FROM ATOMS TO TREES: BUILDING A STRUCTURED FEATURE FOREST WITH HIERARCHICAL SAES

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) have proven effective for extracting monosemantic features from large language models (LLMs), yet these features are typically identified in isolation. Feature splitting and cross-feature organization suggest that SAE dictionaries may admit useful coarse-to-fine structure, although splitting alone does not establish a semantic hierarchy. To study such structure, 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 perturbation of eligible non-root features. At matched L_0, HSAE retains nearly the same variance explained as JumpReLU while achieving the best RAVEL, SCR, and sparse-probing scores among the evaluated methods, including clear gains over the framework-controlled H-SAE and Tree SAE reimplementations. Across layers and models, HSAE improves cross-level activation alignment over post-hoc hierarchies built from independently trained SAEs, while case studies and large-scale hierarchy judgments show coarse-to-fine semantic distinctions. Finally, we demonstrate two applications of the learned hierarchy: reusing parent descriptions for feature interpretation and steering broad concepts with coarse-level features.

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 have non-trivial internal organization: recent work reveals spatial clustering by semantic groups (Li et al., 2025) and co-activation between abstract and specialized features (Clarke et al., 2024). Since navigating flat, unstructured dictionaries is inherently difficult, extracting coarse-to-fine structure could support organized, multi-scale interpretation. *Feature splitting*—where broad concepts break into finer sub-features as dictionary size increases (Bricken et al., 2023)—is one possible source of such structure. Prior work often treats splitting as an obstacle to identifying universal, canonical features (Leask et al., 2025; Bussmann et al., 2025); here we instead use it to construct a hierarchical index over the feature space. This use does not assume that every split reflects a semantic generalization–specialization relation.

For this purpose, we introduce the **Hierarchical Sparse Autoencoder (HSAE)**. 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 partial tree. Model parameters and the parent–child assignments co-evolve through alternating optimization. The training objective enforces a functional dependence across levels, which is distinct from directional conceptual subsumption: (1) a **structural constraint loss** asks each parent feature's decoder contribution to be reconstructed by the sum of its assigned children, and (2) a **random feature perturbation**

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.

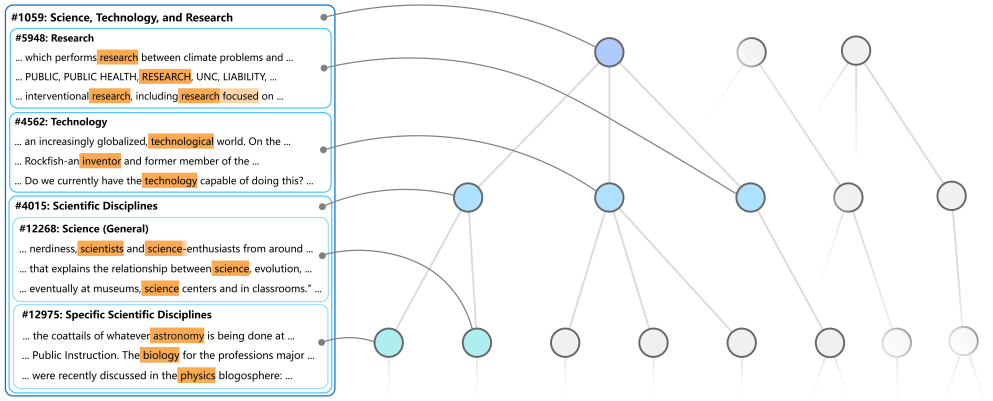


Figure 1: **A selected branch from the HSAE feature forest.** We visualize one learned branch alongside a semantic dashboard of its nodes. Each node includes a unique feature index (e.g., #1059), a human-annotated semantic label, and representative top-activating context snippets. Orange highlights indicate the activation positions, with color intensity reflecting the activation magnitude. In this example, a broad feature labeled *Science, Technology, and Research* (#1059) connects to a more specific feature labeled *Scientific Disciplines* (#4015). This intermediate node further connects to a feature capturing lexical patterns containing the word *science* or *scientists* (#12268) and a feature responding to discipline names such as *biology* or *physics* (#12975). This selected branch is illustrative rather than evidence that every learned edge has this interpretation. For clarity, only a three-layer subgraph and examples of some features are shown.

mechanism substitutes an eligible non-root parent's contribution with the sum of its children during reconstruction. The constraint applies to all assigned parent groups, whereas direct substitution applies only to non-root features that themselves have children. These mechanisms align parent and child features in decoder space. Whether an edge also recovers a pre-existing semantic specialization is a separate empirical question, addressed only in part by the case studies and the LLM judge below.

We apply HSAE to residual-stream activations from multiple layers and models. The learned forest provides a multi-level index over features, and selected branches exhibit progressively finer semantic patterns. For example, Figure 1 moves from a broad science-related feature to lexical and discipline-specific descendants. Our activation-alignment results evaluate this organization at scale, while the semantic reading is further assessed by selected cases and a directional LLM subsumption judgment.

We evaluate the learned edges with activation-alignment metrics and an LLM judge, and compare them with post-hoc hierarchies on independently trained SAEs. Complementary case studies illustrate branches that are plausible semantic taxonomies. On standard feature-quality benchmarks, HSAE preserves JumpReLU-level reconstruction and outperforms the framework-controlled H-SAE and Tree SAE reimplementations on RAVEL, SCR, and Sparse Probing. Ablations isolate the contribution of each proposed mechanism to the activation-alignment objective.

We also study two applications. Reusing a parent description lets an automated interpreter match a fifteen-example child description with three child examples, on the same AutoInterp scorer that produced the parent text. In a single-position logit probe, high-level HSAE features selected for steering strength raise target-token logits more than the flat SAE features we compare against. Both results are preliminary: neither is a held-out human study of annotation cost, nor a generation-level test of steering.

