

INCORPORATING HIERARCHICAL SEMANTICS IN SPARSE AUTOENCODER ARCHITECTURES

Anonymous authors

Paper under double-blind review

ABSTRACT

Sparse dictionary learning (and, in particular, sparse autoencoders) attempts to learn a set of human-understandable concepts that can explain variation on an abstract space. A basic limitation of this approach is that it neither exploits nor represents the semantic relationships between the learned concepts. In this paper, we introduce a modified SAE architecture that explicitly models a semantic hierarchy of concepts. Application of this architecture to the internal representations of large language models shows both that semantic hierarchy can be learned, and that doing so improves both reconstruction and interpretability. Additionally, the architecture leads to significant improvements in computational efficiency.

1 INTRODUCTION

Dictionary learning—and, in particular, sparse autoencoders (SAEs)—have attracted significant attention as a tool for interpreting representations in large language models (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024; Lieberum et al., 2024; Lindsey et al., 2025b). The aim of these methods is to jointly learn some set of human-understandable concepts that can explain the model’s behavior, and a map from the model’s internal representations to these concepts. Empirically, at least some of the features learned by SAEs do seem clearly semantically interpretable—for example, “Golden Gate Claude” used an SAE feature to coherently steer Claude (Templeton et al., 2024). However, despite considerable effort, these models have some strong limitations. In particular, the learned features generally do not suffice to accurately reconstruct the input—limiting the value for interpretability—and, as the model size increases, the features that are learned often seem semantically unnatural. For example, Chanin et al. (2024) find that increasing the model size leads to a phenomenon they call “feature splitting”—where a single high-level concept is represented by multiple low-level features. The underlying tension here is that pushing for accurate reconstruction leads us to increase the number of features, but increasing the number of features can lead to very specialized features that are not generally useful.

The goal of this paper is to improve this reconstruction-interpretability frontier by exploiting the hierarchical structure of semantics. The motivating observation is that concepts that are meaningful to humans are often organized in a hierarchical fashion—for example, corgi, greyhound, and shitzu are all particular instances of the general concept of dog. Standard dictionary learning will not make use of this hierarchical structure at all (instead, representing each feature individually). Intuitively, we would like to modify the data structure such that this hierarchy is explicitly represented. The hope is that this will allow us to capture the reconstruction power of large dictionaries while maintaining interpretability by appealing to the human-understandable structure of the hierarchy.

The main contribution of this paper is a new architecture for SAEs that explicitly, and highly efficiently, models hierarchical relationships between concepts. Concretely:

1. Park et al. (2024a) give foundational results on how hierarchical structure is represented in language models. We show how to translate this theory into a mixture-of-experts type architecture (see Figure 2) that captures hierarchical structure (see Figure 1).
2. We then show that this architecture strongly improves the reconstruction performance of SAEs, while maintaining or improving interpretability.



077

078

079

080

081

082

083

084

085

086

087

088

089

090

091

092

093

094

095

096

097

098

099

100

101

102

103

104

105

106

107

Figure 1: A Hierarchical Sparse Autoencoder architecture learns human interpretable hierarchy. Each box shows 5 of the strongest activating contexts for the feature, all tokens underlined activate the feature while the bold/text weight shows the relative strength of activation. For example, a “marriage” high-level feature and “divorce”, “engagement”, and “marriage” sublatents. Note that “48 sati svadba” is a Serbian wedding reality TV show and “Shaadi” is the Hindi word for marriage.

As an additional benefit, the new architecture is highly computationally efficient. In principle, this can allow scaling SAEs to much larger effective dictionary sizes, allowing fine-grained representations of a vast number of concepts.

2 BACKGROUND AND RELATED WORK

Sparse Autoencoders SAEs are a particular approach to sparse dictionary learning, a general class of unsupervised learning algorithms that aim to find a dictionary of human-interpretable features (or atoms) that can be used to reconstruct the input data. The hope is that by enforcing sparsity we can recover some set of underlying latent factors. There are a large number of methods based on this idea.

We’ll focus on top-k SAE approach of Gao et al. (2024), which has the advantages of being simple and scalable. The idea is to map each input vector x to a latent representation that is sparsified by a TopK operation that preserves the k largest values and sets all others to zero. Then, the vector is reconstructed by decoding this sparse latent representation. In total, the SAE operation is given by:

$$\text{SAE}_k(\mathbf{x}) = \mathbf{D}\text{TopK}_k(\text{LeakyReLU}_\alpha(\mathbf{E}(\mathbf{x} - \mathbf{b}))) \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the input, \mathbf{E} is the encoder matrix, \mathbf{D} is the decoder matrix, \mathbf{b} is a bias vector (with the rows, columns, and dimension of \mathbf{E} , \mathbf{D} , and \mathbf{b} respectively being equal to the number of features) k is a hyperparameter, and $\alpha = \frac{1}{\sqrt{d}}$ is an adaptive threshold for the LeakyReLU activation (where LeakyReLU has some small negative slope below the threshold). The vectors in \mathbf{D} are referred to as “features” or “latent vectors.” The parameters are learned by minimizing the reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \|\mathbf{x} - \text{SAE}_k(\mathbf{x})\|_2^2 \quad (2)$$

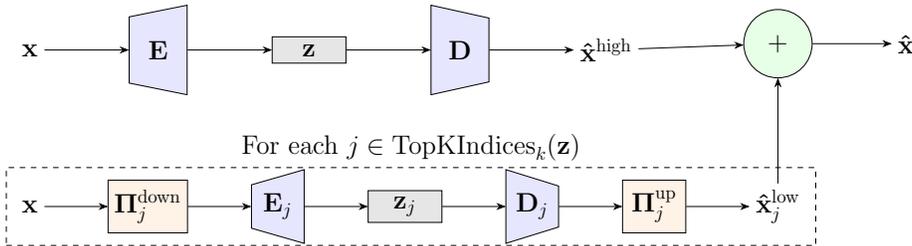


Figure 2: The structure of the Hierarchical Sparse Autoencoder architecture. The design combines a top-level encoder-decoder (upper path) that captures general concepts with expert-specific autoencoders (lower paths) that model refined features. For a given input, only a sparse subset of experts is activated, enhancing computational efficiency while maintaining expressive power.

Related Work The use of sparse autoencoders in LLM interpretability has been a topic of considerable interest (Marks et al., 2025), (Lindsey et al., 2025a), (Engels et al., 2024), (Kissane et al., 2024).

The most closely related is a line of work focused on ameliorating feature splitting. For instance, Matryoshka SAEs (Bussmann et al., 2025) and EWG-SAEs (Li & Ren, 2025) both operate by defining groupings of dictionary atoms and then modifying the training objective of the SAE to encourage some of the groupings to contain coarser features, and other groupings to contain finer features. These approaches are also motivated by the hierarchical nature of semantics. In contrast to the approach we take here, they do not modify the flat set structure of the architecture. Modifying the architecture has the advantages that it both makes the hierarchical structure explicit—improving interpretability—and also allows us to leverage the hierarchical structure to greatly improve computational efficiency, as we will see.

We also highlight Switch SAEs (Mudide et al., 2024), which introduce a mixture-of-experts type routing mechanism shares some structural similarities to our approach. However, this approach is entirely motivated by pushing the reconstruction quality vs sparsity frontier for a given compute budget. Each expert has no high vs low-level distinction and isn’t designed to contain a set of features that should be interpreted together. Indeed, they find that their decoder features do not cluster in any particular way.

The ideas here connect to the broader field of causal representation learning (CRL), which aims to learn the causal factors underlying a data-generating process (Locatello et al., 2019; 2020; Schölkopf et al., 2021; Ahuja et al., 2022a; Brehmer et al., 2022; Lippe et al., 2023; Moran & Aragam, 2025). Recent work has extended CRL to foundation model analysis, seeking identifiable and disentangled representations of concepts (Rajendran et al., 2024; Joshi et al., 2025). In this spirit, we also aim to learn disentangled concept representations.

3 HIERARCHICAL SPARSE AUTOENCODER ARCHITECTURE

We now turn to the development of an architecture that bakes in the hierarchical structure of concepts.

3.1 ARCHITECTURE

Hierarchical Geometry Our main inspiration follows Park et al. (2024a), who find that the representations of categorical concepts in language models have a specific geometric structure. In particular, they show that every categorical concept has *two* representations: a parent feature indicating whether the concept is active, and a low-rank subspace of the representation space containing a polytope where each point is a child feature corresponding to a different possible value of the concept. For example, the concept “dog” is represented by a parent feature indicating whether the concept is active (‘is dog’ vs ‘not dog’), and a low-dimensional subspace containing a polytope where each point is a child feature corresponding to a different breed of dog.

The idea is to create an architecture that matches this dual representation structure; see Figure 2. To achieve this, we propose an architecture with three parts:

1. A top-level SAE that aims to capture the binary feature indicating whether a high-level concept is active. This is a standard SAE with a relatively small number of features.
2. For each atom in the high-level SAE, we have an associated down projector onto a low-dimensional subspace (and an up-projector mapping back to the original space). This corresponds to the low-dimensional subspace associated to the categorical concept. Since each high-level concept spans a low-dimensional subspace, the only requirement for the ‘projection’ is staying within the column space of the concept. Thus, we do not impose idempotence or symmetry and simply use 2 trainable matrices.
3. For each atom in the high-level SAE, we have a low-level SAE that operates on the low-dimensional subspace associated with atom. The elements of each such SAE correspond to the different possible values of the categorical concept (e.g., the different breeds of dog).

That is, the architecture is a high-level SAE where each atom also has a low-level SAE attached to it.

