InverseScope: Scalable Activation Inversion for Interpreting Large Language Models

Anonymous ACL submission

Abstract

Sparse autoencoders (SAEs) have proven effective for extracting monosemantic features from large language models (LLMs), yet these features are typically identified in isolation. However, broad evidence suggests that LLMs capture the intrinsic structure of natural language, where the phenomenon of "feature splitting" in particular indicates that such structure is hierarchical. To capture this, we propose the **Hierarchical Sparse Autoencoder (HSAE)**, which jointly learns a series of SAEs and the parent-child relationships between their features. HSAE strengthens the alignment between parent and child features through two novel mechanisms: a structural constraint loss and a random feature perturbation mechanism. Extensive experiments across various LLMs and layers demonstrate that HSAE consistently recovers semantically meaningful hierarchies, supported by both qualitative case studies and rigorous quantitative metrics. At the same time, HSAE preserves the reconstruction fidelity and interpretability of standard SAEs across different dictionary sizes. Our work provides a powerful, scalable tool for discovering and analyzing the multi-scale conceptual structures embedded in LLM representations.

1 Introduction

Understanding the mechanisms by which large language models (LLMs) represent and process information is of great interest in the machine learning community. Recent advances in mechanistic interpretability have shifted focus from analyzing individual neurons (Olah et al., 2020; Bills et al., 2023), which are often polysemantic, toward identifying more coherent units of analysis. Sparse autoencoders (SAEs) have emerged as a particularly successful approach in this direction (Bricken et al., 2023; Cunningham et al., 2023). By learning an overcomplete dictionary that maps dense activations to a sparse latent space, SAEs scalably extract features that are human-interpretable.

Beyond identifying isolated features, emerging evidence shows that SAE representations are embedded within a sophisticated internal organization: recent work reveals spatial clustering by semantic groups (Li et al., 2025) and hierarchical co-activation between abstract and specialized features (Clarke et al., 2024). Since navigating flat, unstructured dictionaries is inherently difficult, extracting these latent structures is essential for organized, multi-scale interpretation. *Feature splitting*—where broad concepts break into finer subfeatures as dictionary size increases (Bricken et al., 2023)—hints directly at the hierarchy. While prior works have often viewed this as an obstacle to identifying universal, canonical features (Leask et al., 2025; Bussmann et al., 2025), we instead recognize it as evidence of the underlying hierarchy represented in the model. By exploiting feature splitting rather than suppressing it, we transform this behavior into a hierarchical indexing structure for the feature space, organizing the "atoms" of representation into a structured forest.

For this purpose, we introduce the **Hierarchical Sparse Autoencoder (HSAE)**. The HSAE simultaneously trains a series of SAE levels with increasing dictionary sizes, with features at each level explicitly assigned to a parent in the preceding coarser level, forming a tree-structured hierarchy. Model parameters and the parent–child assignments co-evolve through alternating optimization. To ensure that hierarchical links reflect semantic subsumption rather than mere structural adjacency, we design optimization objective to enforce functional dependence across levels via: (1) a **structural constraint loss** encouraging parent features to be reconstructed from their assigned children's outputs, and (2) a **random feature perturbation** mechanism that treats parent and children contributions as interchangeable during reconstruction of original activations. These mechanisms align the learned features into a consistent conceptual taxon-

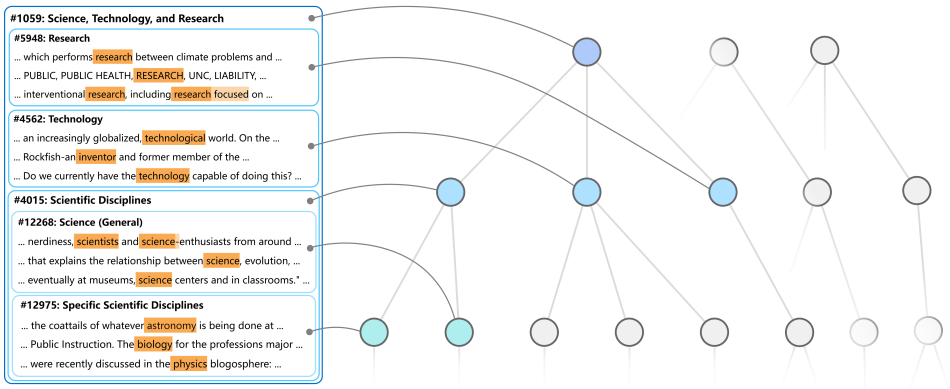


Figure 1: **Hierarchical feature discovery with HSAE.** We visualize a tree within the learned feature forest alongside a semantic dashboard of its nodes. Each node includes a unique feature index (e.g., #1059), a human-annotated semantic label, and representative top-activating context snippets. Orange highlights indicate the activation positions, with color intensity reflecting the activation magnitude. HSAE captures a clear conceptual taxonomy: a broad parent feature representing *Science, Technology, and Research* (#1059) is decomposed into specialized features such as *Scientific Disciplines* (#4015). This intermediate node further branches into a feature capturing lexical patterns containing the word *science* or *scientists* (#12268) alongside a feature representing specific discipline names such as *biology* or *physics* (#12975). For clarity, only a three-layer subgraph and examples of some features are shown.

omy, where higher-level features are incentivized to act as distilled abstractions of their finer-grained descendants.

In our experiments, we apply HSAE to the residual stream activations of different LLMs and find that it captures a rich, multi-layered hierarchical structure among interpretable features. As illustrated in Figure 1, HSAE naturally organizes discovered features into a conceptual forest, revealing a progressive refinement of semantic meaning. In this structure, high-level parent features initially respond to broad, domain-general patterns, which are then systematically decomposed into increasingly specific features at deeper levels.

To provide a comprehensive evaluation of our results, we present quantitative evidence that the discovered structures are statistically robust and semantically meaningful, significantly outperforming baselines that infer relations post-hoc from independently trained SAEs. Complementary qualitative case studies further illustrate consistent semantic hierarchies across domains. We further verify that HSAE matches or exceeds previous SAE methods on feature quality benchmarks, and ablate the necessity of each proposed mechanism.

Alongside rigorous evaluations, we explore the practical utility of the learned hierarchical structure through two downstream applications of HSAE. First, we show that the hierarchical organization enables more efficient and higher-quality feature interpretation by reusing high-level feature descriptions. Second, we demonstrate that HSAE provides more effective control over coarse-grained concepts during activation steering, addressing a limitation of standard SAEs in representing broad semantic categories. Together, these results suggest that explicitly modeling hierarchical structure not only improves interpretability, but also enhances controllability of learned representations.

The main contributions of this work are:

- **A novel hierarchical architecture:** We propose HSAE, a SAE framework that integrate explicit structural priors into training via structural constraint loss and random feature perturbation, enabling the learning of an organized "conceptual forest."
- **Benchmarking hierarchical consistency:** We establish a set of benchmarks to measure hierarchy consistency, introducing statistical metrics alongside an LLM-based automated interpretability framework. Our results demonstrate that HSAE recovers more coherent structures than post-hoc analysis of independent SAEs.
- **Improved interpretability and controllable steering:** We leverage the learned hierarchy to enable more efficient feature interpretation by reusing coarse-grained semantic structure across levels. We further demonstrate that HSAE improves coarse-grained concept steering, highlighting the utility of hierarchical representations for controllable generation.

2 Related work

2.1 Discovering structures in activations

Early word embedding analyses revealed semantic analogies via vector arithmetic (e.g., $v(\texttt{"king"}) - v(\texttt{"man"}) + v(\texttt{"woman"}) \approx v(\texttt{"queen"})$ (Mikolov et al., 2013a,b)) and similar regularities appear in LLM activations (Gurnee and Tegmark, 2024; Heinzerling and Inui, 2024). These observations support the linear representation hypothesis, where semantic concepts (features) in the activation space of LLMs corresponds to one-dimensional directions (Park et al., 2023), motivating the development of SAEs that decompose activations into an overcomplete basis of interpretable linear features (Bricken et al., 2023; Cunningham et al., 2023).

SAEs have uncovered various spatial structures within the feature space: *feature splitting* of broad concepts into finer sub-features (Bricken et al., 2023), multi-scale clusters corresponding to functional domains (e.g., mathematics, code) (Li et al., 2025), and two-dimensional circular manifolds for periodic concepts like weekdays (Engels et al., 2025). Recent work also reveals hierarchical "hub-and-spoke" topologies in co-occurrence graphs, where abstract features act as hubs for specialized ones (Clarke et al., 2024).