The main contributions of this work are:

- **A hierarchical architecture:** We propose HSAE, which jointly trains SAE levels and a dynamically reassigned partial parent–child forest using a reconstruction constraint and non-root feature perturbation.

- **Hierarchy evaluation:** We measure logical-OR agreement and pairwise co-activation against post-hoc hierarchies, and use a directional LLM judge to test whether the designated parent semantically subsumes its child. HSAE scores higher on these metrics, although the activation metrics test the training constraint more directly than an external semantic ontology.
- **Applications with a stated scope:** Parent descriptions reduce the number of child examples needed by an automated interpreter. A single-position logit probe suggests that high-level HSAE features can promote broad concepts. We do not treat either result as a completed human evaluation of annotation cost or generation quality.

2 Related work

2.1 Discovering structures in activations

Linear regularities first observed in word embeddings (Mikolov et al., 2013a;b) also appear in LLM representations (Gurnee & Tegmark, 2024; Heinzerling & Inui, 2024). Together with the linear representation hypothesis (Park et al., 2023), this motivates SAEs as overcomplete decompositions of activations into interpretable directions (Bricken et al., 2023; Cunningham et al., 2023).

SAE feature spaces exhibit splitting (Bricken et al., 2023), functional clusters (Li et al., 2025), low-dimensional manifolds (Engels et al., 2025), and hub-and-spoke co-activation patterns (Clarke et al., 2024). O'Neill, Ye, and coauthors recover parent-like dense features and co-activating sparse children in SAEs of text embeddings (O'Neill et al., 2024), while Llama Scope studies feature geometry and splitting across all sublayers of Llama-3.1-8B (He et al., 2024). These results motivate structured SAE dictionaries, although our setting uses LLM residual streams and learns cross-dictionary assignments jointly.

2.2 Feature splitting

Feature splitting occurs when a broad feature decomposes into specialized latents as dictionary size grows (Bricken et al., 2023). Related failures include composition, where co-occurring concepts merge (Anders et al., 2024; Wattenberg & Viégas, 2024), and absorption, where a general feature misses subcases represented elsewhere (Chanin et al., 2024). These effects arise partly from sparse reconstruction objectives and are measured by dedicated feature-quality benchmarks (Huang et al., 2024; Chanin et al., 2024; Karvonen et al., 2025).

Several methods instead suppress splitting or correlated redundancy (Bussmann et al., 2025; Chanin et al., 2025; Korznikov et al., 2025; Jin et al., 2026); for example, C^2R reduces splitting and absorption without learning a hierarchy (Jin et al., 2026). We use splitting as a source of multi-scale structure, but do not assume that it proves a semantic tree: one-dimensional latents can tile a multi-dimensional concept (Engels et al., 2025; Dalili & Mahdavi, 2026). Our objective does not distinguish such geometric fragmentation from conceptual specialization.

2.3 Architectural evolution of SAEs

SAEs have progressed from L_1-regularized ReLU models (Bricken et al., 2023; Cunningham et al., 2023) through improved gating, sparsity control, activation functions, training dynamics, and scaling (Rajamanoharan et al., 2024a; Gao et al., 2025; Bussmann et al., 2024; Rajamanoharan et al., 2024b; Karvonen et al., 2024; Mudide et al., 2025). Meta-SAEs and attribution-guided distillation further decompose or refine learned dictionaries (Leask et al., 2025; Martin-Linares & Ling, 2025).

Structural approaches include nested dictionaries, matching pursuit, group constraints, and hierarchical CLIP or language-model SAEs (Bussmann et al., 2025; Costa et al., 2025; Engels, 2025; Zaigrajew et al., 2025; Muchane et al., 2025). Matryoshka SAE provides implicit multi-scale organization through nested dictionaries without explicit parent–child edges (Bussmann et al., 2025); H-SAE uses a fixed two-level additive hierarchy selected before training (Muchane et al., 2025; Park et al., 2023); and concurrent Tree SAE learns an explicit multi-level tree with parent gating and decoder reconstruction (Cao et al., 2026).

The distinctive contribution of the **Hierarchical Sparse Autoencoder (HSAE)** is the combination of a dynamically reassigned multi-level partial forest, an explicit ℓ_2 parent–children constraint, and

random substitution of eligible non-root parent contributions by their children during reconstruction. Unlike implicit nesting or a fixed assignment, the edge set adapts as the feature representations evolve; unlike a full tree, the bottom-ranked 20% of children by maximum parent similarity at each adjacent level pair remain unassigned. This combination learns an adaptive cross-resolution index while preserving independently evaluable SAE levels and high-quality sparse features.

Finally, semantic hierarchy requires more than structural agreement. Grandien et al. propose generalization–specialization criteria on visual data (Grandien et al., 2026). Our activation metrics test consistency with the training constraint, while the directional judge in Section 4.3 tests whether the designated parent semantically subsumes its child.

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.

Operational meaning of an edge. HSAE directly optimizes *functional substitution*: for an assigned group, the summed decoder-space contribution of the children should approximate that of the parent. This is distinct from *activation-support containment*, where a child rarely activates without its parent, and from *directional conceptual subsumption*, where the child's meaning is a specialization of the parent's. We measure the first property through the training objective and evaluate activation agreement related to the second property. The third property is an interpretation of some learned edges, not a guarantee of the objective.

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

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

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

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

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 $\mathcal{L}_{\text{SAE},\ell}$, features that themselves have no parent are left unchanged. This includes every root and every unassigned feature. Every remaining feature is drawn independently with probability r. If none of the drawn features has children, the draw is discarded. Otherwise every drawn feature is removed from that level's reconstruction: a feature with children is replaced by the sum of those children, and a feature with no children is turned off. Consequently, root-to-second-level relations receive the parent–children constraint but not direct substitution pressure; substitution applies only to non-root features that also have children.