Now, a critical observation here is that an atom in the low-level SAE can only be active if the high-level atom is also active. That is, for the residual stream vector x to encode the concept of “corgi” it *must* also encode the concept of “dog.” To respect this constraint, we structure the model as a mixture-of-experts type architecture, where a low-level SAE is only called if the associated high-level feature is active. This both respects the hierarchical structure of the concepts and also allows for an enormous computational efficiency gain (see below).

In fact, the Park et al. (2024a) results are more precise than the informal presentation above. In particular, they show that a low level concept (“corgi”) is naturally represented as the *sum* of a vector for the high-level concept (“dog”) and a vector representing the low-level concept in the context of the high level space (i.e., “corgi” = “dog” + “corgi | dog”). It is these contextual representations that live in a low-dimensional space. This overall structure—a low level concept is represented as the sum of a high-level concept plus the low-level concept in the context of the high-level one—is exactly the structure implemented by the H-SAE.

Forward Pass We can now make the architecture precise:

$$\text{H-SAE}(\mathbf{x}) = \sum_{j \in \text{TopKIndices}_k} \left(z_j \mathbf{d}_j + \underbrace{\prod_j^{\text{up}} \text{SAE}_1^j \left(\prod_j^{\text{down}} \mathbf{x} \right)}_{\hat{\mathbf{x}}_j^{\text{low}}} \right) \quad (3)$$

where TopKIndices_k are the indices of the top k features, \mathbf{d}_j is the j -th high-level feature, $z_j = (\text{LeakyReLU}(\mathbf{E}\mathbf{x}))_j$ is the corresponding code, SAE_1^j is the expert-specific autoencoder for high-level feature j , and \prod_j^{up} and \prod_j^{down} are projection matrices (of dimension s) that map the input to the expert subspace and back to the original space, respectively. Note that the low-level SAE uses TopK_1 over its a features, retaining only a single low-level feature for each expert. This corresponds to the idea that the representation of a subordinate concept should be at a particular vertex in the polytope corresponding to the categorical concept.

We depict this architecture visually in Figure 2 and algorithmically in Algorithm 1.

Computational Efficiency Beyond explicitly enforcing the target semantic structure, activating the low-level SAE only when the corresponding high-level feature is active provides a significant computational advantage. The computational cost of a forward pass can be expressed as follows:

$$\begin{aligned} \text{Cost}_{\text{forward}} = & \underbrace{O(jd)}_{\text{High-level encoding}} + \underbrace{O(ksd)}_{\text{Subspace projection}} + \underbrace{O(kas)}_{\text{Low-level encoding}} \\ & + \underbrace{O(kas)}_{\text{Low-level decoding}} + \underbrace{O(ksd)}_{\text{Upward projection}} + \underbrace{O(kd)}_{\text{High-level decoding}} \end{aligned}$$

where j is the number of high-level features (experts), d is the dimensionality of the input activation vector, s is the subspace dimension for each expert, a is the number of low-level features per expert, and k is the sparsity level (number of activated experts). In the typical case where $k \ll j$ (sparsity) and $s \ll d$ (low dimensional subspace) the cost is dominated by the top-level SAE. That is, equipping

Algorithm 1 Hierarchical SAE Forward Pass and Loss Computation

```

216 function ForwardPass( $\mathbf{x}$ )
217
218    $\mathbf{z} \leftarrow \text{TopK}_k(\text{LeakyReLU}_\alpha(\mathbf{E}\mathbf{x}))$  ▷ High-level encoding
219    $\mathcal{K} \leftarrow$  indices of nonzero elements in  $\mathbf{z}$ 
220    $\hat{\mathbf{x}}^{\text{high}} \leftarrow \mathbf{D}\mathbf{z}$  ▷ High-level reconstruction
221    $\hat{\mathbf{x}}^{\text{low}} \leftarrow \mathbf{0}$  ▷ Initialize low-level reconstruction
222   for  $j \in \mathcal{K}$  do ▷ Only process activated experts
223      $\mathbf{x}_j^{\text{sub}} \leftarrow \mathbf{\Pi}_j^{\text{down}}\mathbf{x}$  ▷ Project to expert subspace
224      $\mathbf{z}_j \leftarrow \text{LeakyReLU}_\alpha(\mathbf{E}_j\mathbf{x}_j^{\text{sub}})$  ▷ Low-level encoding
225      $\hat{\mathbf{x}}_j^{\text{sub}} \leftarrow \mathbf{D}_j\mathbf{z}_j$  ▷ Reconstruct in subspace
226      $\hat{\mathbf{x}}^{\text{low}} \leftarrow \hat{\mathbf{x}}^{\text{low}} + \mathbf{\Pi}_j^{\text{up}}\hat{\mathbf{x}}_j^{\text{sub}}$  ▷ Project back and accumulate
227   end for
228    $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}^{\text{high}} + \hat{\mathbf{x}}^{\text{low}}$  ▷ Combined reconstruction
229   return  $\hat{\mathbf{x}}, \mathbf{z}, \{\mathbf{z}_j\}_{j \in \mathcal{K}}$ 
230 end function
231
232 function ComputeLoss( $\mathbf{x}, \hat{\mathbf{x}}, \mathbf{z}, \{\mathbf{z}_j\}_{j \in \mathcal{K}}$ )
233    $\mathcal{L}_{\text{recon}} \leftarrow \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$ 
234    $\mathcal{L}_{\text{top\_recon}} \leftarrow \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2$ 
235    $\mathcal{L}_{\text{recon}} \leftarrow \mathcal{L}_{\text{recon}} + \beta\mathcal{L}_{\text{top\_recon}}$  ▷ Encourage meaningful top-level features
236    $\mathcal{L}_{\text{sparse}} \leftarrow \|\mathbf{z}\|_1 + \sum_{j \in \mathcal{K}} \|\mathbf{z}_j\|_1$ 
237    $\mathcal{L}_{\text{ortho}} \leftarrow \frac{\|\mathbf{D}\mathbf{E} - \text{diag}(\mathbf{D}\mathbf{E})\|_F}{n^2 - n}$ 
238    $\mathcal{L} \leftarrow \mathcal{L}_{\text{recon}} + \lambda_1\mathcal{L}_{\text{ortho}} + \lambda_2\mathcal{L}_{\text{sparse}}$ 
239   return  $\mathcal{L}$ 
240 end function

```

the top-level SAE with hierarchical structure adds negligible computational overhead. Because the hierarchical SAE is much more expressive, this significantly improves SAE scalability. We also note that because the memory cost of a batch gradient step scales with the number of activated parameters, not the total number of parameters, this efficiency also applies to (per-step) training. Indeed, in practice we find that the additional cost imposed by the hierarchical structure is very small.

3.2 TRAINING OBJECTIVE

We also make some minor modifications to the training objective, using the following loss function:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_1\mathcal{L}_{\text{ortho}} + \lambda_2\mathcal{L}_{\text{sparse}} \quad (4)$$

Reconstruction Loss Following standard practice, we measure reconstruction loss with Euclidean distance. Additionally, to encourage the top-level SAE features to be meaningful in their own right (rather than just as routers) we also penalize reconstruction error from the top-level SAE only. The reconstruction loss is then:

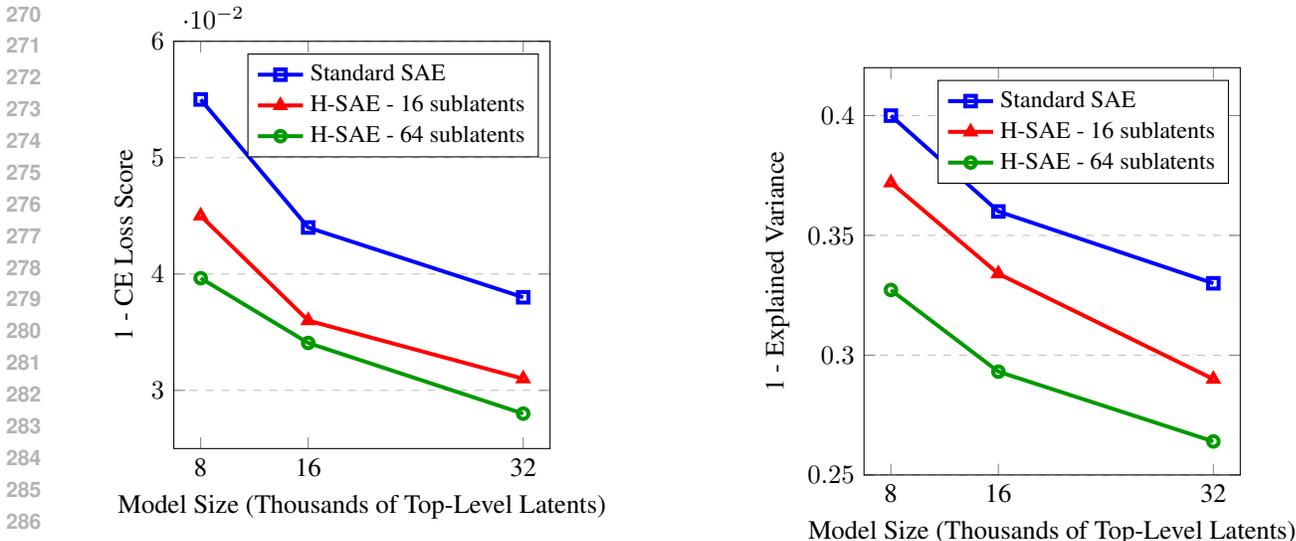
$$\mathcal{L}_{\text{recon}} = \|\mathbf{x} - \text{H-SAE}(\mathbf{x})\|_2^2 + \beta\|\mathbf{x} - \hat{\mathbf{x}}^{\text{high}}\|_2^2, \quad (5)$$

where $\beta = 0.1$ was chosen because the top-level SAE alone has less capacity than the full H-SAE.