2.2 Feature splitting

Feature splitting (Bricken et al., 2023) is an observed phenomenon in SAEs where an interpretable, broad feature decomposes into multiple specialized ones as the dictionary size increases. For example, a general "punctuation mark" feature may split into distinct latents for periods, commas, and question marks. Related phenomena that can affect feature quality include *feature composition*, where frequently co-occurring independent features merge into a single latent (Anders et al., 2024; Wattenberg and Viégas, 2024), and *feature absorption*, where a general feature develops blind spots for subcases captured by more specialized latents (Chanin et al., 2024). These effects can be explained by the sparsity-driven training objectives of SAEs (Chanin et al., 2024). Metrics have been proposed to quantitatively measure these effects within an SAE (Huang et al., 2024; Chanin et al., 2024; Karvonen et al., 2025).

Several recent works treat feature splitting as a drawback and develop methods to mitigate this phenomenon (Bussmann et al., 2025; Chanin et al., 2025; Korznikov et al., 2025). In contrast, we interpret splitting as evidence of an underlying hierarchical conceptual structure and aim to recover this hierarchy directly. We show that leveraging this hierarchical structure, rather than suppressing it, improves both interpretability and downstream utility.

2.3 Architectural evolution of SAEs

The engineering of SAEs has evolved from early formulations based on L_1 regularization and ReLU activations (Bricken et al., 2023; Cunningham et al., 2023), which suffered from various issues such as feature shrinkage. Numerous subsequent works have improved sparsity enforcement, training dynamics, and scalability through alternative learning objectives, activation functions, and architectural designs (Rajamanoharan et al., 2024a; Gao et al., 2025; Bussmann et al., 2024; Rajamanoharan et al., 2024b; Karvonen et al., 2024; Mudide et al., 2025).

A notable direction in this evolution is the incorporation of structure into SAE models (Bussmann et al., 2025; Costa et al., 2025; Engels, 2025; Zaigrajew et al., 2025; Muchane et al., 2025). For hierarchy specifically, Matryoshka SAE (Bussmann et al., 2025) introduces nested dictionaries of increasing size to mitigate feature splitting, implicitly encouraging a multi-scale hierarchy among learned features. H-SAE (Muchane et al., 2025) further incorporates a two-level hierarchy into the SAE architecture, but relies on a predefined fixed hierarchical prior. In contrast, our method learns hierarchical structure directly from the activation distribution by jointly optimizing both features and their parent-child relationships, enabling the explicit and adaptive recovery of feature hierarchies from the activation space.

3 Hierarchical sparse autoencoders

In HSAE, we jointly train a collection of SAEs with increasing dictionary sizes and explicitly enforce a hierarchical structure among their features.

Technically, a standard SAE seeks to decompose the activation vector $x \in \mathbb{R}^d$ into a sparse linear combination of interpretable directions. Let $\{\mathrm{SAE}_\ell(\cdot)\}_{\ell=1}^{L}$ denote a sequence of L SAEs, where

$$\mathrm{SAE}_\ell(x) = \sum_{i=1}^{n_\ell} d_{\ell,i}\, \sigma(e_{\ell,i}^T x).$$

We use $f_{\ell,i}(x) = d_{\ell,i}\, \sigma(e_{\ell,i}^T x)$ to denote a single feature function. Here, ℓ indexes the hierarchy level and n_ℓ represents the dictionary size at level ℓ. Each SAE_ℓ is a standard two-layer JumpRELU

SAE (Rajamanoharan et al., 2024b). For simplicity, we omit the threshold parameter in σ, which is different for each feature (ℓ, i). Note that the hierarchy index ℓ does *not* correspond to depth in a multilayer neural network.

HSAE induces a partial tree structure over features by defining parent–child relationships between selected features in adjacent levels. For each feature (ℓ, i) where $1 \leqslant \ell \leqslant L-1$, we define a set of children indices

$$\mathcal{C}_{(\ell,i)} \subseteq \{1, \ldots, n_{\ell+1}\}.$$

We refer to (ℓ, i) as a parent feature and $\{(\ell+1, j) \mid j \in \mathcal{C}_{(\ell,i)}\}$ as its children. To ensure a tree-structured hierarchy, we require that child sets are disjoint across parent features

$$\mathcal{C}_{(\ell,i)} \cap \mathcal{C}_{(\ell,k)} = \varnothing, \quad \forall\, i \neq k.$$

The hierarchical relations introduced above do not by themselves impose constraints among related features. Instead, they define a structural prior that is instantiated through additional loss terms and auxiliary training mechanisms described later in this section.

3.1 Loss function

Each level's SAE employs a standard loss, which consists of a MSE reconstruction loss and an L_0 sparsity penalty

$$\mathcal{L}_{\text{SAE},\ell}(x) = \| \operatorname{SAE}_\ell(x) - x\|_2^2 + \lambda_\ell \sum_{i=1}^{n_\ell} 1_{\{\sigma(e_{\ell,i}^T x)>0\}}.$$

For each parent-children group, we introduce a parent-children constraint loss as

$$\mathcal{L}_{\text{PC},(\ell,i)}(x) = \begin{cases} \|f_{\ell,i}(x) - \sum_{j\in\mathcal{C}_{\ell,i}} f_{\ell+1,j}(x)\|_2^2, & \mathcal{C}_{\ell,i} \neq \varnothing, \\ 0, & \mathcal{C}_{\ell,i} = \varnothing. \end{cases}$$

The final HSAE loss function is

$$\mathcal{L}_{\text{HSAE}} = \mathbb{E}_{x\sim D}\left[\sum_{\ell=1}^{L} \mathcal{L}_{\text{SAE},\ell}(x) + \rho \left(\sum_{\ell=1}^{L-1}\sum_{i=1}^{n_\ell} \mathcal{L}_{\text{PC},(\ell,i)}(x)\right)\right].$$

where $x \sim D$ is the distribution of activations extracted from the specific site of the target LLM.

3.2 Parent-children feature perturbations

Beyond additional loss terms, we also introduce a structure-aware random perturbation of features to encourage the relationship between parent feature and its children.

When computing $\mathcal{L}_{\text{SAE},\ell}$, each level-$\ell$ feature is randomly replaced by the aggregate contribution of its children at level $\ell+1$.

Formally, given the perturbation rate r, let $z_{\ell,i} \sim \text{Bernoulli}(r)$ for $\ell \leqslant L-1$. We define

$$\tilde{f}_{\ell,i}(x) = (1 - z_{\ell,i}) f_{\ell,i}(x) + z_{\ell,i} \sum_{j\in\mathcal{C}_{\ell,i}} f_{\ell+1,j}(x).$$

and compute the SAE loss $\mathcal{L}_{\text{SAE},\ell}$ with the perturbed version of the original SAE:

$$\widetilde{\operatorname{SAE}}_\ell(x) = \sum_{i=1}^{n_\ell} \tilde{f}_{\ell,i}(x)$$

The random variables $z_{\ell,i}$ are sampled independently across features at each training step. This perturbation allows child features to account for interactions with other features in the parent layer, which are not captured by the parent-child constraint alone.

In practice, we adopt a parent-child constraint weight of $\rho = 0.01$ and a perturbation rate of $r = 5\%$. These values are selected based on ablation results reported in Appendix B (Figure 5) to balance hierarchical alignment with reconstruction fidelity. The additional computational overhead introduced by these mechanisms is modest. In particular, the parent-child constraint reuses features already activated for the reconstruction objective and adds only lightweight element-wise operations without additional matrix multiplications. We provide a more detailed discussion of computational cost in Appendix A.

3.3 Optimization strategy

Training HSAE involves the joint optimization of model parameters $\{e_{\ell,i}, d_{\ell,i}, \sigma_{\ell,i}\}$ and the set of hierarchical relations $\{\mathcal{C}_{\ell,i}\}$ simultaneously. To handle this joint objective, we adopt an alternating optimization strategy. Specifically, the training process alternates between the following two stages:

- **Parameter Optimization**: With the hierarchical structure fixed, we optimize the model parameters via gradient descent for a number of steps.
- **Hierarchy Update**: With the model parameters fixed, we update the hierarchical structure $\{\mathcal{C}_{\ell,i}\}$ to reflect the current feature organization.

This iterative process is maintained throughout training, ensuring that the hierarchical structure co-evolves with the learned SAE features. In practice, we perform a round of hierarchy update every 5000 gradient steps.

Parameter optimization. Most parameter optimization settings in HSAE are identical to those in standard JumpReLU SAEs. We use a straight-through estimator for gradients through the discontinuity induced by the L_0 sparsity penalty. Decoder vectors are normalized after each gradient step to fix their norms. We adopt ghost gradients to mitigate the dead neuron problem (Jermyn and Templeton, 2024).

We also introduce an adaptive sparsity weight to dynamically adjust the sparsity weights λ_ℓ during training, guiding the average L_0 of each level toward a shared target sparsity. The detailed method is provided in Appendix A.