Formally, let $z_{\ell,i} \sim \text{Bernoulli}(r)$ for features that have a parent, and set $z_{\ell,i} = 0$ for features that do not. Let $\mathcal{S}$ be the set of features with $z_{\ell,i} = 1$. If no feature in $\mathcal{S}$ has a nonempty child set, then $\tilde{f}_{\ell,i}(x) = f_{\ell,i}(x)$ for every i. Otherwise

$$\tilde{f}_{\ell,i}(x) = \begin{cases} \sum_{j\in\mathcal{C}_{\ell,i}} f_{\ell+1,j}(x), & z_{\ell,i}=1,\ \mathcal{C}_{\ell,i}\neq\varnothing, \\ 0, & z_{\ell,i}=1,\ \mathcal{C}_{\ell,i}=\varnothing, \\ f_{\ell,i}(x), & z_{\ell,i}=0. \end{cases}$$

Turning off a childless feature is feature dropout on that feature. The parent–child loss is still zero when $\mathcal{C}_{\ell,i} = \varnothing$, so dropout comes only from the perturbation. The SAE loss $\mathcal{L}_{\text{SAE},\ell}$ is computed from the perturbed reconstruction

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

The random variables $z_{\ell,i}$ are sampled independently across features at each training step. For eligible non-root parent features, this perturbation allows their children to account for interactions with other features in the same reconstruction, which are not captured by the parent–children 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 & 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. Each hierarchy update recomputes parent–child assignments from scratch. There is no hysteresis: a child may change parent at every update. For the main results, similarity is the cosine similarity of encoder vectors. Each feature at level $\ell + 1$ is assigned to the parent at level ℓ with the largest similarity. Ties take the smallest parent index, the convention of argmax.

The partial tree is a quantile filter on that same population. Within each adjacent pair of levels, we rank child features by their maximum similarity to any parent and leave the bottom 20% unassigned. Thus, 80% of children receive a parent by construction at each adjacent level pair. The fraction of parents with no children is data-dependent and is not reported in the current experiments. Unassigned children are excluded from $\mathcal{L}_{\mathrm{PC}}$ and from the perturbation sum.

We also compare decoder cosine similarity and an EMA of activation-support Jaccard similarity (decay 0.99). All three choices perform similarly, so we use encoder similarity for its low overhead; results are in Appendix B.

4 Experiments

In this section, we evaluate whether HSAE learns parent–child assignments that are consistent with its training constraint and how the resulting features perform on standard SAE-quality metrics.

Baselines. We construct post-hoc hierarchies from independently trained JumpReLU SAEs (JR-SAEs) (Rajamanoharan et al., 2024b) and Matryoshka SAEs (MSAEs) (Bussmann et al., 2025), matching HSAE's dictionary sizes and using the same assignment procedure. We also provide framework-controlled JumpReLU reimplementations of H-SAE (Muchane et al., 2025) and Tree SAE (Cao et al., 2026), rather than exact reproductions of their native TopK implementations. These controls compare hierarchy designs under the same activation function, training activations, target sparsity, and evaluation pipeline. They should not be read as reproductions of the originally published TopK results. Tree SAE is matched in dictionary size and sparsity. H-SAE retains its coupled 2K parent and 16K supplementary dictionaries and is evaluated jointly. A four-level H-SAE extension performed substantially worse, so we report its native two-level design.

Experimental setup. Unless otherwise specified, all models discussed in this section are trained on 100M activations extracted from the layer-13 residual stream of `google/gemma-2-2b` Riviere et al. (2024) as it processes the mini-PILE dataset Kaddour (2023). HSAE, the post-hoc baselines, and Tree SAE use four levels with dictionary sizes of 2048, 4096, 8192, and 16384; H-SAE instead retains its coupled 2K–16K two-level architecture. We target a sparsity of $L_0 = 50$. Detailed experimental settings are provided in Appendix A. Appendix F reports the same comparisons at other layers, one additional model, and two other target sparsities.

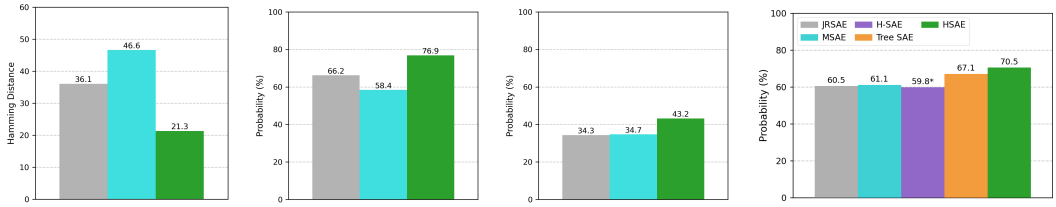
4.1 SAE feature quality

We evaluate each HSAE level as an individual SAE using variance explained and the SAEBench protocol Karvonen et al. (2025). All independently evaluable methods use a 16K dictionary, and sparsity is closely matched at $L_0 \approx 50$. Evaluation details, together with HSAE and MSAE results at every dictionary size, are provided in Appendix E.

As shown in Table 1, HSAE leads RAVEL, SCR, and Sparse Probing, outperforming both framework-controlled hierarchical baselines while retaining higher variance explained than H-SAE

Model	L_0	Var. Exp.	Absorption	AutoInterp	RAVEL	SCR	Sparse Probing
Hierarchical SAE	50.6	0.710	0.922	0.869	<u>0.661</u>	<u>0.344</u>	<u>0.893</u>
JumpReLU SAE	50.7	<u>0.712</u>	0.844	0.861	0.624	0.324	0.855
Matryoshka SAE	50.1	0.668	0.986	<u>0.882</u>	0.636	0.338	0.873
H-SAE	50.6	0.651	<u>0.988</u>	0.836	0.612	0.311	0.865
Tree SAE	50.2	0.708	0.936	0.874	0.618	0.331	0.869

Table 1: SAE-quality results with a 16K evaluated dictionary or supplementary layer. Higher is better for all metrics except L_0; the best result in each quality metric is <u>underlined</u>.