Auxiliary Losses Park et al. (2024b) show that the representations of “causally separable” (individually manipulable) features are bi-orthogonal in a certain sense.¹ This suggests that we could discourage semantic redundancy in the learned features (e.g., having multiple features for “dog”) by imposing a suitable orthogonality constraint on the learned features. To operationalize this, we introduce a bi-orthogonality penalty on the top-level features:

$$\mathcal{L}_{\text{ortho}} = \frac{\|\mathbf{D}\mathbf{E} - \text{diag}(\mathbf{D}\mathbf{E})\|_F}{n^2 - n}, \quad (6)$$

¹Namely, the dot product between primal and dual space representations is zero.



(a) The H-SAE has better reconstruction performance than a standard SAE as measured 1 - CE Loss score across different model sizes and with both 16 and 64 sublattents per expert. Lower values indicate better downstream model performance, using the SAEbench CE loss score.

(b) The H-SAE has better reconstruction performance than a standard SAE as measured 1 - Explained Variance across different model sizes and with both 16 and 64 sublattents per expert. Lower values indicate better capture of the data’s inherent structure.

Figure 3: H-SAEs have better reconstruction performance as measured by explained variance and the downstream CE loss of a Gemma 2-2B using SAE-reconstructed activation vectors.

where $\text{diag}(\mathbf{DE})$ is the diagonal matrix formed by the diagonal elements of the top-level decoder multiplied by the top-level encoder, and $\|\cdot\|_F$ denotes the Frobenius norm. This pushes the encoder representation for feature i and decoder representation for feature j to be orthogonal if $i \neq j$. Mechanically, this means that if we ran the encoder on the autoencoder’s own output, whether feature z_i was active in the first encoding would not effect feature z_j in the second encoding.

The motivation here is to encourage compositionality in the learned features. However, it is not clear how to empirically measure compositionality, and so we are unsure whether this term actually achieves this goal. Nevertheless, we do observe empirically that adding this term does an excellent job of mitigating the “dead atom” phenomena where many features are never used in reconstructions. We use this instead of the auxiliary dead latent loss proposed by Gao et al. (2024), but we do not believe this is a crucial component. See Section B for ablations.

We also include a small ℓ_1 sparsity penalty on the latent values on both the top and low-level features outside the top k to further encourage specialization; this is the $\mathcal{L}_{\text{sparse}}$ term. Again, it’s unclear how to measure the target compositionality effect, and we do not believe this is a key component. See Section B for ablations.

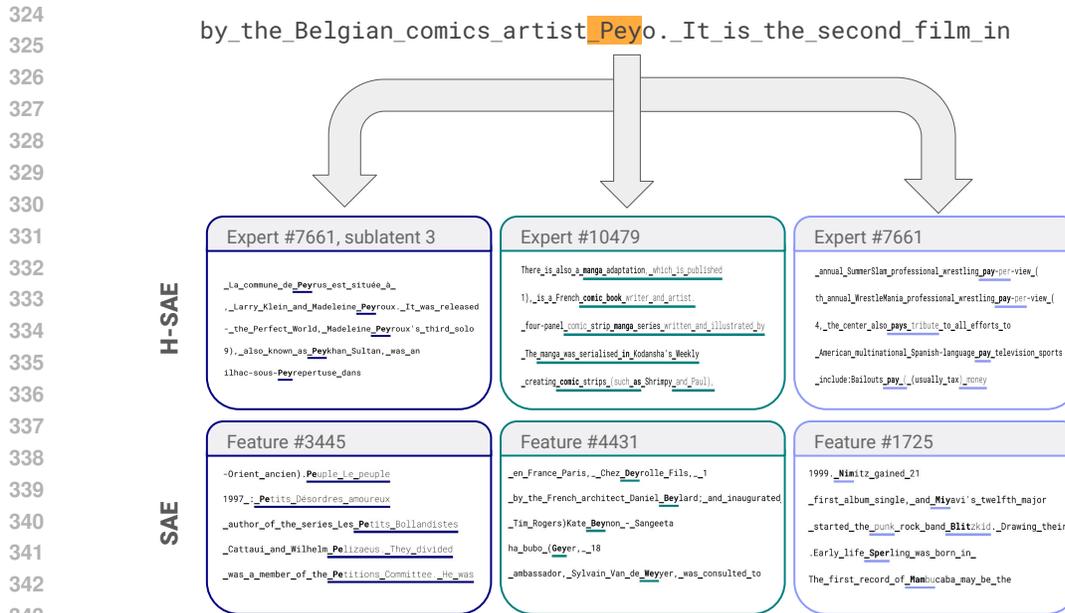
4 EXPERIMENTS

The main motivation for this work is that by incorporating the hierarchical structure of semantics we can improve the reconstruction-interpretability frontier. Accordingly, we want to answer three questions:

1. Does the H-SAE improve reconstruction?
2. Does the H-SAE maintain, or improve, interpretability?
3. Does the model in fact learn hierarchical semantics?

We will see that the answer to all three questions is yes.

Experimental Setup Our training data consists of 1 billion residual stream vectors extracted from layer 20 of Gemma 2-2B . We collect this data by running Gemma on a large corpus of Wikipedia



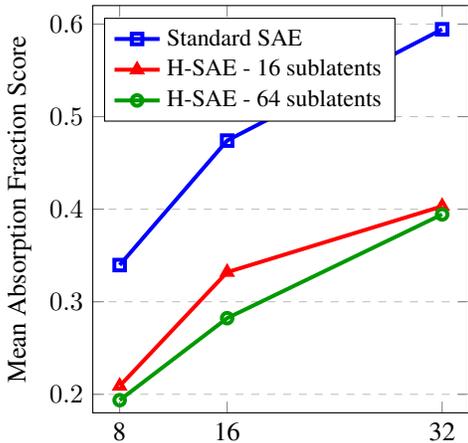
344 Figure 4: A comparison decomposition of the same context/embedding in the H-SAE architecture
 345 and standard SAE. The H-SAE represents a token about the creator of the Smurfs cartoon using
 346 a high level “Pay” homophone feature, and a low-level “Pey” token feature. This feature is then
 347 composed with a comic books feature. Whereas, the SAE has features for “words that end with ‘ey’”
 348 and “Pe” words, but none of the top-32 contain a high-level feature for the ‘comic book’ context of
 349 this sentence, rather the SAE uses a ‘part of an artists’ name feature to reconstruct the embedding.
 350

352 articles, spanning multiple languages and a wide range of topics. For each article, we extract the
 353 residual stream vectors corresponding to the first 256 tokens. Following standard practice, we
 354 normalize the vectors to unit norm (Lieberum et al., 2024). However, we do not subtract any mean
 355 vector nor include a bias term as Gao et al. (2024) do, as we found this decreased training stability.

356 As a baseline, we use the TopK SAE architecture (Gao et al., 2024). The H-SAE is trained using
 357 the objective function described in Equation (4). All models are trained on the same data for 4
 358 epochs. The baseline top-k autoencoders empirically perform equivalently on reconstruction loss to
 359 the Gemma Scope JumpReLU autoencoders (Lieberum et al., 2024). See Section A for more details
 360 on training.
 361

362 **Reconstruction** Figure 3 shows the reconstruction performance of the H-SAE and the baseline at a
 363 variety of dictionary sizes. We measure both 1 - explained variance (i.e. normalized ℓ^2 reconstruction
 364 loss) and the LLM CrossEntropy loss induced by replacing the original activations with the recon-
 365 structions (normalized by SAEbench between 0 and 1). As expected, adding hierarchical capacity to
 366 the model improves reconstruction performance. Indeed, this effect is dramatic. The H-SAE with 64
 367 sublatents per expert is on par with the standard SAE with 32 top-level features, despite having 1/4
 368 the compute cost.
 369

370 **Matryoshka SAE** Matryoshka SAEs (Bussmann et al., 2025) are the most similar SAE architecture
 371 conceptually, hoping to reduce undesirable feature absorption and increase hierarchical interpretability.
 372 To compare these models, we start by training them on the token unembeddings, as in other ablations.
 373 The features are fairly similar, with the H-SAE often finding sparser representations, finding high-
 374 level features the M-SAE does not, and fewer uninterpretable features. These factors also combine
 375 with the reconstruction loss and computational advantage the H-SAE has over both an M-SAE and
 376 standard SAE. Additionally, we train both an H-SAE and M-SAE on the synthetic benchmark from
 377 the M-SAE paper, and find the H-SAE outperforms the M-SAE. Full results and experimental details
 are in Section D.



(a) The standard SAE shows higher absorption, indicating greater feature splitting, while our H-SAE architecture maintains more coherent representations as measured by the mean fraction of first-letter classifications that show signs of absorption.

Language Pair	H-SAE	SAE
English vs French	8.448	9.772
English vs Spanish	8.190	9.598
English vs German	9.372	10.938
French vs Spanish	5.748	7.170
French vs German	7.306	9.038
Spanish vs German	7.476	9.270

(b) The H-SAE architecture has less divergence in features for the same sentence in different languages. Higher values indicate more redundant features, as measured by the mean set difference between the same token in different languages, demonstrating our H-SAE architecture’s superior ability to learn highly composable features.