Similarity-based hierarchy update. The hierarchy update step is guided by a general principle of feature similarity. Each feature at level $\ell + 1$ will select a new parent feature at level ℓ that has the most similarity according to a similarity metric.

We consider several instantiations of this principle. Similarity can be defined using feature co-activation statistics, measured as the proportion of training examples on which two features are jointly active, with statistics tracked via an exponential moving average. Alternatively, similarity can be computed in parameter space using the cosine similarity between encoder or decoder feature vectors. Empirically, HSAE is largely insensitive to the specific choice of similarity metric, with all variants exhibiting comparable behavior in the ablations reported in Appendix B (Figure 4). Unless otherwise specified, we use encoder-vector similarity throughout this paper.

Partial tree structure. Recognizing that not all features naturally conform to a hierarchical taxonomy, we relax the rigid tree constraint by allowing a subset of child features to remain unassigned. Specifically, we implement a quantile-based filtering mechanism where features whose maximum similarity scores fall within the given bottom percentile of all candidates are excluded from the hierarchy. We treat this exclusion level as a tunable hyperparameter, which is set to 20% in our experiments. This partial tree structure provides the necessary flexibility to accommodate independent features—those capturing idiosyncratic or atomic concepts—without forcing them into incoherent parent-child relationships, thereby preserving the structural faithfulness of the discovered hierarchy.

Appendix B shows that the parent-child constraint provides the primary alignment signal, while random perturbation further refines cross-level consistency. We also include ablations on dynamic partial-tree updates and hyperparameter sensitivity. We refer readers to the appendix for the full results.

4 Experiments

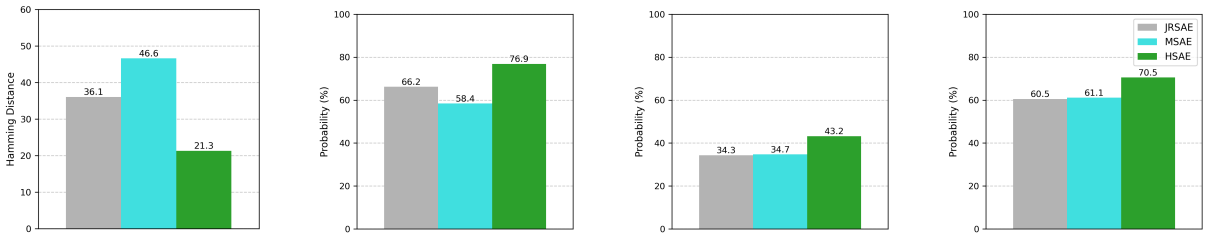
In this section, we present a comprehensive evaluation of HSAE, with rigorous quantitative analyses demonstrating its ability to capture a significantly richer internal organization than baseline SAEs. We first evaluate the learned structures using statistical metrics and automated interpretability assessments, showing that they are recovered consistently across the model. Complementary qualitative case studies (in Appendix D) provide concrete examples of the interpretable hierarchies learned by HSAE. To ensure that this hierarchical organization does not compromise core SAE performance, we further benchmark HSAE against established SAE evaluation protocols, measuring reconstruction fidelity and feature quality relative to baselines.

Baselines. We compare HSAE against hierarchies constructed post-hoc from independently trained SAEs. Specifically, we train JumpReLU SAEs (JRSAEs) (Rajamanoharan et al., 2024b) and Matryoshka SAEs (MSAEs) (Bussmann et al., 2025), two well-established SAE architectures, with dictionary sizes matched to the corresponding levels of HSAE, and then apply the same similarity-based assignment strategy to organize their features into hierarchies. These baselines represent structure obtained purely via post-hoc interpretation.

Experimental setup. Unless otherwise specified, all models discussed in this section are trained on 100M residual stream activations extracted from the 13th layer of `gemma2-2b` (Riviere et al., 2024) as it processes the mini-PILE dataset (Kaddour, 2023). For both HSAEs and baselines, we train four levels of SAEs with dictionary sizes of 2048, 4096, 8192, and 16384. We target a sparsity of $L_0 = 50$ for each level. Detailed experimental settings are provided in Appendix A. To ensure the robustness of our findings, we also conduct extensive experiments across different layers and target sparsity. These additional results are included in Appendix F. We observe consistent results across all tested settings.

4.1 Statistical metrics

To quantitatively evaluate the structural faithfulness of the discovered hierarchy, we employ three



(a) Parent-Child Align ↓ (b) $P(\mathrm{P} \mid \mathrm{C})$ ↑ (c) $P(\mathrm{C} \mid \mathrm{P})$ ↑ (d) AutoInterp ↑

Figure 2: **Quantitative evaluation of hierarchical structure. (a) Parent-child alignment**: Measured by the Hamming Distance between ground-truth parent activations and their logical-OR reconstruction from children, indicating how well parent features summarize their descendants' activations. **(b) Parent-given-child activation probability**: measuring the necessity of the parent feature for its children. **(c) Child-given-parent activation probability**: reflecting the coverage of children within the parent's activation space. **(d) AutoInterp**: LLM-based automated interpretability scores evaluating the semantic alignment between parent and child features.

metrics that measure the alignment between parent and child features. To avoid overfitting, all metrics are evaluated on held-out test splits.

Hierarchical consistency via logical-OR. We first assess whether parent features serve as effective semantic summaries of their children. We construct a logical-OR prediction where a parent is predicted to be active if at least one of its assigned children is active. We then measure the Hamming distance between this child-based prediction and the ground-truth parent activation pattern. This metric evaluates the logical alignment of the hierarchy: a lower distance indicates that parent activations are a faithful abstraction of lower-level patterns.

Conditional co-activation probabilities. Beyond binary logical alignment, we evaluate the statistical dependency between levels through two complementary conditional probabilities, averaged over all assigned parent-child pairs:

- **Parent-given-child probability**: Defined as $P(\mathrm{P} \mid \mathrm{C}) = P(\sigma(e_p^T x) > 0 \mid \sigma(e_c^T x) > 0)$, this measures **hierarchical necessity**. A high probability indicates that a child feature rarely activates without its parent, ensuring that child features remain within their parents' semantic scope.

- **Child-given-parent probability**: Defined as $P(\mathrm{C} \mid \mathrm{P}) = P(\sigma(e_c^T x) > 0 \mid \sigma(e_p^T x) > 0)$, this measures the **semantic relevance** of each child feature to its parent. This metric evaluates the statistical strength of memberships within the hierarchy, where a higher value suggests that the parent feature is constituted by sub-components consistently co-active with it.

Results. As illustrated in Figure 2, HSAE consistently outperforms the post-hoc baseline across all measured metrics. The substantial reduction in Hamming distance indicates that HSAE's parent features provide a clearer and more complete logical summary of their constituent children. The improved co-activation probabilities further establish the statistical preconditions for semantic consistency, showing that the discovered hierarchical relations are functionally grounded rather than incidental consequences of vector similarity. Overall, these results indicate that jointly optimizing hierarchical relations and feature parameters extracts structure from the activation distribution more effectively than post-hoc analysis. A notable secondary finding is that Matryoshka SAE exhibits weaker alignment than JumpReLU SAE, which is expected because Matryoshka SAE explicitly reduces feature splitting and thus weakens cross-level hierarchical organization.

4.2 Auto-interpretability of hierarchies

To evaluate whether the discovered hierarchical relations correspond to genuine semantic abstractions, we develop an automated pipeline using LLMs as judges. For a given parent-child pair, we independently retrieve high-activation examples for each feature and present them to the LLM. The LLM is tasked with a binary classification: determining whether the two sets of examples exhibit a valid hierarchical conceptual relationship (Yes/No) and providing an associated confidence score. This automated setup allows for a large-scale evaluation of thousands of feature pairs, providing a statistically significant measure of semantic consistency. To assess alignment with human judgments, we

conduct a blinded study on 50 randomly sampled feature pairs, where the LLM's annotations achieve a Cohen's Kappa of 0.7165 with human annotations, indicating substantial agreement. Detailed sampling strategies, prompt templates, and LLM configurations are provided in Appendix C.

Results. Figure 2d reports the rate at which the LLM identifies a valid hierarchical relationship. Again, HSAE significantly outperforms post-hoc baselines. This alignment between semantic judgment and statistical metrics provides strong evidence for the structural faithfulness of HSAE.

4.3 Case studies

To provide intuition for the hierarchical structures captured by HSAE, we present empirical case studies in Figure 1 and more in Appendix D. As illustrated in Figure 1, HSAE decomposes an abstract root feature (e.g., *Science, Technology, and Research*) into coherent meso-scale features corresponding to its constituent domains. These features further branch into fine-grained features that follow distinct lexical or semantic patterns, such as specific scientific disciplines. Across these examples, we observe a consistent alignment between the learned hierarchy and human intuition, suggesting that the activation distribution itself exhibits a hierarchical structure.

We further analyze the geometry of learned features using UMAP projections. Specifically, we group activations by the features they activate and compare activation groups corresponding to sibling features against those corresponding to unrelated features at the same hierarchical level. We consistently observe that activations associated with sibling features form tight clusters, while those associated with unrelated features remain well separated. This pattern suggests that HSAE captures a structured organization of the latent space, where features sharing a common ancestor occupy nearby regions. Visualizations supporting this observation are provided in Figure 10 in Appendix D.

4.4 SAE feature quality

In this subsection, we evaluate HSAE from a standard SAE perspective by treating each level as an independent SAE. We first examine the reconstruction–sparsity trade-off by measuring the average L_0 and the fraction of variance explained, and then evaluate performance using the SAEBench evaluation protocol (Karvonen et al., 2025). Evaluation details are provided in Appendix E.

As shown in Table 1, at 16K dictionary size, HSAE achieves performance comparable to, and in some cases exceeding, baseline SAEs across most metrics. The overall results demonstrate that the explicit hierarchical structure captured by HSAE does not come at the expense of feature quality. Instead, HSAE provides these structural insights as an additional benefit while maintaining the competitive reconstruction and interpretability standards of modern SAEs. Evaluations for all levels are demonstrated in Section F.1 in Appendix F.

One notable result is the absorption score. Although not intended, HSAE successfully mitigates feature absorption as well. Despite substantial differences in implementation between HSAE and Matryoshka SAE, the similarity in both design principles and empirical outcomes suggests that introducing hierarchical structure may help mitigate the feature absorption problem. Additionally, our results illustrate that feature absorption can be reduced without eliminating feature splitting.

5 Applications

In this section, we present two distinct experiments to demonstrate the utility of HSAE. In the first experiment, we illustrate how the parent-child feature hierarchy can be leveraged to enhance the efficiency and quality of feature interpretation. In the second experiment, we demonstrate HSAE's superior capacity for capturing coarse-grained concepts relative to previous SAE methods. Together, these applications highlight the practical advantages of structural organization in feature discovery.

5.1 Hierarchy for feature interpretation

Standard SAE interpretation workflows are bottlenecked by the atomized nature of flat feature spaces, where features are treated independently and every feature must be interpreted from scratch. In contrast, HSAE organizes fine-grained features as refinements of higher-level parents, enabling semantic information to propagate through the hierarchy. We leverage this structure as a semantic scaffold for interpretation: high-level descriptions are reused as priors for interpreting descendant features, transforming interpretation from an exhaustive search into a guided multi-scale refinement process. This substantially reduces annotation overhead while improving descriptive precision.

To evaluate this, we conducted a systematic

Model	L_0	Var. Exp.	Absorption	AutoInterp	RAVEL	SCR	Sparse Probing
Hierarchical SAE	50.6	0.710	0.922	0.869	0.654	0.324	**0.873**
JumpReLU SAE	50.7	0.712	0.844	0.861	**0.661**	**0.338**	0.855
Matryoshka SAE	49.6	0.668	0.986	**0.882**	0.626	0.336	0.869

Table 1: Evaluation results comparing HSAE with JumpReLU SAE (JRSAE) and Matryoshka SAE (MSAE) at a dictionary size of 16K across multiple SAE benchmarks. Higher values indicate better performance for all metrics except L_0. Best downstream results are highlighted in **bold**.

study on 1,000 parent–child pairs extracted from our HSAE. We compared a standard AutoInterp baseline against our hierarchy-augmented approach. In the augmented setting, the LLM-based annotator is provided with the parent's interpretation as a "semantic prior" before generating the description for the child feature.

Our results show that leveraging hierarchical priors reduces the number of examples required for feature interpretation by *more than 50%* without sacrificing quality. While the standard AutoInterp baseline requires 15 examples per child feature to achieve an average score of 0.8505, our approach attains a comparable score of 0.8518 using only 3 examples. This improvement translates into substantial savings at scale. For a 4-level HSAE, interpreting all features requires 116,736 examples (2048×15 for roots and $(4096+8192+16384) \times 3$ for descendants), compared to 245,760 examples (16384×15) needed to interpret a flat 16K SAE layer. Additional experimental settings and results are provided in Appendix G.

5.2 Coarse-grained concepts steering

An important downstream application of SAEs is steering, where features are added to the activations to promote corresponding concepts during generation. However, standard SAEs present a trade-off: features in large dictionaries tend to capture fine-grained concepts, while features in smaller dictionaries often suffer from reduced quality. As a result, they lack reliable features for steering coarse-grained concepts, such as promoting a broad domain like science. In HSAE, high-level features are constrained to align with collections of lower-level features, allowing them to remain monosemantic while capturing coarse-grained concepts.

To evaluate this, we design an activation steering experiment targeting coarse-grained concepts. We select 16 broad concepts (e.g., Science, Coding), each represented by a diverse set of 8 target tokens. We apply steering by increasing the output logits of these tokens under different neutral prefixes (e.g., "The topic of this passage is about"). We measure performance by the average increase in logits over the target tokens. To reduce variance, results are averaged across 16 different prefixes. Detailed experimental settings are provided in Appendix G.

We evaluate SAEs at different scales, comparing the high-level 4K dictionary of HSAE with JumpReLU SAEs (4K, 16K) and Matryoshka SAE (16K). Since our goal is to study coarse-grained concepts, we compare against both small and large JumpReLU dictionaries for fairness, while only evaluating the 16K Matryoshka SAE because its nested design is explicitly intended to preserve high-level concepts within larger dictionaries. For each concept, we first use AutoInterp label to filter 10 candidate features in each SAE that align with the concept, then select the feature with the strongest steering effect.

HSAE(4K)	JRSAE(4K)	JRSAE(16K)	MSAE(16K)
1.653	0.897	0.762	1.052

Table 2: Average logit increases on target tokens. The increase is the higher the better.

As shown in Table 2, HSAE achieves the largest increase in target token logits, substantially outperforming the baselines. This suggests that HSAE features can promote broader semantic fields better than flat SAE features, highlighting HSAE's ability to capture higher-level concepts.

6 Limitations

Despite the improvements introduced by HSAE, several limitations remain. The current hierarchy uses a fixed depth, which may not capture the full variability of semantic abstractions. In addition, residual parent-child misalignments persist, though it remains unclear whether they stem from methodological limitations or inherently non-hierarchical representations. Future work could explore more flexible hierarchies and their scaling behavior.

References

Evan Anders, Clement Neo, Jason Hoelscher-Obermaier, and Jessica N. Howard. 2024. Sparse autoencoders find composed features in small toy models. https://www.lesswrong.com/posts/a5wwqza2cY3W7L9cj/sparse-autoencoders-find-composed-features-in-small-toy.

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

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, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. 2024. Batchtopk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. 2025. Learning multi-level features with matryoshka sparse autoencoders. In *Forty-second International Conference on Machine Learning*.

David Chanin, Tomáš Dulka, and Adrià Garriga-Alonso. 2025. Feature hedging: Correlated features break narrow sparse autoencoders. *arXiv preprint arXiv:2505.11756*.

David Chanin, James Wilken-Smith, Tom'avs Dulka, Hardik Bhatnagar, and Joseph Bloom. 2024. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *ArXiv*, abs/2409.14507.

Matthew A. Clarke, Hardik Bhatnagar, and Joseph Bloom. 2024. Compositionality and ambiguity: Latent co-occurrence and interpretable subspaces. https://www.lesswrong.com/posts/WNoqEivcCSg8gJe5h/compositionality-and-ambiguity-latent-co-occurrence-and.

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba Ba. 2025. From flat to hierarchical: Extracting sparse representations with matching pursuit. *arXiv preprint arXiv:2506.03093*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Joshua Engels. 2025. Negative results on group saes. https://www.lesswrong.com/posts/jKKbRKuXNaLujnojw/negative-results-on-group-saes. LessWrong blog post.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. 2025. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*.

Benjamin Heinzerling and Kentaro Inui. 2024. Monotonic representation of numeric attributes in language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 175–195, Bangkok, Thailand. Association for Computational Linguistics.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. 2024. Ravel: Evaluating interpretability methods on disentangling language model representations. *ArXiv*, abs/2402.17700.

Adam Jermyn and Adly Templeton. 2024. Ghost grads: An improvement on resampling.

Jean Kaddour. 2023. The minipile challenge for data-efficient language models. *ArXiv*, abs/2304.08442.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *ArXiv*, abs/2503.09532.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. 2024. Measuring progress in dictionary learning for language model interpretability with board game models. In *Advances in Neural Information Processing Systems*, volume 37, pages 83091–83118. Curran Associates, Inc.

Anton Korznikov, Andrey Galichin, Alexey Dontsov, Oleg Rogov, Elena Tutubalina, and Ivan Oseledets. 2025. Ortsae: Orthogonal sparse autoencoders uncover atomic features. *arXiv preprint arXiv:2509.22033*.

Patrick Leask, Bart Bussmann, Michael T Pearce, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. 2025.

Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations*.

Yuxiao Li, Eric J Michaud, David D Baek, Joshua Engels, Xiaoqing Sun, and Max Tegmark. 2025. The geometry of concepts: Sparse autoencoder feature structure. *Entropy*, 27(4):344.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Mark Muchane, Sean Richardson, Kiho Park, and Victor Veitch. 2025. Incorporating hierarchical semantics in sparse autoencoder architectures. *arXiv preprint arXiv:2506.01197*.

Anish Mudide, Joshua Engels, Eric J Michaud, Max Tegmark, and Christian Schroeder de Witt. 2025. Efficient dictionary learning with switch sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. https://distill.pub/2020/circuits/zoom-in.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. In *Causal Representation Learning Workshop at NeurIPS 2023*.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024a. Improving sparse decomposition of language model activations with gated sparse autoencoders. In *Advances in Neural Information Processing Systems*, volume 37, pages 775–818. Curran Associates, Inc.

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

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 176 others. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

Martin Wattenberg and Fernanda Viégas. 2024. Relational composition in neural networks: A survey and call to action. In *ICML 2024 Workshop on Mechanistic Interpretability*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *ArXiv*, abs/2505.09388.

Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. 2025. Interpreting clip with hierarchical sparse autoencoders. *arXiv preprint arXiv:2502.20578*.

A Detailed settings

A.1 Experiment setups

Data preparation. For all experiments, we collect a dataset of 100M activation vectors from the target LLM. These activations are generated using input prompts with a sequence length of 1024 tokens. Following standard practice, we exclude activations from the `bos` token position. All activation vectors are normalized with a constant scaler to have a RMS norm expectation of $\sqrt{d}$ before being used for training.

Artifact licenses and terms of use. We use only publicly available artifacts and follow their respective licenses and terms of use, including the pretrained models (`gemma2-2b` and `qwen3-4b`), the mini-PILE dataset, and the SAEBench framework. We do not redistribute third-party artifacts and report only derived evaluation results. Our use is consistent with the intended research use of these artifacts: we use them only for analysis and evaluation, and we do not deploy derived artifacts in production or non-research settings.

Training hyperparameters. We train all models using a batch size of 1024 with the Adam optimizer ($\beta_1 = 0, \beta_2 = 0.995$). We employ a cosine learning rate scheduler, starting with a linear warm-up phase covering the first 10% of training steps, followed by a decay to zero.

Computational resources and efficiency. All training and evaluation tasks are conducted on NVIDIA A800 GPUs. For a target model with hidden dimension $d = 2304$, training a 4-level HSAE (with dictionary sizes ranging from 2048 to 16384) takes approximately 6 hours, compared to

4.5 hours for training independent SAEs of equivalent sizes. The additional overhead in HSAE is modest, and primarily arises from computing the hierarchical constraint loss and the memory latency introduced by fragmented indexing in the feature perturbation step.

A.2 Dynamic sparsity control

To maintain a consistent level of sparsity throughout training, we implement a feedback control mechanism to dynamically adjust the sparsity penalty weight λ_ℓ. Specifically, we define a target sparsity $\hat{L}_0$ and model the desired trajectory of the average L_0 as a first-order exponential decay process:

$$\Delta L_0 = -\eta(L_0 - \hat{L}_0)$$

where η represents the decay rate, set to 0.001 in our experiments. We track the current L_0 using an exponential moving average (EMA) with a momentum of 0.999 to filter out high-frequency noise from individual batches.During each training step, we compare the observed change in L_0 against the theoretical ΔL_0 derived from the exponential model. If the observed reduction in sparsity is more rapid than the target trajectory, the sparsity weight λ_ℓ is decreased; conversely, if the model remains denser than expected, λ_ℓ is increased. In practice, we found that this velocity-based control significantly reduces oscillations in training compared to the control methods described in (Karvonen et al., 2025).

B Ablation studies

B.1 Training Mechanisms

To isolate the contributions of different components in HSAE, we study two mechanisms for enforcing hierarchical alignment: **the parent-child constraint loss** and **random feature perturbation**. We evaluate their effects by measuring the Hamming distance between parent and child activations.

The full HSAE achieves the lowest Hamming distance of 21.3. Removing the constraint loss leads to a substantial degradation, increasing the distance to 29.7, indicating that explicit supervision is the primary driver of alignment. Using the constraint loss alone reduces the distance, but remains higher than the full model. Incorporating random perturbation further improves alignment, yielding an additional reduction of 0.9. These results show that the two mechanisms play complementary roles: the constraint loss establishes coarse alignment, while random perturbation refines it to better capture fine-grained structure.

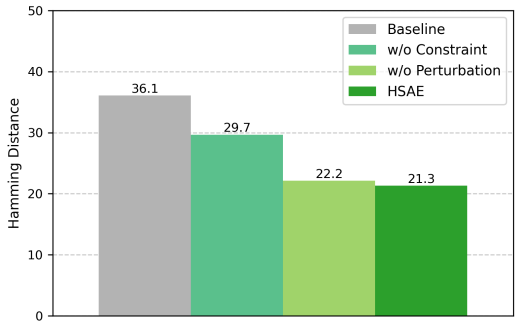


Figure 3: **Ablation of training mechanisms.** We compare the parent-child alignment across: (1) *Baseline*: independently trained SAEs with post-hoc alignment; (2) *w/o Constraint*: without parent-child constraint loss; (3) *w/o Perturbation*: without random perturbation; and (4) *HSAE*: our complete implementation.

B.2 Similarity Metric for Hierarchy Updates

We investigate the impact of different similarity metrics used during the hierarchy update. As shown in Figure 4, we compare three potential proxies for feature relatedness: (1) **Co-activation** (statistically derived co-activation probability), (2) **Decoder** (cosine similarity between decoder vectors), and (3) **Encoder** (cosine similarity between encoder vectors).

Our results indicate that all three metrics yield nearly indistinguishable performance in terms of the final hierarchical consistency. Since the choice of similarity metric does not significantly alter the outcome, we adopt Encoder similarity as the default for all experiments, as it offers the most direct and computationally efficient implementation within our training pipeline.

B.3 Hyperparameter tuning

We examine the impact of key hyperparameters on HSAE performance, specifically the parent-child constraint weight ρ and the random perturbation rate r. These parameters are essential for effective hierarchical training; as expected, setting both to zero causes the method to degenerate into independently trained SAEs.

Our analysis reveals a nuanced trade-off: while hierarchical constraints guide structured feature discovery, an excessively large constraint weight ρ imposes overly rigid priors that can bottleneck representational capacity and reduce Variance Explained. Similarly, a perturbation rate r that is too

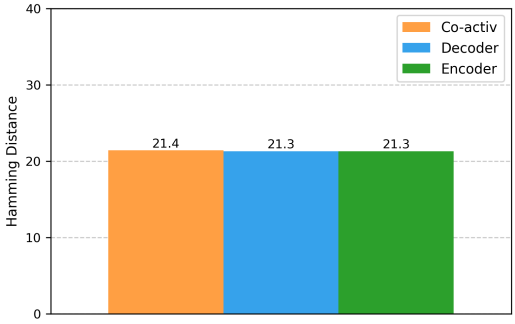


Figure 4: **Ablation of similarity metric used for hierarchy update.** (1) *Co-activ*: Use statistically derived co-activation probability as similarity; (2) *Decoder*: Use decoder vector similarity; (3) *Encoder*: Use encoder vector similarity. All three metrics provide similar results.

high introduces excessive noise into training, potentially causing instability and degraded feature quality. In contrast to the moderate settings used in the main experiments, these extreme configurations expose a non-negligible trade-off between hierarchical consistency and reconstruction performance. We visualize these relationships across parameter regimes in Figure 5.

The final hyperparameter selection reflects a principled balance between reconstruction fidelity and hierarchical consistency. We prioritize configurations that preserve fidelity—ensuring that the SAE captures the original activation space—while achieving strong hierarchical alignment.

B.4 Tree structural assumptions

We further investigate how different architectural assumptions regarding the tree topology affect hierarchical alignment. Specifically, we compare the default HSAE with three alternative configurations: **Binary Tree**, which restricts each parent to a maximum of two children; **Fix Tree**, where the hierarchy is determined early in training and remains static; and **Full Tree**, which forcefully assigns every feature to a parent.

As shown in Figure 6, the default HSAE outperforms all constrained variants. The performance gap compared to the Binary Tree value of 22.8 suggests that conceptual decomposition in LLMs is naturally multi-branching rather than strictly dyadic. The significantly higher Hamming distance of the Fix Tree at 24.2 highlights the necessity of an alternating optimization process, as feature representations and their optimal hierarchical assignments must co-evolve during training. Finally, the Full Tree baseline yields sub-optimal consistency with a distance of 24.0. This indicates that the more flexible Partial Tree structure is necessary for capturing a more robust hierarchical structure of LLM activations, as it avoids forcing assignments for features that do not exhibit strong similarity to any potential parent.

C Auto-interpretability of hierarchies

This appendix provides implementation details for the auto-interpretability assessment described in Section 4.3. Our pipeline extends the SAEBench evaluation framework (Karvonen et al., 2025) with custom modifications to evaluate the hierarchical relations between HSAE features.

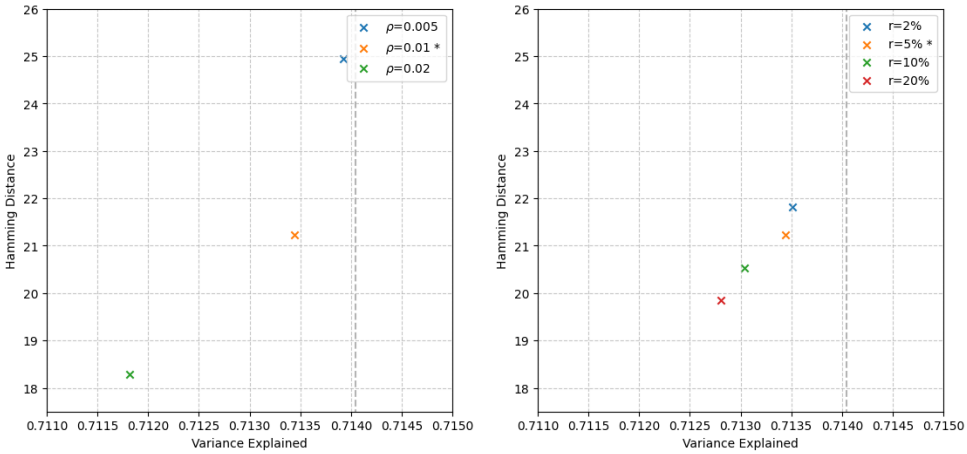
Feature Pairs Sampling and Data Collection. We randomly sample 5000 parent-child pairs from the learned hierarchical structure in HSAE. Activations are computed on sequences from the mini-PILE testing dataset. Following the SAEBench methodology, for each feature we collect 10 sequences with the highest activation values and sample 5 sequences with probability proportional to their activation values. These sequences are formatted by highlighting activating tokens with «token» syntax.

LLM Judge Configurations. We employ `Qwen3-Max` as the judge model.

The system prompt is as follows:

```
We're studying neurons in a sparse autoencoder (SAE) within a neural network. Each neuron activates on specific words, substrings, or concepts in short documents, with activating words indicated by « ... ». You will be given two sets of documents where two different neurons activate. Your task is to compare the activation patterns of these two neurons and determine if there is a parent-child relationship between them. A parent-child relationship means that one neuron's activating concept (the child) is a subset or a more specific version of the other neuron's activating concept (the parent). Analyze the provided examples and output your judgment in the following format:

HaveRelationship: [Yes/No]
Confidence: [High/Medium/Low]
```



(a) Parent-Child Constraint Weight ρ

(b) Random Perturbation Rate r

Figure 5: **Hyperparameters' effects to reconstruction-hierarchy consistency trade-off.** Increasing parent-child constraint weight ρ or perturbation rates r improve hierarchical alignment but eventually degrades Variance Explained.

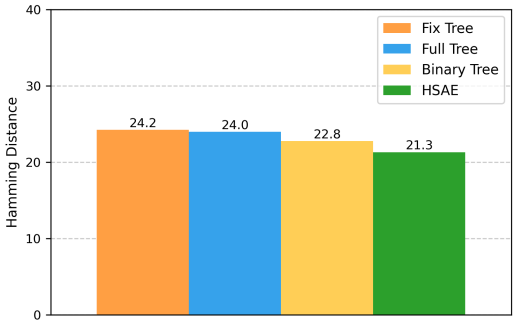


Figure 6: **Ablation of tree structural assumptions.** We compare parent-child alignment across different structural variants: (1) *Fix Tree*: static hierarchy after early initialization; (2) *Full Tree*: mandatory assignment for all features; (3) *Binary Tree*: a hierarchy constrained to a maximum branching factor of two; and (4) *HSAE*: our flexible partial tree approach.

`Do not include any additional text, explanations, or formatting.`

The user prompt template:

`Here are the activating documents for Neuron A:`
`[list of examples with activating tokens highlighted]`

`And for Neuron B:`
`[list of examples with activating tokens highlighted]`

`Based on these documents, determine if Neuron A and Neuron B have a parent-child relationship.`

D More case studies

In this section, we provide further qualitative evidence of the hierarchical structures discovered by HSAE.

D.1 Empirical examples

Figure 7 illustrates the conceptual organization of *Time*. Here, a coarse-grained root feature captures general temporal references. In the subsequent level, it is partitioned into features representing daily timescales versus longer-term durations (e.g., weeks, months), reflecting a clear organizing principle aligned with human cognition. The daily timescale branch further resolves into specialized sub-domains, segregating immediate deictic markers (e.g., "today") from the occurrences of "day" and sub-daily intervals. These examples suggest that the decompositions learned by HSAE are not merely stochastic. Rather, they consistently align with human intuition and possess inherent interpretability across semantic scales. We have provided more empirical studies in Appendix D.

Figure 8 illustrates the hierarchical organization of *Financial Resource Concepts*. A broad root feature (#1967) serves as the semantic anchor, which the model partitions into distinct functional domains: capital acquisition and fiscal status, categorized as *Funding Flows and Financial Capacity* (#3062), and the execution of financial duties, labeled as *Financial Transactions and Obligations* (#4829). Within these branches, the hierarchy further resolves into specialized leaf features that cap-

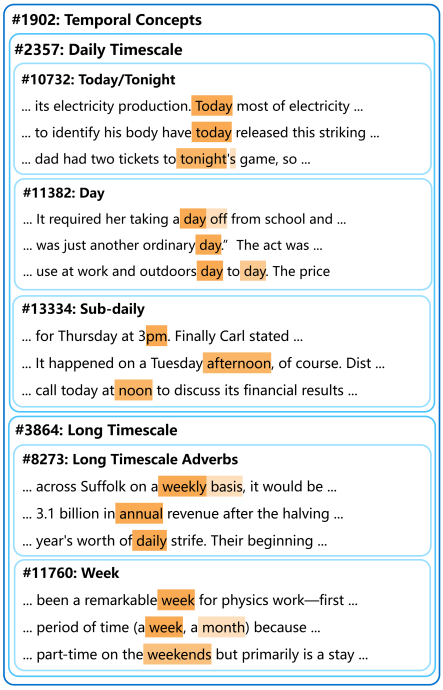


Figure 7: **Hierarchical feature discovery of temporal concepts.** The root broad temporal feature (#1902) branches into a *Daily Timescale* (#2357) and a *Longer Timescale* (#3864). At the next level, the daily timescale branch splits into *Today/Tonight* (#10732), the unit concept *Day* (#11382), and *Sub-daily* intervals including morning/afternoon or am/pm (#13334). Similarly, longer-term concepts are partitioned into *Adverbs* (#8273) and specific units like *Week* (#11760).

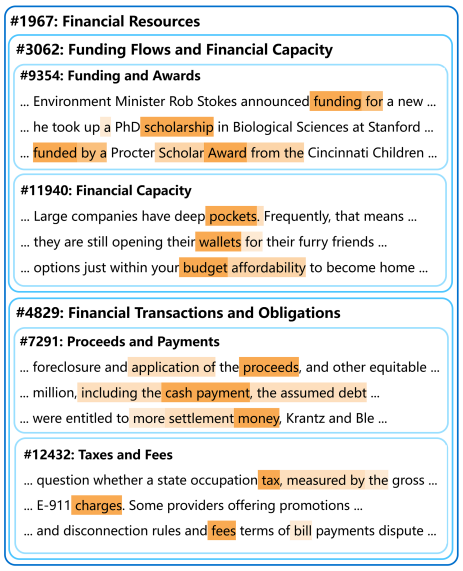


Figure 8: **Hierarchical feature discovery of financial resource concepts.** The root feature (#1967) representing financial concepts branches into *Funding Flows and Financial Capacity* (#3062) and *Financial Transactions and Obligations* (#4829). Within the *Funding Flows* branch, HSAE further distinguishes between external support (*Funding and Awards*, #9354) and internal economic status (*Financial Capacity*, #11940). Within the latter, the hierarchy resolves into granular features for specific settlements (*Proceeds and Payments*, #7291) and statutory costs (*Taxes and Fees*, #12432).

ture subtle semantic nuances. For instance, the model distinguishes between formal institutional support (e.g., *scholarship*, *awards* in #9354) and informal or abstract expressions of wealth (e.g., *deep pockets*, *budget* in #11940). Concurrently, the transactional branch successfully separates commercial settlements (e.g., *proceeds*, *cash payment* in #7291) from mandatory statutory costs (e.g., *tax*, *fees* in #12432).

Figure 9 illustrates the hierarchical organization of *Writing and Language Concepts*. A broad root feature (#1307) serves as the semantic anchor, which the model partitions into distinct functional domains: active processes such as *Writing* (#2378) and *Reading Actions* (#4370), and their corresponding outputs, categorized as *Text Content* (#5006) and *Written Material* (#3681). Within the *Written Material* branch, the hierarchy further resolves into granular sub-features that distinguish between structured digital formats (*document*, #7996) and physical media (*paper*, #8366).

D.2 Geometric manifestation of hierarchy

To understand how the discovered hierarchy manifests in the original activation space, we visualize the activation clusters using UMAP projection in Figure 10. Each plot contrasts the activations triggered by a set of sibling features (which share the same parent) against those triggered by an unrelated feature from a distant branch within the same hierarchical level.

We observe that unrelated features remain clearly isolated from sibling clusters. This geometric arrangement suggests that HSAE's structural constraints successfully capture the natural clustering of the LLM's latent space, where features with a common hierarchical ancestor are encoded in nearby geometric regions.

While sibling features exhibit spatial proximity in the UMAP visualization, we should note that their co-activation rate does not increase compared to baseline SAEs. This indicates that HSAE learns a disentangled decomposition rather than simply clustering redundant features. Specifically, although the parent-child constraint encourages

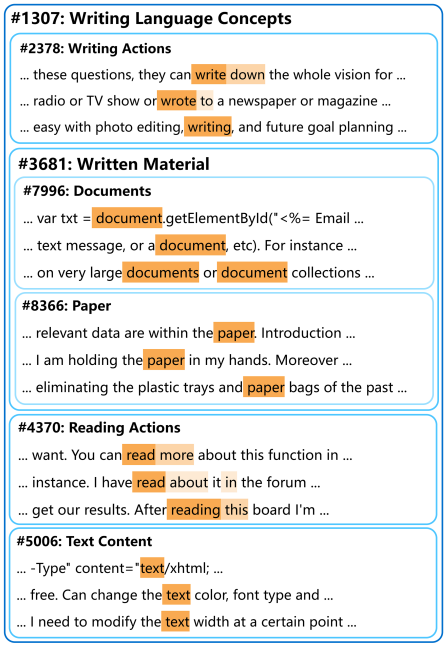


Figure 9: **Hierarchical feature discovery of writing and language concepts.** The root feature (#1307) representing broad language-related concepts branches into specialized domains: *Writing Actions* (#2378), *Written Material* (#3681), *Reading Actions* (#4370), and *Text Content* (#5006). Within the *Written Material* branch, HSAE further distinguishes between digital or structured documents (*Documents*, #7996) and physical or raw media (*Paper*, #8366).

structural organization, it does not force sibling features to fire simultaneously on the same tokens. On the contrary, the sparsity penalty inherent in the SAE objective naturally discourages such co-activation, ensuring that each sibling feature remains a distinct and sparse functional unit.

D.3 Branching factor and semantic clarity

We further investigate how a feature's structural properties correlate with its interpretability. Specifically, we compare the AutoInterp scores of root features having only a single child against those that branch into multiple children. We find that multi-child roots exhibit a 2.49% higher average score—a performance delta comparable to the interpretability gain typically achieved by quadrupling the dictionary size (from 16k to 64k) in standard SAEs. This suggests an underlying link between a feature's hierarchical positioning and the semantic clarity of its represented concept. Corresponding results are plotted in Figure 11.

E SAEBench evaluation details

We evaluate our models using the SAEBench framework (Karvonen et al., 2025), adopting the default configurations for comparability. Since SAEBench provides a diverse set of metrics, we specify the exact metric identifiers used for our reporting in Table 3. Notably, for Absorption, we report (1 − score) to align it with other metrics where a higher value indicates superior performance.

Benchmark	Metric
Var. Exp.	`explained_variance_legacy`
Absorption	`mean_absorption_fraction_score*`
AutoInterp	`autointerp_score`
RAVEL	`disentanglement_score`
SCR	`scr_metric_threshold_20`
Sparse Probing	`sae_top_5_test_accuracy`

Table 3: Mapping of reported benchmarks to specific SAEBench metrics.

F More evaluations

Reporting convention. Unless otherwise specified, all reported results are mean values computed over the corresponding evaluation set (e.g., parent-child pairs, activation samples, or hierarchical levels).

F.1 All dictionary sizes

We report SAEBench results across all dictionary sizes for HSAE and baseline methods in Sec-

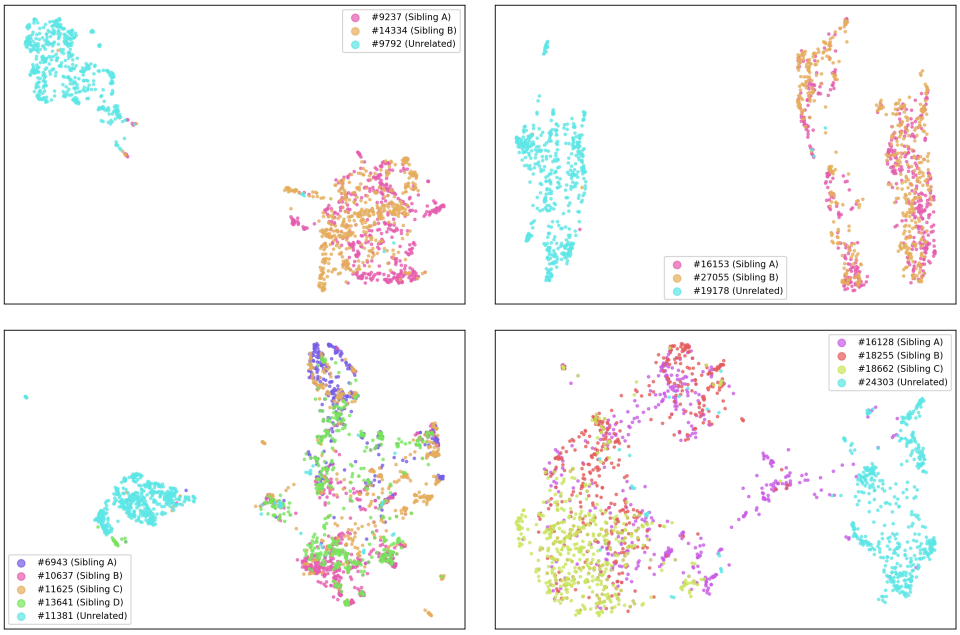


Figure 10: **Geometric manifestation of hierarchical relations.** UMAP projection of activations that trigger different features. In all cases, the unrelated feature is clearly separated from sibling features.

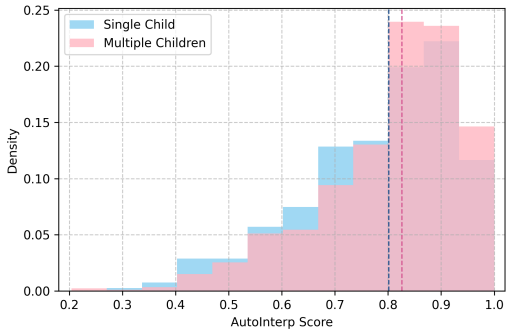


Figure 11: **Interpretability versus structural branching.** Histogram of AutoInterp scores for root features with single versus multiple children. Features with multiple children tend to be more interpretable than those with a single child. The average AutoInterp score gap is 2.49%, comparable to the performance gain achieved by quadrupling the dictionary size (from 16k to 64k) in standard SAEs.

tion F.1. The table summarizes reconstruction and interpretability metrics under a consistent evaluation protocol.

F.2 More activation sources and target LLMs

To evaluate the generalizability of HSAE, we extend our experiments across multiple dimensions. First, we move beyond the 13th layer of `gemma-2-2b` to train HSAE on residual stream activations from early and late stages (layers 6 and 20). Second, we apply our method to the 18th layer of `qwen3-4b` (Yang et al., 2025) to verify cross-model consistency. Finally, we test the robustness of the discovered hierarchy under varying capacity constraints by training on `gemma-2-2b-layer-13` with target sparsities $L_0 \in \{80, 100\}$.

As summarized in Table 5, HSAE consistently maintains reconstruction fidelity comparable to the baselines across all tested settings. Here, the reported L_0 and Variance Explained represent the average performance across all hierarchical levels. Simultaneously, it yields significant improvements in parent-child alignment and co-activation probabilities. These results collectively demonstrate HSAE's ability to recover structured feature taxonomies is not dependent on specific layer depths, model architectures, or sparsity levels.

We note a slight discrepancy in the *Variance Explained* scores for `gemma2-2b-layer-13` between Section F.1 and Table 5. This shift is attributed to the different evaluation corpora employed: the benchmarks in Section F.1 strictly follow the SAEBench protocol using OpenWebText, whereas the results in Table 5 are computed using the mini-PILE testing dataset. Despite these distributional shifts, the reconstruction fidelity between HSAE and baseline remain comparable.

Model	Dict. Size	L_0	Var. Exp.	Absorp.	AutoInterp	RAVEL	SCR	SP
Hierarchical SAE	2048	49.4	0.612	0.983	0.807	0.521	0.272	0.868
	4096	50.0	0.651	0.981	0.814	0.582	0.286	0.873
	8192	50.5	0.685	0.951	0.859	0.627	0.314	0.874
	16384	50.6	0.710	0.922	0.869	0.654	0.324	0.873
JumpReLU SAE	2048	49.4	0.613	0.988	0.813	0.539	0.245	0.863
	4096	49.8	0.651	0.974	0.829	0.609	0.302	0.879
	8192	50.3	0.684	0.901	0.854	0.627	0.320	0.862
	16384	50.7	0.712	0.844	0.861	0.661	0.338	0.855
Matryoshka SAE	2048	49.0	0.594	0.996	-	0.492	0.215	0.845
	4096	49.2	0.622	0.990	-	0.560	0.283	0.848
	8192	49.6	0.646	0.996	-	0.592	0.293	0.851
	16384	49.6	0.668	0.986	0.882	0.626	0.336	0.869

Table 4: Evaluation results for Hierarchical SAE, JumpReLU SAE and Matryoshka SAE at different dictionary size on multiple SAE benchmarks. All metrics except L_0 are the higher the better.

Activation Source	Model	L_0	Var. Exp.	$P(\mathrm{P} \mid \mathrm{C})$	$P(\mathrm{C} \mid \mathrm{P})$	Ham. Dist. (↓)
gemma2-2B-layer-6	HSAE	50.1	0.759	0.752	0.403	23.2
	JRSAE	50.0	0.759	0.651	0.331	34.6
gemma2-2b-layer-13	HSAE	50.1	0.713	0.769	0.432	21.3
		80.1	0.743	0.749	0.421	36.2
		100.7	0.757	0.747	0.415	46.3
	JRSAE	49.8	0.714	0.662	0.343	36.1
		80.0	0.743	0.634	0.326	61.9
		100.0	0.758	0.619	0.317	78.9
gemma2-2b-layer-20	HSAE	50.0	0.729	0.800	0.461	19.4
	JRSAE	50.0	0.727	0.729	0.395	30.6
qwen3-4b-layer-18	HSAE	50.2	0.988	0.347	0.197	28.7
	JRSAE	50.0	0.988	0.309	0.179	46.4

Table 5: Evaluation results for HSAE and baseline SAEs on multiple benchmarks. All metrics except L_0 and Hamming Distance are the higher the better. Hamming Distance is the lower the better.

G Applications

G.1 Hierarchy for feature interpretation

Experimental setup. We randomly sample 1,000 parent–child pairs from the HSAE hierarchy trained on `gemma-2-2b-layer-13`. Parent interpretations are obtained via the standard AutoInterp framework (Karvonen et al., 2025) with 15 activation examples, using `Qwen3-Max` as the judge. For child features, we compare two context conditions: (i) *Without parent*: no additional context is provided; the judge sees only the child's activation examples. (ii) *With parent*: the parent's interpretation (obtained from 15 examples) is prepended to the prompt before presenting the child's examples. Within each condition, we vary the number of child activation examples (3 or 15) to measure the efficiency–accuracy trade-off. We further examine the effect of parent interpretation quality by separately reporting results on the subset where the parent's own AutoInterp score exceeds 0.9 (approximately 45% of all pairs).

Results. Section G.1 presents the full set of AutoInterp scores. Several observations are noteworthy.

Providing parent context consistently improves interpretation quality at low example budgets on the full set of pairs. With only 3 child examples, the score rises from 0.8384 (*Without parent*) to 0.8518 (*With parent*), an improvement of 1.34 percentage points. Crucially, the parent-augmented 3-example score (0.8518) closely matches the 15-example baseline without parent context (0.8505), demonstrating that parent interpretations can effectively replace the majority of examples otherwise required by the child.

When focusing on the scenario where both setups use 15 child examples, parent context yields only a marginal gain on the full set of pairs (0.8505 vs. 0.8507), suggesting that 15 examples alone already capture a large portion of the interpretability signal for a typical parent–child pair.

Furthermore, the benefit of parent context becomes more pronounced when the parent interpretation itself is highly reliable. On the subset with parent AutoInterp score > 0.9, adding parent context lifts the child score from 0.9231 to 0.9299, a gain of 0.68 percentage points. This improvement is meaningful within the narrow AutoInterp score range typically observed across SAE architectures (often only a few percentage points separating strong and weak methods), and it indicates that high-quality parent abstractions provide a useful inductive bias even when the child already has abundant examples.

G.2 Coarse-grained feature steering

Here we list the detailed datasets used in coarse-grained feature steering experiments, including coarse-grained concepts and the corresponding target tokens, neutral prefix, and some detailed experimental settings. Notice that according to different tokenizer used, certain words will be truncated into prefix pieces.

Dataset. The coarse-grained concepts we studied are listed in Section G.2. The neutral prefix used to calculate the logit is listed in Section G.2.

Experimental setup. We evaluate logits only at the first inference position, with no rollout being performed. To ensure a fair comparison, we compute the normalized logit increase by subtracting the average logit increase over all tokens from the average over all target tokens. We only perform activation steering on the last inference position and with a strength factor of 0.2.

H AI assistant usage statement

We used AI assistants for writing and editing support (e.g., grammar polishing, wording refinement, and checklist drafting) and for minor coding assistance in manuscript maintenance. All technical decisions, experiment design, implementation choices, result interpretation, and final manuscript content were verified and approved by the authors.

Context	Subset	# Examples	AutoInterp Score
Without parent	All parents	3	0.8384
	All parents	15	0.8505
	Parent score > 0.9	15	0.9231
With parent	All parents	3	0.8518
	All parents	15	0.8507
	Parent score > 0.9	15	0.9299

Table 6: AutoInterp scores under different context conditions and example budgets.

Concepts	Representative Words
1. Science	physics, biology, chemistry, quantum, laboratory, evolution, telescope, molecule
2. Coding	Python, Java, compiler, algorithm, debug, database, repository, syntax
3. Mathematics	calculus, geometry, statistics, equation, theorem, prime, matrix, integral
4. Finance	investment, inflation, bankruptcy, portfolio, liquidity, dividend, currency, mortgage
5. Law	constitution, litigation, jurisdiction, attorney, felony, contract, legislation, verdict
6. Medicine	diagnosis, pathology, antibiotics, surgery, vaccine, neurology, anatomy, prescription
7. History	civilization, revolution, dynasty, archaeology, renaissance, empire, medieval, treaty
8. Philosophy	aesthetics, ethics, epistemology, logic, existentialism, dialectic, stoicism, paradox
9. Geography	archipelago, glacier, continent, latitude, plateau, topography, desert, canyon
10. Arts	sculpture, surrealism, symphony, canvas, choreography, masterpiece, galleries, aesthetic
11. Literature	metaphor, protagonist, anthology, narrative, fiction, sonnet, manuscript, biography
12. Politics	democracy, parliament, diplomacy, ideology, sanctions, suffrage, lobbyist, manifesto
13. Sports	championship, marathon, stadium, athletics, quarterback, tournament, endurance, referee
14. Technology	semiconductor, robotics, automation, fiber-optic, hardware, broadband, encryption, drone
15. Religion	scripture, theology, pilgrimage, ritual, monastery, prophecy, cathedral, doctrine
16. Cooking	fermentation, seasoning, patisserie, cuisine, nutrition, ingredients, convection, marinate

Table 7: Coarse-Grained Concepts and Representative Words

Neutral Prefix Phrase
The topic of this passage is about
This text provides a discussion on
The following content focuses on
In this context, we examine
This section is dedicated to
The subject matter involved is
A central theme of this writing is
This discourse explores the area of
Information presented here relates to
The subsequent paragraphs describe
This document primarily concerns
We can observe concepts related to
The primary focus of this analysis is
This specific material pertains to
Broadly speaking, this covers
The essential focus of this excerpt is

Table 8: Neutral Semantic Prefixes for Activation Inversion