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

Figure 2: **Hierarchy evaluation.** **(a)** Logical-OR parent–children disagreement; **(b–c)** pairwise conditional co-activation; **(d)** directional semantic-subsumption judgments.

and Tree SAE. Against JumpReLU, it improves the reported Absorption and AutoInterp scores while retaining nearly identical variance explained. Thus, although HSAE does not lead every metric, its additional hierarchy objective preserves reconstruction and improves downstream feature utility in this setting.

4.2 Statistical metrics

To quantitatively evaluate the structural faithfulness of the discovered hierarchy, we employ three metrics that measure the alignment between parent and child features. To avoid overfitting, all metrics are evaluated on held-out test splits. We do not include H-SAE or Tree SAE in these activation-consistency comparisons because their hard routing constraints directly determine key outcomes: in both architectures, a child can activate only when its parent is active, making $P(\mathrm{P} \mid \mathrm{C}) = 1$ by construction; when H-SAE routing guarantees an active child for each active parent, its group-level logical-OR disagreement is also zero by construction. Reporting these quantities side by side would therefore conflate architectural guarantees with learned statistical consistency.

Hierarchical consistency via logical-OR. For each parent with a nonempty child set, and for each held-out activation x, let $P(x) = \mathbf{1}\{\sigma(e_p^\top x) > 0\}$ and let $C_\vee(x)$ be the logical OR of the same indicator over its assigned children. Unassigned children are excluded, and parents with no children are omitted. The Hamming distance in Figure 2a, the ablations, and Table 5 is the mean, over held-out activations, of the number of parents for which $P(x) \neq C_\vee(x)$. Adjacent levels are pooled into that count. The distance is not divided by the number of parents or by dictionary size. A lower value means fewer bit disagreements with the logical-OR summary. It does not, by itself, show that the child is a semantic specialization of the parent.

Conditional co-activation probabilities. We also report two pairwise conditional probabilities, macro-averaged over assigned parent–child pairs:

- **Parent-given-child**: $P(\mathrm{P} \mid \mathrm{C}) = P(\sigma(e_p^T x) > 0 \mid \sigma(e_c^T x) > 0)$. This is high when a child rarely fires without its assigned parent.
- **Child-given-parent**: $P(\mathrm{C} \mid \mathrm{P}) = P(\sigma(e_c^T x) > 0 \mid \sigma(e_p^T x) > 0)$. This measures how often an individual child is active when its parent is active, averaged over assigned edges. It should not be interpreted as the fraction of a parent's activations covered by its full set of children.

Results. Figure 2 shows lower Hamming distance and higher pairwise conditional probabilities for HSAE than for the post-hoc baselines. These gaps are consistent with the parent–child constraint, which pushes a parent's decoder contribution toward the sum of its children and therefore encourages their activations to agree. They are not an independent test that the edges recover a pre-existing semantic hierarchy. MSAE aligns less well than JRSAE on these metrics. That is expected if MSAE reduces the feature splitting that the post-hoc matching uses.

4.3 Interpretability of hierarchies

Case studies. Figure 1 illustrates a coarse-to-fine branch that progresses from a broad science-related feature to scientific disciplines and then to more specific lexical and disciplinary patterns. Appendix D presents additional branches concerning temporal, financial, and language-related concepts, showing how a coarse feature can separate into functionally distinct descendants across successive levels.

We also inspect the activation geometry associated with these relations using UMAP. Activations associated with sibling features are compared with those of an unrelated feature from a distant branch; the appendix presents representative projections, while the same separation pattern is observed across all sampled groups in this analysis. Together, these examples provide an empirical check that the learned edges can support coherent coarse-to-fine readings. They do not, however, establish that every edge in the forest is a semantic specialization.

Large-scale hierarchy judgment. To move beyond the selected case studies, we test directional semantic subsumption at scale. We show an LLM judge the activating examples of each feature, presenting the designated parent as Neuron A and its assigned child as Neuron B. The judge answers whether Neuron A semantically subsumes Neuron B—that is, whether B's concept is a subset or more specific version of A's concept. The prompt is reproduced in Appendix C. We independently sample 5000 parent–child pairs from each method for the automatic score. Sampling is not matched across methods by hierarchy level, activation frequency, branch size, or filtering status.

Results. Figure 2d is the fraction of assigned edges for which the judge determines that the designated parent semantically subsumes the child. HSAE scores 70.5%, compared with 67.1% for Tree SAE and 60.5% and 61.1% for the JumpReLU and Matryoshka post-hoc hierarchies, respectively. H-SAE obtains 59.8%; we mark this result with an asterisk because its native two-level hierarchy differs from the four-level hierarchies used by the other methods, making the score not directly comparable. On 500 pairs labeled with the same directional criterion, agreement between the automatic and human judgments is Cohen's $\kappa = 0.7233$, indicating substantial agreement and validating the direction-sensitive judge more directly than the activation metrics.

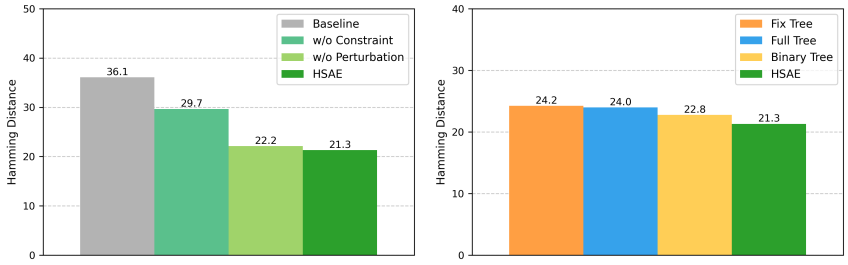
4.4 Ablation studies

We ablate both the mechanisms used to align parent and child features and the structural assumptions imposed on the hierarchy. All variants are evaluated using the logical-OR Hamming distance defined above, with lower values indicating stronger activation consistency.

Training mechanisms. Figure 3a reports Hamming distances of 36.1 without either hierarchy mechanism, 29.7 with perturbation alone, 22.2 with the constraint alone, and 21.3 for full HSAE. The constraint supplies most of the alignment gain, while perturbation provides a complementary 0.9-point improvement.

Hierarchy structure. Figure 3b shows that dynamic partial HSAE (21.3) improves over Fix Hierarchy (24.2), Full Tree (24.0), and Binary Tree (22.8), supporting dynamic reassignment, selective exclusion, and flexible branching. Fix Hierarchy isolates H-SAE's fixed-assignment choice within our pipeline rather than reproducing its complete architecture.

Hierarchy-update similarity has little effect: co-activation, decoder, and encoder variants score 21.4, 21.3, and 21.3, respectively. We use encoder similarity for its lower overhead; additional hyperparameter results are in Appendix B.



(a) Training mechanisms (b) Tree structures

Figure 3: **Ablations of HSAE design choices.** **(a)** Training mechanisms; **(b)** assignment and branching constraints. Lower is better.

5 Applications

The learned hierarchy provides direct access to features at different semantic resolutions. Parent features can supply reusable context for interpreting their descendants, while coarse decoder directions offer interventions on broad concepts that a flat, fine-grained dictionary may distribute across multiple features. We study these two uses through hierarchy-guided interpretation and concept steering; full prompts and results are in Appendix G.

5.1 Hierarchy-guided feature interpretation

Standard automated SAE interpretation describes each feature independently from its activating examples; HSAE can instead condition a child's interpretation on a previously generated parent description. Across 1,000 parent–child pairs, using 15 child examples alone scores 0.8505, while using a parent description generated from 15 parent examples and only 3 child examples scores 0.8518. Hierarchical context therefore reduces the required child-specific evidence without lowering the automated score. This automated result does not measure human annotation effort or error propagation from an incorrect parent description.

5.2 Coarse-grained concept steering

We probe 16 broad domains using target tokens and neutral prefixes. At the final token position of the layer-13 residual stream of `google/gemma-2-2b`, we apply $h \leftarrow h + 0.1 \, \|h\|_2 \, d_f$. For each method and concept, we retain ten candidates with matching AutoInterp labels and report the largest normalized target-logit increase, averaged over the same prefixes.

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

Table 2: Mean normalized target-token logit increase after selecting the strongest of ten candidates on the same prefixes. Higher is a larger in-sample shift, not a generation-level success rate.

HSAE produces the largest target-logit increase under the same candidate protocol. The maxima remain in-sample and the probe does not evaluate generated text, so this is a promising use case rather than evidence of generation-level controllability.

6 Limitations

HSAE's fixed-depth partial tree cannot represent variable abstraction depths, overlapping parents, or non-hierarchical geometry, and residual disagreements may reflect either optimization failures or the activation structure itself. Its applications remain automated or in-sample rather than human- or generation-level evaluations. Future work should explore more flexible structures, larger independent human studies, and generation-level tests.

REFERENCES

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

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. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=d4dpOCqybL.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=m25T5rAy43.

Tue M. Cao, Hoang X. Nhat, Raed Alharbi, Phi Le Nguyen, and My T. Thai. Tree SAE: Learning hierarchical feature structures in sparse autoencoders. *arXiv preprint arXiv:2605.07922*, 2026.

David Chanin, James Wilken-Smith, Tom'avs Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *ArXiv*, abs/2409.14507, 2024. URL https://api.semanticscholar.org/CorpusID:272827216.

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

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

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

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

Seyed Arshan Dalili and Mehrdad Mahdavi. Subspace-aware sparse autoencoders for effective mechanistic interpretability. *arXiv preprint arXiv:2606.06333*, 2026.

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

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

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.

Nils Grandien, David Steinmann, Felix Friedrich, and Kristian Kersting. Do sparse autoencoders learn meaningful concept hierarchies? *arXiv preprint arXiv:2606.22994*, 2026.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jE8xbmvFin.

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

Benjamin Heinzerling and Kentaro Inui. Monotonic representation of numeric attributes in language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 175–195, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.18. URL https://aclanthology.org/2024.acl-short.18/.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. Ravel: Evaluating interpretability methods on disentangling language model representations. *ArXiv*, abs/2402.17700, 2024. URL https://api.semanticscholar.org/CorpusID:268032000.

Adam Jermyn and Adly Templeton. Ghost grads: An improvement on resampling., 2024. URL https://transformer-circuits.pub/2024/jan-update/index.html#dict-learning-resampling.

Haoran Jin, Xiting Wang, Shijie Ren, Hong Xie, and Defu Lian. C^2R: Cross-sample consistency regularization mitigates feature splitting and absorption in sparse autoencoders. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. arXiv:2606.30609.

Jean Kaddour. The minipile challenge for data-efficient language models. *ArXiv*, abs/2304.08442, 2023. URL https://api.semanticscholar.org/CorpusID:258180536.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. Measuring progress in dictionary learning for language model interpretability with board game models. 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. 83091–83118. Curran Associates, Inc., 2024. doi: 10.52202/079017-2644. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9736acf007760cc2b47948ae3cf06274-Paper-Conference.pdf.

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. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *ArXiv*, abs/2503.09532, 2025. URL https://api.semanticscholar.org/CorpusID:276937927.

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

Patrick Leask, Bart Bussmann, Michael T Pearce, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9ca9eHNrdH.

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

Cristina P Martin-Linares and Jonathan P Ling. Attribution-guided distillation of matryoshka sparse autoencoders. *arXiv preprint arXiv:2512.24975*, 2025.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013a. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff (eds.), *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, Georgia, June 2013b. Association for Computational Linguistics. URL https://aclanthology.org/N13-1090/.

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

Anish Mudide, Joshua Engels, Eric J Michaud, Max Tegmark, and Christian Schroeder de Witt. Efficient dictionary learning with switch sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=k2ZVAzVeMP.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Charles O'Neill, Christine Ye, Kartheik Iyer, and John F. Wu. Disentangling dense embeddings with sparse autoencoders. *arXiv preprint arXiv:2408.00657*, 2024. O'Neill and Ye are joint first authors. Also presented as "Sparse Autoencoders for Dense Text Embeddings Reveal Hierarchical Feature Sub-Structure" at the NeurIPS 2024 Workshop on Scientific Methods for Understanding Neural Networks.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Causal Representation Learning Workshop at NeurIPS 2023*, 2023. URL https://openreview.net/forum?id=T0PoOJg8cK.

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. *ArXiv*, abs/2407.14435, 2024b. URL https://api.semanticscholar.org/CorpusID:271298201.

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, Nino Vieillard, Piotr Stańczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, 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, Boxi Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Christoper A. Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozi'nska, D. 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-Pluci'nska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost R. van Amersfoort, Josh Gordon, Josh Lipschultz, Joshua Newlan, Junsong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, L. Sifre, Lena Heuermann, Leti cia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Gorner, 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, Nen shad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Peng chong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Marie Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Se bastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomás Kociský, 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, Daniel Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeffrey Dean, Demis Hassabis, Koray Kavukcuoglu, Clément 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. *ArXiv*, abs/2408.00118, 2024. URL https://api.semanticscholar.org/CorpusID:270843326.

Martin Wattenberg and Fernanda Viégas. Relational composition in neural networks: A survey and call to action. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. URL https://openreview.net/forum?id=zzCEiUIPk9.

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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025. URL https://api.semanticscholar.org/CorpusID:278602855.

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

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 (`google/gemma-2-2b` and `Qwen/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 is 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 our training traces, this velocity-based control reduces oscillations relative to the control method described by 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 in Figure 3a.

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.

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 change the measured alignment in this ablation, we adopt encoder similarity as the default. It is the cheapest of the three to compute during training.

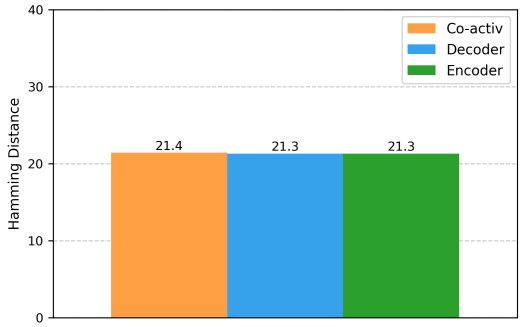
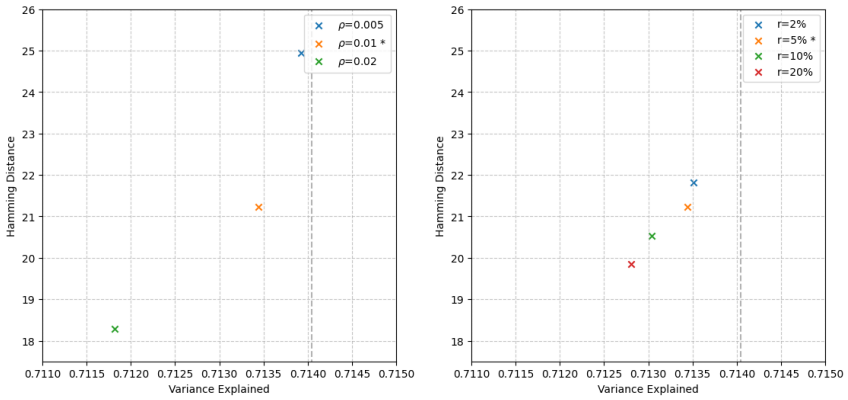


Figure 4: **Ablation of similarity metric used for hierarchy update.** (1) *Co-activ*: Use statistically derived co-activation probability as similarity; (2) *Decoder*: Use decoder vector similarity; (3) *Encoder*: Use encoder vector similarity. All three metrics provide similar results.

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 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.



(a) Parent-Child Constraint Weight ρ (b) Random Perturbation Rate r

Figure 5: **Effects of hyperparameters on the reconstruction–hierarchy-consistency trade-off.** Larger parent–child constraint weights ρ or perturbation rates r improve hierarchical alignment but eventually degrade Variance Explained.

We select the moderate setting $\rho = 0.01$ and $r = 5\%$ as an operating point on the observed reconstruction–alignment trade-off. This choice prioritizes reconstruction fidelity while retaining much of the measured alignment gain; it is not claimed to be a globally optimal setting.

B.4 TREE STRUCTURAL ASSUMPTIONS

The comparison of a dynamic hierarchy with a frozen tree, a full tree, and a binary tree is reported in Section 4.4 and Figure 3b. Freezing the assignments raises the Hamming distance from 21.3 to 24.2.

C HIERARCHY JUDGMENT DETAILS

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

Feature-pair sampling and data collection. For each evaluated method, we independently sample 5000 parent–child pairs from its hierarchy. 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.  Neuron A is the
designated parent and Neuron B is its assigned child.  Your task
is to determine whether Neuron A semantically subsumes Neuron B.
Semantic subsumption means that Neuron B's activating concept
is a subset or a more specific version of Neuron A's activating
concept.  Analyze the provided examples and output your judgment
in the following format:

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

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 whether Neuron A semantically
subsumes Neuron B.
```

Human agreement check. The prompt above explicitly tests the assigned direction by asking whether the designated parent, Neuron A, semantically subsumes its child, Neuron B. On 500 pairs, its judgments have Cohen's $\kappa = 0.7233$ agreement with human judgments made under the same directional criterion, indicating substantial agreement.

D MORE CASE STUDIES

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

D.1 EMPIRICAL EXAMPLES

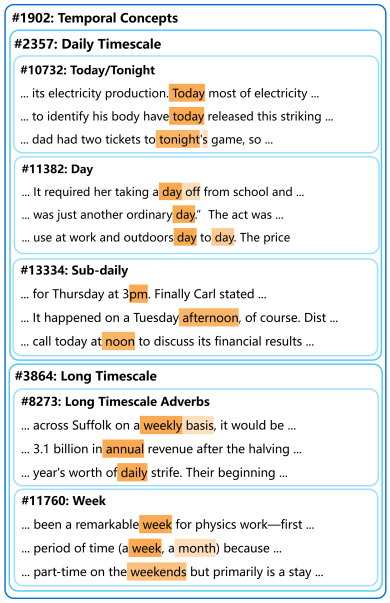


Figure 6: **Hierarchical feature discovery of temporal concepts.** The root broad temporal feature (#1902) branches into a *Daily Timescale* (#2357) and a *Longer Timescale* (#3864). At the next level, the daily timescale branch splits into *Today/Tonight* (#10732), the unit concept *Day* (#11382), and *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 6 shows a selected branch concerning *Time*. A coarse root responds to general temporal references and connects to features for daily versus longer timescales (e.g., weeks and months). The daily branch further separates immediate deictic markers (e.g., "today"), occurrences of "day", and sub-daily intervals. This branch is compatible with a coarse-to-fine semantic reading, but selected examples cannot establish how frequently that reading holds over the full forest.

Figure 7 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 capture 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 8 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).

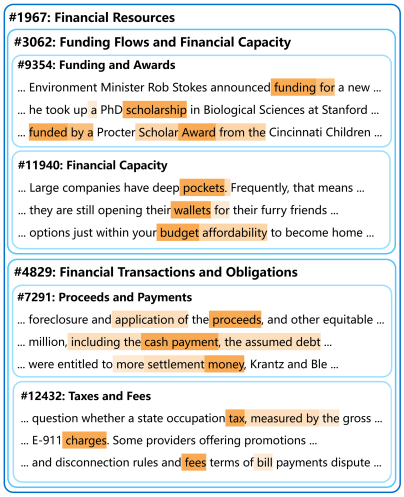


Figure 7: **Hierarchical feature discovery of financial resource concepts.** The root feature (#1967) representing financial concepts branches into *Funding Flows and Financial Capacity* (#3062) and *Financial Transactions and Obligations* (#4829). Within the *Funding Flows* branch, HSAE further distinguishes 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).

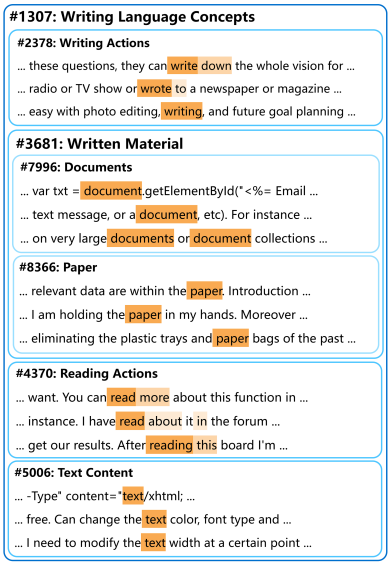


Figure 8: **Hierarchical feature discovery of writing and language concepts.** The root feature (#1307) representing broad language-related concepts branches into specialized domains: *Writing Actions* (#2378), *Written Material* (#3681), *Reading Actions* (#4370), and *Text Content* (#5006). Within the *Written Material* branch, HSAE further distinguishes between digital or structured documents (*Documents*, #7996) and physical or raw media (*Paper*, #8366).

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 9. 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.

Across all sampled feature groups in this analysis, the chosen unrelated features are isolated from the sibling groups; Figure 9 shows four representative projections. This pattern is compatible with nearby activation-space regions for features sharing a parent, although UMAP does not preserve all distances.

Sibling co-activation does not increase relative to the baseline SAEs in our measurement. This observation is inconsistent with a simple account in which the plots arise only because HSAE creates simultaneously active duplicates, although it is not by itself proof of disentanglement. The parent–child constraint does not explicitly require siblings to fire on the same tokens, and the SAE sparsity penalty discourages co-activation.

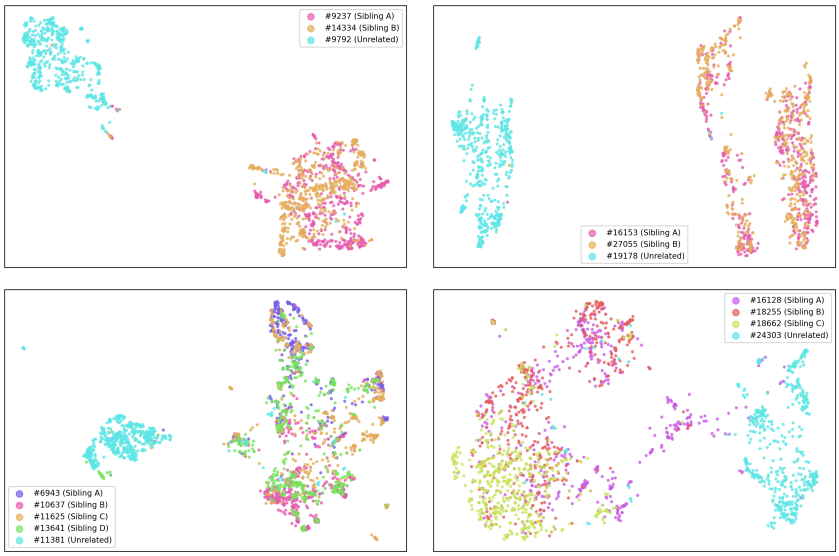


Figure 9: **Geometric manifestation of hierarchical relations.** Representative UMAP projections of activations that trigger different features. The unrelated feature is clearly separated from sibling features, as observed across all sampled groups in this analysis.

D.3 Branching factor and semantic clarity

We further examine the association between branching factor and automated interpretability. Root features with multiple children have a 2.49% higher mean AutoInterp score than roots with one child in this model; the magnitude is similar to a previously reported score change from increasing SAE dictionary size from 16K to 64K. Activation frequency and other properties may confound this descriptive correlation, so it does not show that branching causes semantic clarity. Corresponding histograms are plotted in Figure 10.

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 $(1 - \text{score})$ to align it with other metrics where a higher value indicates superior performance.

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).

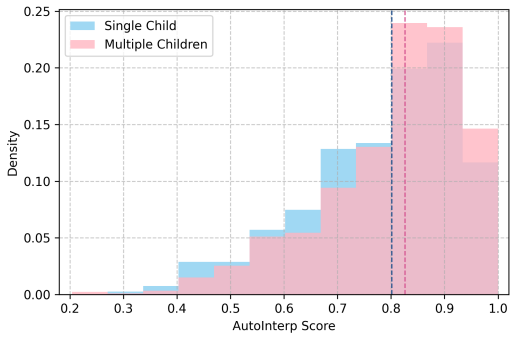


Figure 10: **Interpretability versus structural branching.** Histogram of AutoInterp scores for root features with single versus multiple children. In this model, the latter group has a 2.49% higher mean score.

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.

E.1 All dictionary sizes

We report SAEBench results across all dictionary sizes for HSAE and MSAE in Table 4. The table summarizes reconstruction and interpretability metrics under a consistent evaluation protocol.

F More evaluations

F.1 More activation sources and target LLMs

To check whether the observed pattern appears beyond the primary setting, we repeat the comparison along several dimensions. First, we move beyond the layer-13 residual stream of `google/gemma-2-2b` to train HSAE on residual-stream activations from layers 6 and 20. Second, we apply the method to layer 18 of `Qwen/Qwen3-4B` Yang et al. (2025). Finally, we train on `google/gemma-2-2b-layer-13` with target sparsities $L_0 \in \{80, 100\}$.

As summarized in Table 5, reconstruction fidelity stays close to the JumpReLU baseline in every setting we tried. Parent–child Hamming distance is lower for HSAE, and the pairwise conditional probabilities are higher. The pattern appears at layers 6, 13, and 20 of `google/gemma-2-2b`, at one layer of `Qwen/Qwen3-4B`, and at target sparsities $L_0 \in \{50, 80, 100\}$.

We note a slight discrepancy in the *Variance Explained* scores for `google/gemma-2-2b-layer-13` between Table 4 and Table 5. This shift is due to the different evaluation corpora: the benchmarks in Table 4 strictly follow the SAEBench protocol using OpenWebText, whereas the results in Table 5 are computed using the mini-PILE testing dataset. Despite this distribution shift, the reconstruction fidelity of HSAE and the baseline remains 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	0.661	0.344	0.893
Matryoshka SAE	2048	49.6	0.594	0.996	0.807	0.492	0.215	0.845
	4096	49.7	0.622	0.990	0.828	0.560	0.283	0.848
	8192	50.0	0.646	0.996	0.860	0.592	0.293	0.851
	16384	50.1	0.668	0.986	0.882	0.636	0.338	0.873

Table 4: Evaluation results for Hierarchical SAE and Matryoshka SAE at different dictionary sizes on multiple SAE benchmarks. Higher is better for all metrics except L_0.

Activation Source	Model	L_0	Var. Exp.	$P(\text{P} \mid \text{C})$	$P(\text{C} \mid \text{P})$	Ham. Dist. (↓)
`google/gemma-2-2b-layer-6`	HSAE	50.1	0.759	0.752	0.403	23.2
	JRSAE	50.0	0.759	0.651	0.331	34.6
`google/gemma-2-2b-layer-13`	HSAE	50.7	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	50.6	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
`google/gemma-2-2b-layer-20`	HSAE	50.0	0.729	0.800	0.461	19.4
	JRSAE	50.0	0.727	0.729	0.395	30.6
`Qwen/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 across multiple settings. Higher is better for all metrics except L_0 and Hamming Distance; lower is better for Hamming Distance.

G Applications

G.1 Hierarchy for feature interpretation

Experimental setup. We randomly sample 1,000 parent–child pairs from the HSAE hierarchy trained on `google/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. Table 6 presents the full set of AutoInterp scores. Several observations are noteworthy.

Providing parent context raises the automated score at the low example budget on this 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. The parent-augmented 3-example score (0.8518) closely matches the 15-example baseline without parent context (0.8505). This shows interchangeability under the same automated scorer, but not a reduction in human annotation cost.

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.

The score gap is larger on the subset selected to have a high parent AutoInterp score. 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. Because the subset and outcome use the same automated evaluation ecosystem, we treat this as suggestive rather than evidence of improved human interpretation.

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.

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 Table 7. The neutral prefix used to calculate the logit is listed in Table 8.

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

Experimental setup. Steering edits the layer-13 residual stream of `google/gemma-2-2b`, and only at the last token position. If h is that activation and d_f is the decoder direction of the selected feature, the edited activation is

$$h \leftarrow h + 0.1 \, \|h\|_2 \, d_f.$$

There is no rollout. The score is the increase in the mean logit of the target tokens after subtracting the mean logit increase over the vocabulary, averaged over the prefixes in Table 8. For each SAE and concept, we retain ten candidates whose AutoInterp label matches the concept and report the candidate with the largest score on the same prefixes and tokens. Selection and reporting are therefore not separated.

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

We compare the 4K level of HSAE with JumpReLU SAEs at 4K and 16K and with Matryoshka SAE at 16K. The 16K Matryoshka dictionary is included because its nested training is intended to retain coarse features inside a large dictionary.

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.