Figure 5: H-SAEs exhibit less undesirable feature absorption and fewer redundant features across languages

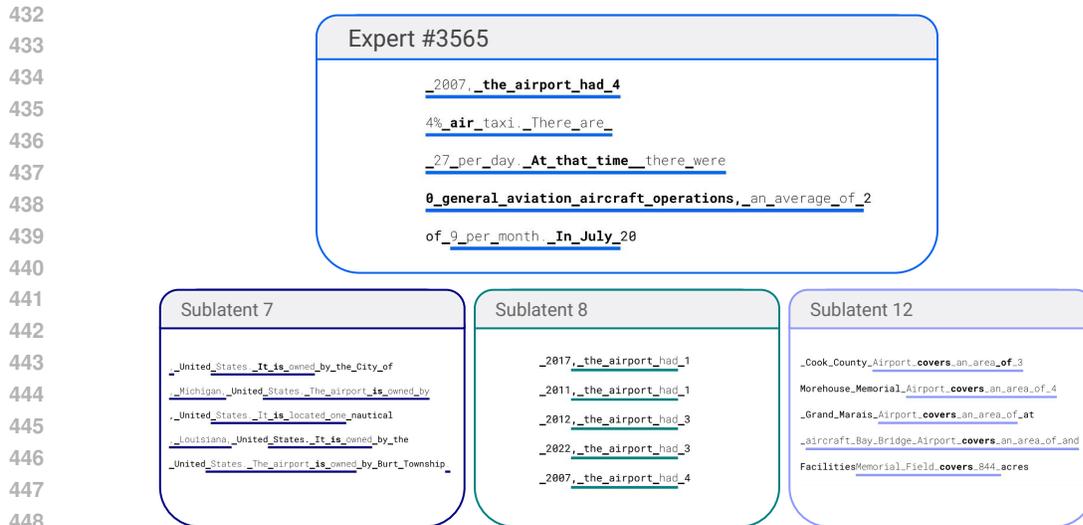
4.1 INTERPRETABILITY

By itself, an improvement in reconstruction performance may not be meaningful. The reason is that there are many possible modifications that improve reconstruction performance by sacrificing interpretability. This concern is somewhat mitigated by the fact that the extra capacity of the H-SAE is highly constrained; low-level features can only be activated when their corresponding high-level feature is active. Nevertheless, we would like to directly evaluate the interpretability of the H-SAE against a standard SAE.

Figure 4 shows a comparison of the baseline and H-SAE decomposition of the same context. We observe that the H-SAE has features that are of the same quality, if not better. This behavior is typical. See Section C for more examples of features and supplementary materials for an interactive notebook including visualizations of hundreds of features from each model.

To complement the visualizations, we perform some more systematic tests. Recent work has shown that general-purpose automated interpretability evaluations are misleading—particularly regarding more abstract features (Heap et al., 2025). Accordingly, we focus in particular on feature splitting and absorption. These are aspects that we would expect to see the greatest effect on through learning more general top-level features, and are relatively measurable. To test feature absorption, we run the first letter classification benchmark from SAEBench (Karvonen et al., 2025). This benchmark tests the improvement in probing performance for a first letter classification task as the number of features used increases above 1. This task should only require a single feature if there is no undesirable splitting or absorption. As expected, we find that the H-SAE architecture both improves performance on this task and decreases the rate of increase in absorption as width increases. See Figure 5a, where we report the fraction of the projection from SAE activations onto a first-letter classification probe that is not explained by a single feature.

We can also test for the presence of duplicate features or ‘redundancy’. One particular example of this that we and others observe is equivalent syntactic features from different languages (e.g. a French period and an English period) (Lindsey et al., 2025b). So, we sample 1,000 English sequences from the training data, consisting of between 16 and 48 tokens. We then use Gemini-2.0-Flash to translate these sentences into French, Spanish, and German (the most common non-English languages in the training data). Then, we collect the top 8 most strongly activated features on the last token of the context for the standard and H-SAE. Ideally, the features activated by the same token in different



450 Figure 6: The H-SAE has a “airports” high-level feature and “US airport”, “the token ‘airport’”, and
 451 “airport size” sublatents.

452

453

454 languages should be similar. We measure the size of the symmetric set difference between the top 8
 455 features activated by the same token in different languages (a value between 0 and 16); see Figure 5b.
 456 As expected, the H-SAE uses more similar sets of features across all language pairs than the baseline
 457 SAEs.

458

459 **Hierarchical Semantics** Finally, we turn to the question of whether the H-SAE is indeed learning
 460 hierarchical semantics. To this end, we visualize examples where low-level features are strongly
 461 activated. Figures 1, 4 and 6 show examples from the 16k experts x 16 sublatents H-SAE. We observe
 462 that hierarchical semantics are clearly emerging. See the supplementary materials for an interactive
 463 notebook including hundreds of such visualizations.

464 5 DISCUSSION

465

466

467

468 The main question in this work is whether the interpretability-reconstruction frontier can be improved
 469 by exploiting the hierarchical structure of semantics. We find that the answer is yes. Incorporat-
 470 ing hierarchy dramatically improves reconstruction and moderately improves interpretability of
 471 the top-level features. Further, the two-level structure effectively communicates semantic relation-
 472 ships between concepts—e.g., we observed high-level latents capturing regional concepts like “Bay
 473 Area,” with corresponding expert-specific sub-latents representing entities like “Stanford” that exist
 474 within that region. As a further benefit, the hierarchical structure allows for large increases in the
 475 computational efficiency relative to the effective number of atoms.

476

477 **Limitations and Further Work** In this work we primarily analyze the architectural changes in
 478 the context of a simple ‘standard’ SAE approach. Incorporating hierarchy leads to an improvement,
 479 but the results are still imperfect—e.g., we still find some level hard-to-interpret features, and we
 480 still far short of perfect reconstruction. It is unclear whether this reflects a fundamental limitation of
 481 dictionary learning, or whether there is some additional set of changes that would lead to dramatically
 482 improved performance. For example, our ablation studies on word unembeddings suggest that using
 483 a reconstruction objective other than Euclidean distance can massively improve interpretability; see
 484 Section B.3. Similarly, work in causal representation learning has shown simple sparsity objectives
 485 have fundamental limitations (Locatello et al., 2019), and developed a variety of more sophisticated
 objectives leading to much better performance (Locatello et al., 2020; Ahuja et al., 2022b; Lippe
 et al., 2022). It would be exciting to find ways to incorporate such insights into LLM interpretability.

486 REPRODUCIBILITY

487
488 All the main experiments and results are reproducible using the code and data provided in the
489 supplementary material. The full dataset used for training is too large to include directly as it is
490 10TB, but the full code used to create it is included. Model checkpoints are provided for all H-SAEs
491 and standard SAEs trained. To aid with evaluation, supplementary material also includes hundreds of
492 randomly selected features to demonstrate the consistent interpretability and quality of the learned
493 features.

494
495 ACKNOWLEDGEMENTS496
497 REFERENCES

498
499 Kartik Ahuja, Jason Hartford, and Yoshua Bengio. Weakly supervised representation learning with
500 sparse perturbations, 2022a. URL <https://arxiv.org/abs/2206.01101>.

501
502 Kartik Ahuja, Jason S. Hartford, and Yoshua Bengio. Weakly Supervised Representation Learning
503 with Sparse Perturbations. *Advances in Neural Information Processing Systems*, 35:15516–15528,
504 December 2022b. URL [https://proceedings.neurips.cc/paper_files/paper/](https://proceedings.neurips.cc/paper_files/paper/2022/hash/63d3bae2c1f525745003f679e45bcf7b-Abstract-Conference.html)
505 [2022/hash/63d3bae2c1f525745003f679e45bcf7b-Abstract-Conference.](https://proceedings.neurips.cc/paper_files/paper/2022/hash/63d3bae2c1f525745003f679e45bcf7b-Abstract-Conference.html)
506 [html](https://proceedings.neurips.cc/paper_files/paper/2022/hash/63d3bae2c1f525745003f679e45bcf7b-Abstract-Conference.html).

507
508 James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclau-
509 rin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang.
510 Jax: composable transformations of python+numpy programs. Computer software, 2018. URL
<http://github.com/jax-ml/jax>.

511
512 Johann Brehmer, Pim de Haan, Phillip Lippe, and Taco Cohen. Weakly supervised causal representa-
513 tion learning, 2022. URL <https://arxiv.org/abs/2203.16437>.

514
515 Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick
516 Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec,
517 Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina
518 Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and
519 Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary
520 features/index.html.

521
522 Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level fea-
523 tures with matryoshka sparse autoencoders, 2025. URL [https://arxiv.org/abs/2503.](https://arxiv.org/abs/2503.17547)
[17547](https://arxiv.org/abs/2503.17547).

524
525 David Chanin, Julian Wilken-Smith, Tomas Dulka, Harsh Bhatnagar, and Joseph Bloom. A is
526 for absorption: Studying feature splitting and absorption in sparse autoencoders, 2024. URL
<https://doi.org/10.48550/arXiv.2409.14507>.

527
528 Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoen-
529 coders find highly interpretable features in language models, 2023. URL [https://arxiv.](https://arxiv.org/abs/2309.08600)
530 [org/abs/2309.08600](https://arxiv.org/abs/2309.08600).

531
532 Joshua Engels, Isaac Liao, Eric J. Michaud, Wes Gurnee, and Max Tegmark. Not All Language
533 Model Features Are Linear, May 2024. URL <http://arxiv.org/abs/2405.14860>.
[arXiv:2405.14860 \[cs\]](http://arxiv.org/abs/2405.14860).

534
535 Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya
536 Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint*
537 *arXiv:2406.04093*, 2024.

538
539 Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Sparse Autoencoders Can
Interpret Randomly Initialized Transformers, January 2025. URL [http://arxiv.org/abs/](http://arxiv.org/abs/2501.17727)
[2501.17727](http://arxiv.org/abs/2501.17727). [arXiv:2501.17727 \[cs\]](http://arxiv.org/abs/2501.17727).

- 540 David D. Johnson. Penzai + treescop: A toolkit for interpreting, visualizing, and editing models as
541 data, 2024.
542
- 543 Shruti Joshi, Andrea Dittadi, Sébastien Lachapelle, and Dhanya Sridhar. Identifiable steering via
544 sparse autoencoding of multi-concept shifts, 2025. URL <https://arxiv.org/abs/2502.12179>.
545
- 546 Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau,
547 Eoin Farrell, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Arthur Conmy, Samuel
548 Marks, and Neel Nanda. Saebench: A comprehensive benchmark for sparse autoencoders in
549 language model interpretability, 2025. URL <https://doi.org/10.48550/arXiv.2503.09532>.
550
- 551 Patrick Kidger and Cristian Garcia. Equinox: Neural networks in jax via callable pytrees and filtered
552 transformations, 2021. URL <https://doi.org/10.48550/arXiv.2111.00254>.
553
- 554 Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda.
555 Interpreting Attention Layer Outputs with Sparse Autoencoders, June 2024. URL <http://arxiv.org/abs/2406.17759>. arXiv:2406.17759 [cs].
556
- 557 Ed Li and Junyu Ren. UNLOCKING HIERARCHICAL CONCEPT DISCOVERY IN LANGUAGE
558 MODELS THROUGH GEOMETRIC REGULARIZATION. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL <https://openreview.net/forum?id=i31cKXiym>.
559
560
- 561 Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant
562 Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse
563 autoencoders everywhere all at once on gemma 2, 2024. URL <https://arxiv.org/abs/2408.05147>.
564
565
- 566 Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner,
567 Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly
568 Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam
569 Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley
570 Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language
571 model. *Transformer Circuits Thread*, 2025a. URL <https://transformer-circuits.pub/2025/attribution-graphs/biology.html>.
572
- 573 Jack Lindsey, Will Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Chris
574 Citro, Daniel Abrahams, Shan Carter, Ben Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton,
575 Trenton Bricken, Callum McDougall, Hoagy Cunningham, Tom Henighan, Adam Jermyn,
576 Andy Jones, and Joshua Batson. On the biology of a large language model, 2025b. URL <https://transformer-circuits.pub/2025/attribution-graphs/biology.html>.
577
- 578 Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M. Asano, Taco Cohen, and Stratis Gavves.
579 CITRIS: Causal Identifiability from Temporal Intervened Sequences. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 13557–13603. PMLR, June 2022. URL <https://proceedings.mlr.press/v162/lippe22a.html>. ISSN: 2640-3498.
581
582
- 583 Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M. Asano, Taco Cohen, and Efstratios Gavves.
584 Biscuit: Causal representation learning from binary interactions, 2023. URL <https://arxiv.org/abs/2306.09643>.
585
- 586 Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf,
587 and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled
588 representations, 2019. URL <https://arxiv.org/abs/1811.12359>.
- 589 Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael
590 Tschannen. Weakly-supervised disentanglement without compromises, 2020. URL <https://arxiv.org/abs/2002.02886>.
591
592
- 593 Jeremy Maitin-Shepard and Laramie Leavitt. Tensorstore for high-performance, scalable array
storage. Computer software, 2022. URL <https://github.com/google/tensorstore>.

- 594 Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller.
595 Sparse feature circuits: Discovering and editing interpretable causal graphs in language models,
596 2025. URL <https://arxiv.org/abs/2403.19647>.
597
- 598 Gemma E. Moran and Bryon Aragam. Towards interpretable deep generative models via causal
599 representation learning, 2025. URL <https://arxiv.org/abs/2504.11609>.
600
- 601 Anish Mudide, Joshua Engels, Eric J. Michaud, Max Tegmark, and Christian Schroeder de Witt.
602 Efficient dictionary learning with switch sparse autoencoders, 2024. URL <https://arxiv.org/abs/2410.08201>.
603
- 604 Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and
605 hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024a.
606
- 607 Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry
608 of large language models. In *Forty-first International Conference on Machine Learning*, 2024b.
609
- 610 Goutham Rajendran, Simon Buchholz, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar.
611 Learning interpretable concepts: Unifying causal representation learning and foundation models,
612 2024. URL <https://arxiv.org/abs/2402.09236>.
613
- 614 Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner,
615 Anirudh Goyal, and Yoshua Bengio. Towards causal representation learning, 2021. URL <https://arxiv.org/abs/2102.11107>.
616
- 617 Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen,
618 Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L
619 Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers,
620 Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan.
621 Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Trans-
622 former Circuits Thread*, 2024. URL [https://transformer-circuits.pub/2024/
scaling-monosemanticity/index.html](https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html).

624 A TRAINING DETAILS

626 A.1 INPUT DATA CONSTRUCTION

- 627
- 628 • The 20231101 Wikipedia dump is used to construct the data. Approximately 500 million
629 English tokens are selected by taking the first 256 tokens among articles on English and
630 Simple English Wikipedia. The articles are selected by choosing the top 2 million longest
631 articles (using length as a proxy for quality to filter out low quality stub articles) and then
632 randomly among those articles until 500 million tokens have been selected.
 - 633 • For non-English tokens, articles are chosen randomly from the largest 128 non-English
634 Wikipedias (as measured by active users). The mix of languages is also proportional to the
635 number of active users. This is used as a rough proxy for Wikipedia quality.
 - 636 • Tokens are then packed into sequences to fill the Gemma 2-2B context window and ac-
637 tivations are collected using Penzai and stored on disk with Tensorstore (Johnson, 2024;
638 Maitin-Shepard & Leavitt, 2022).
 - 639 • BOS and EOS tokens are stripped from the data and shuffled on disk prior to training. SAE
640 training requires data access well in excess of 1GB/s to saturate a 8xA100 node and so the
641 data must be shuffled ahead of time.

643 A.2 MODEL IMPLEMENTATION AND HYPERPARAMETERS

- 644
- 645 • The hierarchical sparse autoencoder is implemented in JAX and Equinox (Bradbury et al.,
646 2018; Kidger & Garcia, 2021).
 - 647 • We train with a batch size of 32,512 and top-k of 32. When the number of sublatents per
expert is 16, the subspace dimension is 4. Otherwise, it is 8.

- We set the orthogonality penalty and top-level reconstruction coefficients to 0.1. The ℓ_1 coefficient is 0.001. For standard SAEs that require an auxiliary loss to prevent dead latents we use a coefficient of $1/30$.
- We use the adam optimizer with global norm clipping of 0.75 and b1 of 0.9.
- The learning rate is $5 \cdot 10^{-4}$, initialized to 10^{-11} with a 1,000 steps linear warmup and cosine decay over the remainder of the training run.
- Additionally, the regularizers also warmup from 0 over the first 1,000 steps.

A.3 TRAINING DYNAMICS

- We track an exponential moving average over 300 batches (i.e. roughly every million tokens) of the number of times latents activate. Both the auxiliary loss and orthogonality loss push the number of latents that don't activate in 1 million tokens (i.e. dead) well under 1%. The dead atom auxiliary loss coincidentally has exactly the same compute cost as the addition of hierarchy with the hyperparameters used for experiments. However, in terms of wall-time the H-SAE actually trains slightly faster due to the auxiliary loss being a much more memory-intensive operation using a naive implementation. We do not attempt to use maximally efficient implementations so detailed compute analysis is not conducted.

B ABLATIONS

B.1 PRE-PROCESSING

We conduct ablations by evaluating our H-SAE architecture applied to the token unembedding matrix of the Gemma 2-2B model, rather than internal activations, as the unembedding matrix provides a small data set that is more suitable for ablation studies. Gemma 2-2B has 256,128 tokens in its vocabulary, however many of these are relatively rare and meaningless tokens. After discarding these, we are left with 186,032 tokens.

Previous work (Park et al., 2024b) has shown that the semantic structure of the unembedding matrix is fundamentally non-Euclidean, meaning that the standard Euclidean inner product does not align orthogonality with semantic independence. Following this line of work, we whiten the unembedding matrix by multiplying it by the inverse square root of the covariance matrix, which has been shown to align the inner product with the underlying geometry of the data. Empirically, we observe that this whitening step is necessary for the model to learn meaningful features.

B.2 ABLATION SETUP

For the ablations, we apply our H-SAE architecture to this whitened matrix, using a top-level dictionary with 2048 latent vectors (experts) and 32 sublatents per expert, $k = 5$. We considered an ℓ_1 strength of 2.5×10^{-3} and an orthogonality penalty of 2.5×10^{-2} . As our focus was on ablations, we were not concerned with the absolute performance of the model, but rather with the relative performance of different configurations, so we did not tune hyperparameters beyond those being compared. We trained for 10,000 steps with a batch size of 8192 and a cosine-decay learning rate schedule starting at 8192×10^{-15} and peak value of 8192×10^{-6} .

We are interested in whether the orthogonality loss and ℓ_1 regularizer are necessary for interpretability and hierarchy. Because this is fundamentally a qualitative question, we do not report quantitative results. Instead, we qualitatively analyze the learned features and their hierarchical structure by examining the top activated features for a set of candidate tokens (e.g., “puppy”). We analyze these features in the same way as we do for the embeddings, examining the max-activating examples for each feature.

We test the orthogonality loss with and without the ℓ_1 regularizer, and find that the orthogonality loss does not seem to be necessary for interpretability and hierarchy. Nonetheless, we chose to keep it in our final runs on the embeddings as it helps reduce the number of dead latents, which is a practical concern.

We also test the ℓ_1 regularizer with and without the orthogonality loss, and find that it too does not seem to be necessary for interpretability and hierarchy. Ultimately, we chose to keep it for our final runs on the embeddings to allow for some adaptivity beyond the fixed top-k selection, as we have observed in practice that very low activations are often not meaningful, so we would like to encourage the model to use only the most informative features.

Displayed below in Figures 7, 8, 9, 10, 11, 12, and 13 are the example results of our ablations studies on the unembedding matrix. These were not chosen randomly but rather to show a variety of different types of features.



Figure 7: Ablation studies on the unembedding matrix show no obvious advantage from the orthogonality or ℓ_1 regularizers, though we chose to keep them for our final runs on the embeddings.

B.3 IMPORTANCE OF THE CAUSAL INNER PRODUCT

As briefly mentioned in our pre-processing section, there are theoretical reasons to believe that the Euclidean inner product does not align with the underlying geometry of the unembedding space. Our results below (in Figures 14, 15, 16, 17, 18, 19, and 20) validate this hypothesis, as we find that the unwhitened unembedding matrix does not yield meaningful features. The features are completely different from the baseline, and do not seem to have any coherent meaning, as illustrated below.

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

Explanations for word: 'Queen'

- ◆ Top-Level Feature 951 (Activation: 0.5322)
Words that maximally activate this feature:
[' King', ' king', 'King', ' kings', ' KING']
↳ Low-Level Feature: 24 (Activation: -0.00093935)
Words that maximally activate this low-level feature:
[' crowns', 'könig', ' crowns', ' crown', ' roy']
- ◆ Top-Level Feature 1724 (Activation: 0.2876)
Words that maximally activate this feature:
[' lady', ' woman', ' Lady', ' lady', ' Lady']
↳ Low-Level Feature: 15 (Activation: 0.00006847)
Words that maximally activate this low-level feature:
[' girlfriend', ' girlfriend', ' girlfriend', ' girlfriends', ' madam']
- ◆ Top-Level Feature 1247 (Activation: 0.2281)
Words that maximally activate this feature:
[' Qu', ' qu', 'Qu', ' QU', ' qu']
↳ Low-Level Feature: 13 (Activation: 0.00012654)
Words that maximally activate this low-level feature:
[' QUEUE', ' Que', 'Que', ' QUE', ' QUE']
- ◆ Top-Level Feature 1914 (Activation: 0.1337)
Words that maximally activate this feature:
[' Officer', ' Engineer', ' Professor', ' Trainer', ' Surgeon']
↳ Low-Level Feature: 29 (Activation: 0.00004217)
Words that maximally activate this low-level feature:
[' Presidents', ' Blogger', ' g', ' Maestro', ' Appellant']
- ◆ Top-Level Feature 939 (Activation: 0.0612)
Words that maximally activate this feature:
[' chairman', ' Chairman', ' chairman', ' Chairman', ' CEO']
↳ Low-Level Feature: 8 (Activation: 0.00004819)
Words that maximally activate this low-level feature:
[' President', ' president', ' President', ' president', ' PRESIDENT']

(a) Baseline

Explanations for word: 'Queen'

- ◆ Top-Level Feature 225 (Activation: 0.5392)
Words that maximally activate this feature:
[' king', ' King', 'King', ' kings', ' KING']
↳ Low-Level Feature: 29 (Activation: 0.00001101)
Words that maximally activate this low-level feature:
[' Princess', '公主', 'wang', ' Reino', ' Regno']
- ◆ Top-Level Feature 275 (Activation: 0.2505)
Words that maximally activate this feature:
[' Qu', ' qu', 'Qu', ' QU', ' qu']
↳ Low-Level Feature: 20 (Activation: 0.00013162)
Words that maximally activate this low-level feature:
['quán', ' Q', ' q', 'Qs', ' q']
- ◆ Top-Level Feature 863 (Activation: 0.1929)
Words that maximally activate this feature:
[' women', ' woman', ' Women', ' Woman', ' women']
↳ Low-Level Feature: 17 (Activation: 0.00006456)
Words that maximally activate this low-level feature:
[' dames', 'princess', 'femme', ' femmes', 'ladies']
- ◆ Top-Level Feature 1988 (Activation: 0.1217)
Words that maximally activate this feature:
[' Coach', ' Officer', ' Seller', ' Supervisor', ' Engineer']
↳ Low-Level Feature: 19 (Activation: 0.04881151)
Words that maximally activate this low-level feature:
[' Engineer', 'Engineer', ' Philosopher', 'Executive', ' Mama']
- ◆ Top-Level Feature 1560 (Activation: 0.0386)
Words that maximally activate this feature:
[' sheet', ' sheet', ' Sheet', 'Sheet', ' sheets']
↳ Low-Level Feature: 18 (Activation: 0.00012621)
Words that maximally activate this low-level feature:
['worksheet', 'ւր', 'ռը', ' feuille', 'Workbook']

(c) No L1

Explanations for word: 'Queen'

- ◆ Top-Level Feature 712 (Activation: 0.5264)
Words that maximally activate this feature:
[' King', ' king', 'King', ' kings', ' KING']
↳ Low-Level Feature: 12 (Activation: -0.00047339)
Words that maximally activate this low-level feature:
[' Royal', ' royal', 'Royal', 'royal', ' ROYAL']
- ◆ Top-Level Feature 1247 (Activation: 0.2378)
Words that maximally activate this feature:
[' qu', ' Qu', 'Qu', ' QU', ' qu']
↳ Low-Level Feature: 23 (Activation: 0.00012804)
Words that maximally activate this low-level feature:
['ق', 'ق', 'Que', 'QUE', 'ques']
- ◆ Top-Level Feature 1051 (Activation: 0.1930)
Words that maximally activate this feature:
[' girl', ' girls', ' Girl', ' girls', ' female']
↳ Low-Level Feature: 17 (Activation: 0.00011276)
Words that maximally activate this low-level feature:
[' Lady', 'Lady', ' lady', ' lady', ' LADY']
- ◆ Top-Level Feature 1911 (Activation: 0.1788)
Words that maximally activate this feature:
[' mama', ' Mama', 'Mama', ' mamma', 'mama']
↳ Low-Level Feature: 29 (Activation: 0.12758416)
Words that maximally activate this low-level feature:
['madre', ' Baba', 'e', ' mère', 'mother']
- ◆ Top-Level Feature 1260 (Activation: 0.0884)
Words that maximally activate this feature:
[' Father', ' Leader', ' Contractor', ' Supervisor', ' Resident']
↳ Low-Level Feature: 26 (Activation: 0.00015458)
Words that maximally activate this low-level feature:
[' Ambassador', 'Ambassador', ' Treasurer', ' Lieutenant', ' Auditor']

(b) No Orthogonality

Explanations for word: 'Queen'

- ◆ Top-Level Feature 1253 (Activation: 0.5076)
Words that maximally activate this feature:
[' King', ' king', 'King', ' kings', ' KING']
↳ Low-Level Feature: 3 (Activation: 0.00013319)
Words that maximally activate this low-level feature:
['crown', ' prince', '왕', ' hoàng', '太子']
- ◆ Top-Level Feature 1428 (Activation: 0.3291)
Words that maximally activate this feature:
[' woman', ' lady', ' Woman', ' women', 'woman']
↳ Low-Level Feature: 13 (Activation: 0.00006558)
Words that maximally activate this low-level feature:
[' nuns', ' 少女', 'LADY', ' actresses', ' heiress']
- ◆ Top-Level Feature 1849 (Activation: 0.2592)
Words that maximally activate this feature:
[' Q', ' q', 'Q', ' q', ' qu']
↳ Low-Level Feature: 5 (Activation: -0.00062655)
Words that maximally activate this low-level feature:
[' Quebec', 'Quebec', 'QUAD', ' Quadrant', ' Quests']
- ◆ Top-Level Feature 1967 (Activation: 0.1479)
Words that maximally activate this feature:
[' Governor', ' Father', ' Supervisor', ' Lieutenant', ' Secretary']
↳ Low-Level Feature: 8 (Activation: 0.00014508)
Words that maximally activate this low-level feature:
[' Elder', 'Elder', ' Governor', ' Inventor', ' Philosopher']
- ◆ Top-Level Feature 315 (Activation: 0.0570)
Words that maximally activate this feature:
[' queue', ' queues', ' Queue', 'queue', 'Queue']
↳ Low-Level Feature: 10 (Activation: -0.00282054)
Words that maximally activate this low-level feature:
['enqueue', ' Que', ' queue', 'enqueue', ' que']

(d) No Orthogonality or L1

Figure 8: Ablation studies on the unembedding matrix show no obvious advantage from the orthogonality or ℓ_1 regularizers, though we chose to keep them for our final runs on the embeddings.

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

Explanations for word: 'Chicago'

- ◆ Top-Level Feature 1927 (Activation: 0.4056)
Words that maximally activate this feature:
[' Toronto', ' Chicago', ' Atlanta', ' Mumbai', ' Denver']
↳ Low-Level Feature: 10 (Activation: 0.00009848)
Words that maximally activate this low-level feature:
[' Madrid', ' Madrid', ' Barcelona', ' Barcelona', ' Tucson']
- ◆ Top-Level Feature 859 (Activation: 0.1282)
Words that maximally activate this feature:
[' American', ' America', ' american', ' American', ' Americans']
↳ Low-Level Feature: 5 (Activation: 0.00002996)
Words that maximally activate this low-level feature:
[' USA', ' USA', ' الولايات', ' EUA', ' CBA']
- ◆ Top-Level Feature 175 (Activation: 0.1170)
Words that maximally activate this feature:
[' Texas', ' Alabama', ' Florida', ' Louisiana', ' Texas']
↳ Low-Level Feature: 19 (Activation: 0.05223560)
Words that maximally activate this low-level feature:
[' Iowa', ' Iowa', ' Kansas', ' Iowa', ' Kansas']
- ◆ Top-Level Feature 98 (Activation: 0.1094)
Words that maximally activate this feature:
[' Chrom', ' chrom', ' chrom', ' Chrome', ' Chrom']
↳ Low-Level Feature: 0 (Activation: -0.00002403)
Words that maximally activate this low-level feature:
[' Chromosome', ' Chromosome', ' browser', ' thermo', ' nhĩm']
- ◆ Top-Level Feature 1512 (Activation: 0.0887)
Words that maximally activate this feature:
[' il', ' il', ' IL', ' IL', ' IL']
↳ Low-Level Feature: 8 (Activation: 0.00015619)
Words that maximally activate this low-level feature:
[' ila', ' Illusion', ' ilo', ' ilation', ' ʻnʻ']

(a) Baseline

Explanations for word: 'Chicago'

- ◆ Top-Level Feature 1622 (Activation: 0.2890)
Words that maximally activate this feature:
[' Paris', ' Berlin', ' Tokyo', ' Madrid', ' Nairobi']
↳ Low-Level Feature: 4 (Activation: 0.05622706)
Words that maximally activate this low-level feature:
[' Chicago', ' Chicago', ' CHICAGO', ' chicago', ' Boston']
- ◆ Top-Level Feature 805 (Activation: 0.1590)
Words that maximally activate this feature:
[' Illinois', ' Missouri', ' Wisconsin', ' Michigan', ' Minnesota']
↳ Low-Level Feature: 13 (Activation: 0.00003956)
Words that maximally activate this low-level feature:
[' Delaware', ' Delaware', ' Montana', ' Dakota', ' Dakota']
- ◆ Top-Level Feature 546 (Activation: 0.0862)
Words that maximally activate this feature:
[' Ch', ' ch', ' CH', ' Ch', ' Cho']
↳ Low-Level Feature: 11 (Activation: 0.00009995)
Words that maximally activate this low-level feature:
[' CHI', ' echa', ' chir', ' chir', ' chin']
- ◆ Top-Level Feature 1756 (Activation: 0.0847)
Words that maximally activate this feature:
[' il', ' il', ' Ill', ' il', ' IL']
↳ Low-Level Feature: 14 (Activation: -0.00006478)
Words that maximally activate this low-level feature:
[' ʻnʻ', ' IL', ' cile', ' sill', ' cil']
- ◆ Top-Level Feature 175 (Activation: 0.0654)
Words that maximally activate this feature:
[' American', ' American', ' american', ' Americans', ' America']
↳ Low-Level Feature: 11 (Activation: 0.00004174)
Words that maximally activate this low-level feature:
[' Амери', ' Amerikan', ' CBA', ' 美国的', ' AMERICAN']

(b) No Orthogonality

Explanations for word: 'Chicago'

- ◆ Top-Level Feature 904 (Activation: 0.2868)
Words that maximally activate this feature:
[' Tokyo', ' Berlin', ' Madrid', ' Delhi', ' London']
↳ Low-Level Feature: 5 (Activation: 0.07947742)
Words that maximally activate this low-level feature:
[' Yogyakarta', ' Boston', ' Boston', ' boston', ' CHICAGO']
- ◆ Top-Level Feature 937 (Activation: 0.1200)
Words that maximally activate this feature:
[' Chrom', ' chrom', ' Chrome', ' chrom', ' Chrom']
↳ Low-Level Feature: 2 (Activation: -0.00002368)
Words that maximally activate this low-level feature:
[' browser', ' Browser', ' browsers', ' browser', ' Browser']
- ◆ Top-Level Feature 1815 (Activation: 0.0733)
Words that maximally activate this feature:
[' California', ' Pennsylvania', ' Massachusetts', ' California', ' Connecticut']
↳ Low-Level Feature: 24 (Activation: 0.03427194)
Words that maximally activate this low-level feature:
[' Iowa', ' Iowa', ' Missouri', ' Nebraska', ' Wisconsin']
- ◆ Top-Level Feature 1908 (Activation: 0.0595)
Words that maximally activate this feature:
[' Che', ' CHE', ' che', ' che', ' Che']
↳ Low-Level Feature: 16 (Activation: -0.00006747)
Words that maximally activate this low-level feature:
[' Chess', ' Chess', ' chess', ' chess', ' Ches']
- ◆ Top-Level Feature 1293 (Activation: 0.0562)
Words that maximally activate this feature:
[' American', ' Americans', ' America', ' American', ' american']
↳ Low-Level Feature: 9 (Activation: 0.00003683)
Words that maximally activate this low-level feature:
[' statunitense', ' الولايات', ' estadounidense', ' 미국', ' 美国']

(c) No L1

Explanations for word: 'Chicago'

- ◆ Top-Level Feature 904 (Activation: 0.2514)
Words that maximally activate this feature:
[' Tokyo', ' Paris', ' Madrid', ' Berlin', ' Delhi']
↳ Low-Level Feature: 17 (Activation: 0.05849538)
Words that maximally activate this low-level feature:
[' NYC', ' CHICAGO', ' Brooklyn', ' Chicago', ' Boston']
- ◆ Top-Level Feature 1246 (Activation: 0.0796)
Words that maximally activate this feature:
[' American', ' Americans', ' America', ' american', ' American']
↳ Low-Level Feature: 9 (Activation: 0.04920823)
Words that maximally activate this low-level feature:
[' américain', ' amerikan', ' amer', ' statunitense', ' USD']
- ◆ Top-Level Feature 624 (Activation: 0.0655)
Words that maximally activate this feature:
[' ch', ' Ch', ' CH', ' ch', ' CH']
↳ Low-Level Feature: 18 (Activation: 0.00012482)
Words that maximally activate this low-level feature:
[' chi', ' Chi', ' Chi', ' chi', ' CHI']
- ◆ Top-Level Feature 1584 (Activation: 0.0599)
Words that maximally activate this feature:
[' il', ' il', ' il', ' il', ' IL']
↳ Low-Level Feature: 20 (Activation: 0.00000981)
Words that maximally activate this low-level feature:
[' Wil', ' a', ' ILL', ' ildo', ' ilos']
- ◆ Top-Level Feature 1983 (Activation: 0.0564)
Words that maximally activate this feature:
[' chemical', ' Chemical', ' chemical', ' Chemical', ' chemicals']
↳ Low-Level Feature: 30 (Activation: 0.00002974)
Words that maximally activate this low-level feature:
[' química', ' kimia', ' chemists', ' 化学', ' 化学']

(d) No Orthogonality or L1

Figure 9: Ablation studies on the unembedding matrix show no obvious advantage from the orthogonality or ℓ_1 regularizers, though we chose to keep them for our final runs on the embeddings.

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

Explanations for word: 'London'

- ◆ Top-Level Feature 1927 (Activation: 0.3059)
Words that maximally activate this feature:
[' Toronto', ' Chicago', ' Atlanta', ' Mumbai', ' Denver']
↳ Low-Level Feature: 20 (Activation: 0.05103210)
Words that maximally activate this low-level feature:
[' Cairo', 'Cairo', 'cairo', ' London', 'London']
- ◆ Top-Level Feature 1197 (Activation: 0.1395)
Words that maximally activate this feature:
[' English', ' english', 'English', ' ENGLISH', 'english']
↳ Low-Level Feature: 22 (Activation: 0.10184363)
Words that maximally activate this low-level feature:
['British', ' british', 'british', ' Великобритания', '■']
- ◆ Top-Level Feature 1109 (Activation: 0.0617)
Words that maximally activate this feature:
[' Italy', ' Spain', ' India', ' Sweden', ' Greece']
↳ Low-Level Feature: 8 (Activation: 0.00007217)
Words that maximally activate this low-level feature:
['Brasil', ' Brasil', ' BRASIL', ' BRAZIL', 'brasil']
- ◆ Top-Level Feature 1494 (Activation: 0.0328)
Words that maximally activate this feature:
[' layer', ' Layer', ' layers', 'layer', 'Layer']
↳ Low-Level Feature: 5 (Activation: 0.00003101)
Words that maximally activate this low-level feature:
['lay', 'Lay', 'Lay', ' Lay', ' lay']
- ◆ Top-Level Feature 780 (Activation: 0.0312)
Words that maximally activate this feature:
[' Lamp', ' Lamp', ' amp', 'lamp', 'amp']
↳ Low-Level Feature: 14 (Activation: 0.00011100)
Words that maximally activate this low-level feature:
['mps', ' amplifiers', ' 𠂇', 'amping', 'Q']

(a) Baseline

Explanations for word: 'London'

- ◆ Top-Level Feature 1246 (Activation: 0.4000)
Words that maximally activate this feature:
[' UK', 'UK', ' London', 'London', ' Britain']
↳ Low-Level Feature: 32 (Activation: -0.00007063)
Words that maximally activate this low-level feature:
[' Gloucestershire', ' Warwickshire', ' Hertfordshire', ' Wiltshire', ' Oxfordshire']
- ◆ Top-Level Feature 1622 (Activation: 0.4351)
Words that maximally activate this feature:
[' Paris', ' Berlin', ' Tokyo', ' Nairobi', ' Nairobi']
↳ Low-Level Feature: 26 (Activation: 0.00009957)
Words that maximally activate this low-level feature:
[' Lagos', 'Lagos', ' Abuja', ' Brussels', ' Abuja']
- ◆ Top-Level Feature 1915 (Activation: 0.0706)
Words that maximally activate this feature:
[' L', 'L', ' 𠂇', ' LC', ' LR']
↳ Low-Level Feature: 20 (Activation: 0.00004070)
Words that maximally activate this low-level feature:
[' LOR', 'LOR', 'Loren', 'Lorde', 'LLE']
- ◆ Top-Level Feature 1109 (Activation: 0.0691)
Words that maximally activate this feature:
[' Spain', ' Italy', ' Poland', ' Russia', ' Sweden']
↳ Low-Level Feature: 20 (Activation: 0.00008624)
Words that maximally activate this low-level feature:
[' Angola', ' 𠂇', ' Sweden', ' Azerbaijan', ' Algeria']
- ◆ Top-Level Feature 1197 (Activation: 0.0492)
Words that maximally activate this feature:
[' English', ' english', 'English', ' ENGLISH', 'english']
↳ Low-Level Feature: 15 (Activation: 0.00001979)
Words that maximally activate this low-level feature:
[' angle', ' 英语', ' inglese', ' Engl', ' Ayr']

(b) No Orthogonality

Explanations for word: 'London'

- ◆ Top-Level Feature 904 (Activation: 0.4297)
Words that maximally activate this feature:
[' Tokyo', ' Berlin', ' Madrid', ' Delhi', ' London']
↳ Low-Level Feature: 27 (Activation: -0.0000157)
Words that maximally activate this low-level feature:
[' Helsinki', 'Helsinki', ' Copenhagen', ' Tallinn', ' Kiewe']
- ◆ Top-Level Feature 1796 (Activation: 0.1050)
Words that maximally activate this feature:
[' Nottingham', ' Leicester', ' Lancashire', ' Yorkshire', ' Gloucestershire']
↳ Low-Level Feature: 29 (Activation: 0.06130284)
Words that maximally activate this low-level feature:
[' UK', 'UK', ' britannique', ' イギリス', ' britann']
- ◆ Top-Level Feature 1233 (Activation: 0.0799)
Words that maximally activate this feature:
[' L', 'L', ' 𠂇', ' 𠂇', ' LR']
↳ Low-Level Feature: 12 (Activation: 0.00002094)
Words that maximally activate this low-level feature:
[' Lu', ' Lu', ' LU', ' Luc', ' LU']
- ◆ Top-Level Feature 1174 (Activation: 0.0453)
Words that maximally activate this feature:
[' Spain', ' Canada', ' Italy', ' France', ' Germany']
↳ Low-Level Feature: 24 (Activation: -0.00002373)
Words that maximally activate this low-level feature:
[' Italy', ' Italia', 'Italy', ' italy', ' ITALY']
- ◆ Top-Level Feature 1815 (Activation: 0.0342)
Words that maximally activate this feature:
[' California', ' Pennsylvania', ' Massachusetts', ' California', ' Connecticut']
↳ Low-Level Feature: 12 (Activation: 0.00007537)
Words that maximally activate this low-level feature:
[' Sonoma', ' Alabama', 'Alabama', ' California', 'Texas']

(c) No L1

Explanations for word: 'London'

- ◆ Top-Level Feature 904 (Activation: 0.3255)
Words that maximally activate this feature:
[' Tokyo', ' Paris', ' Madrid', ' Berlin', ' Delhi']
↳ Low-Level Feature: 26 (Activation: 0.00774494)
Words that maximally activate this low-level feature:
[' London', ' 伦敦', ' dubai', 'PARIS', ' ローマ']
- ◆ Top-Level Feature 1254 (Activation: 0.3215)
Words that maximally activate this feature:
[' British', ' Britain', 'British', ' UK', ' Brits']
↳ Low-Level Feature: 25 (Activation: 0.00004446)
Words that maximally activate this low-level feature:
[' Scotsman', ' 영', ' ■', ' Britton', ' Brexit']
- ◆ Top-Level Feature 1873 (Activation: 0.0714)
Words that maximally activate this feature:
[' Italy', ' Ireland', ' France', ' Brazil', ' Poland']
↳ Low-Level Feature: 10 (Activation: 0.00006569)
Words that maximally activate this low-level feature:
[' indonesia', ' Jamaica', ' INDONESIA', ' ישראל', ' Polsce']
- ◆ Top-Level Feature 1425 (Activation: 0.0353)
Words that maximally activate this feature:
[' L', 'L', ' 𠂇', ' LC', ' LR']
↳ Low-Level Feature: 19 (Activation: 0.00001737)
Words that maximally activate this low-level feature:
[' LUIS', ' LSM', ' 𠂇', ' 𠂇', ' L']
- ◆ Top-Level Feature 1109 (Activation: 0.0353)
Words that maximally activate this feature:
[' the', ' charge', ' flow', ' not', ' schedule']
↳ Low-Level Feature: 2 (Activation: 0.00006976)
Words that maximally activate this low-level feature:
[' the', ' not', ' email', ' sheet', ' the']

(d) No Orthogonality or L1

Figure 10: Ablation studies on the unembedding matrix show no obvious advantage from the orthogonality or ℓ_1 regularizers, though we chose to keep them for our final runs on the embeddings.

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

Explanations for word: 'Twitter'

- ◆ Top-Level Feature 1959 (Activation: 0.5850)
Words that maximally activate this feature:
['tweet', 'tweets', 'Tweet', 'tweet', 'tweeting']
↳ Low-Level Feature: 24 (Activation: 0.0001509)
Words that maximally activate this low-level feature:
['tweeting', 'twitter', 'Twitter', 'TWITTER', 'twitter']
- ◆ Top-Level Feature 1939 (Activation: 0.3562)
Words that maximally activate this feature:
['YouTube', 'Youtube', 'youtube', 'Google', 'YouTube']
↳ Low-Level Feature: 7 (Activation: -0.00020170)
Words that maximally activate this low-level feature:
['goog', 'Flickr', 'yahoo', 'Wikipedia', 'Wiki']
- ◆ Top-Level Feature 506 (Activation: 0.1397)
Words that maximally activate this feature:
['social', 'Social', 'social', 'social', 'SOCIAL']
↳ Low-Level Feature: 3 (Activation: -0.0001523)
Words that maximally activate this low-level feature:
['socialists', 'socialist', 'Socialist', 'Socialists', 'socialism']
- ◆ Top-Level Feature 1136 (Activation: 0.0638)
Words that maximally activate this feature:
['blockchain', 'Blockchain', 'NFTs', 'TikTok', 'Blockchain']
↳ Low-Level Feature: 27 (Activation: 0.00007173)
Words that maximally activate this low-level feature:
['新冠', 'Trump', 'Trump', 'Biden', 'https']
- ◆ Top-Level Feature 2046 (Activation: 0.0511)
Words that maximally activate this feature:
['Tuesday', 'Wednesday', 'Thursday', 'Monday', 'Friday']
↳ Low-Level Feature: 22 (Activation: 0.00005836)
Words that maximally activate this low-level feature:
['Sunday', 'sunday', 'sunday', 'SUNDAY', 'SUNDAY']

(a) Baseline

Explanations for word: 'Twitter'

- ◆ Top-Level Feature 1694 (Activation: 0.4420)
Words that maximally activate this feature:
['Facebook', 'facebook', 'YouTube', 'Facebook', 'YouTube']
↳ Low-Level Feature: 31 (Activation: 0.11588523)
Words that maximally activate this low-level feature:
['tweet', 'Tweet', 'Tweet', 'twitter', 'Twitter']
- ◆ Top-Level Feature 902 (Activation: 0.0544)
Words that maximally activate this feature:
['social', 'Social', 'social', 'social', 'SOCIAL']
↳ Low-Level Feature: 16 (Activation: -0.01861620)
Words that maximally activate this low-level feature:
['SOC', 'soc', '社会', 'socials', 'soc']
- ◆ Top-Level Feature 1939 (Activation: 0.0433)
Words that maximally activate this feature:
['web', 'Web', 'web', 'web', 'WEB']
↳ Low-Level Feature: 4 (Activation: 0.00003201)
Words that maximally activate this low-level feature:
['ウェブ', 'weber', 'webview', 'webs', 'webpage']
- ◆ Top-Level Feature 1210 (Activation: 0.0392)
Words that maximally activate this feature:
['Java', 'Java', 'JAVA', 'java', 'PHP']
↳ Low-Level Feature: 22 (Activation: 0.00003138)
Words that maximally activate this low-level feature:
['MySQL', 'mysql', 'mysql', 'MySQL', 'Kubernetes']
- ◆ Top-Level Feature 1549 (Activation: 0.0377)
Words that maximally activate this feature:
['Smartphone', 'smartphone', 'Smartphones', 'smartphones', 'smartphone']
↳ Low-Level Feature: 5 (Activation: 0.00001795)
Words that maximally activate this low-level feature:
['iPhone', 'iPhone', 'iphone', 'iphone', 'iPad']

(b) No Orthogonality

Explanations for word: 'Twitter'

- ◆ Top-Level Feature 512 (Activation: 0.5074)
Words that maximally activate this feature:
['tweet', 'tweets', 'Tweet', 'tweet', 'tweeting']
↳ Low-Level Feature: 26 (Activation: 0.00009981)
Words that maximally activate this low-level feature:
['Twit', 'Twe', 'Hashtag', 'twe', 'Twitter']
- ◆ Top-Level Feature 1403 (Activation: 0.3461)
Words that maximally activate this feature:
['YouTube', 'Youtube', 'Pinterest', 'LinkedIn', 'YouTube']
↳ Low-Level Feature: 30 (Activation: 0.04850825)
Words that maximally activate this low-level feature:
['instagram', 'instagram', 'INSTAGRAM', 'Instagram', 'Instagram']
- ◆ Top-Level Feature 1549 (Activation: 0.0641)
Words that maximally activate this feature:
['social', 'Social', 'social', 'social', 'SOCIAL']
↳ Low-Level Feature: 14 (Activation: -0.00001860)
Words that maximally activate this low-level feature:
['socials', 'Social', 'socially', 'social', 'social']
- ◆ Top-Level Feature 450 (Activation: 0.0560)
Words that maximally activate this feature:
['blog', 'