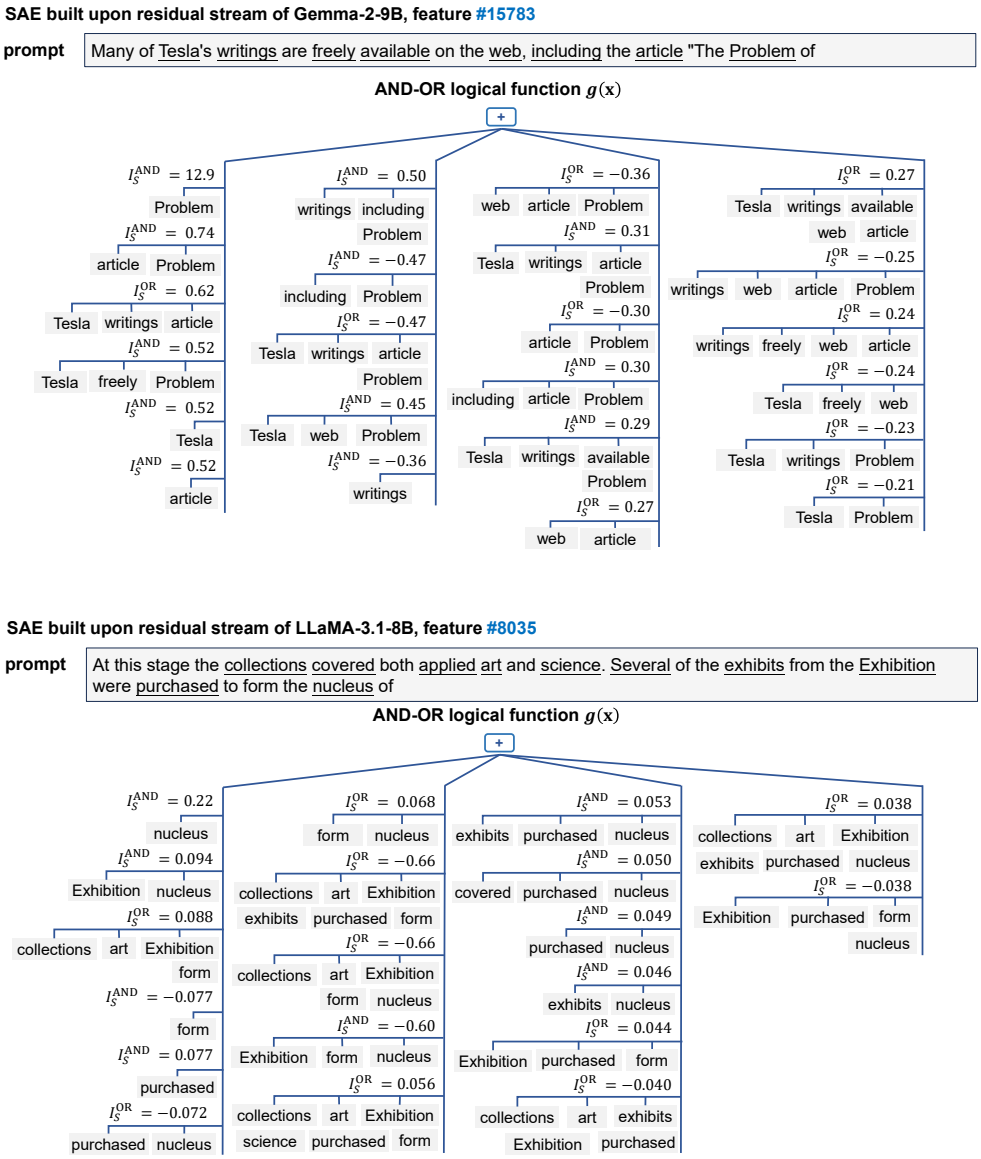


Figure 17: Interaction-based explanations for another two SAE feature activations. These SAE features activate at the position of the last token of the prompt. Interactions are computed among content words marked with an underline.

Feature #10378, SAE built upon the residual stream of Gemma-2-9B

1: Formed (A) in November (B) 1990 (C) by the equal (D) merger (E) of

I^{AND}(E)= 23.78, I^{OR}(A,B,C,E)= −2.57, I^{AND}(D,E)= −2.52, I^{OR}(A,B,C)= 2.29, I^{OR}(A,C)= 2.18, I^{OR}(A,C,E)= −2.11, I^{AND}(C,E)= 2.00, I^{AND}(C,D,E)= 1.95, I^{OR}(A,C,D)= −1,93, I^{OR}(A,C,D,E)= 1.35, …

2: These final (A) missions (B) would combine (C) the I (D) and J (E) types (F) in

I^{AND}(C)= 14.10, I^{AND}(C,F)= 2.93, I^{AND}(B,C,E)= 2.77, I^{AND}(A,B,C,F)= 1.98, I^{AND}(B,C)=−1.98 I^{OR}(E,F)= 1.90, I^{AND}(C,E)= 1.87, I^{OR}(C,E,F)= −1.76, I^{AND}(F)= 1.64, I^{AND}(A,B,C)=−1.49, …

3: They also plot (A) and combine (B) measurements (C) of geological (D) structures (E) in

I^{AND}(A,B,C,D,E)= 6.98, I^{AND}(A,B,C,D)= −4.38, I^{AND}(B)= 2.90, I^{AND}(A,B,D)= 2.66, I^{AND}(B,C,E)= −2.56, I^{AND}(B,C,D,E)= 2.35, I^{AND}(B,C)= 1.91, I^{AND}(C)= −1.77, I^{OR}(B,E)= 1.77, I^{AND}(A)= 1.49, …

Shared semantics: terms expressing merging/combining

Feature #2372, SAE built upon the residual stream of LLaMA-3.1-8B

1: Salaries (A) for primary (B) teachers (C) in Ireland (D) depend (E) mainly (F)

I^{AND}(A,B,C,D,E,F)= −1.32, I^{AND}(A,D,E,F)= 1.21, I^{AND}(A,E,F)= 1.06, I^{AND}(F)= 0.90, I^{AND}(A)= 0.67, I^{AND}(A,D,F)= −0.64, I^{OR}(A,F)= −0.56, I^{AND}(A,B,D,E,F)= 0.54, I^{AND}(E)= 0.45, I^{OR}(E,F)= −0.35, …

2: Salary (A) ranges (B) from $40,000 (C) /year (D) to $90,000 (E) /yr (F).

I^{AND}(B,C,D,E,F)= 0.43, I^{AND}(A,B,E,F)= −0.35, I^{AND}(A)= 0.30, I^{OR}(A,E)= 0.24, I^{AND}(A,B,E)=0.23 I^{AND}(A,C,E)= 0.22, I^{AND}(A,C,E,F)= −0.21, I^{AND}(A,E)= 0.18, I^{AND}(A,B,C,D,E)= 0.17, I^{AND}(A,B,C,D,E,F)=0.16, …

3: US (A) teachers (B) are generally paid (C) on (D) graduated (E)

I^{AND}(B,C,D,E)= 0.42, I^{AND}(D,E)= 0.089, I^{AND}(E)= 0.085, I^{AND}(D)= 0.054, I^{AND}(B,E)= −0.051, I^{AND}(B,C,D)= 0.045

Shared semantics: words related to salary and income

Feature #28374, SAE built upon the MLP output of Gemma-2-2B

1: These buildings (A) are predominantly (B) four stories (C) high (D)

I^{AND}(B,C,D)= 2.21, I^{AND}(D)= 1.08, I^{OR}(A,C)= 1.03, I^{OR}(A,C,D)= −0.87, I^{AND}(A,C,D)= −0.86, I^{AND}(A,D)= 0.82, I^{AND}(A,B,C,D)= 0.39, I^{OR}(A,B,C)= 0.12,

2: While the Maria Skłodowska-Curie Institute of Oncology (A) it is one of the largest (B) and most modern (C) oncological (D) institutions (E) in Europe (F). The clinical (G) section (H) is located (I) in a 10-floor (J)

I^{AND}(J)= 7.94, I^{AND}(G)= −0.51, I^{AND}(G,J)= 0.5, I^{AND}(G,J)= −0.35, I^{AND}(F)=0.32, I^{OR}(G,H,I)= 1.90, I^{AND}(F,J)= 1.87, I^{OR}(G,H,I,J)= −1.76, I^{AND}(C)= 1.64, I^{AND}(I,J)=−1.49, …

3: Henry Young (A) Darracott (B) Scott (C), also of the Royal (D) Engineers (E). He designed (F) to the north (G) west (H) of the garden (I) the five-storey (J)

I^{AND}(J)= 6.67, I^{AND}(F)= 0.85, I^{OR}(F,J)= −0.78, I^{AND}(F,J)= 0.65, I^{AND}(I)= 0.47, I^{AND}(F,G,H,I,J)= 2.35, I^{AND}(D,E,J)= 1.91, I^{OR}(I,J)= −1.77, I^{AND}(D)= 1.77, I^{AND}(G)= 1.49, …

Shared semantics: building floors

Figure 18: More examples of feature activations sharing similar semantics.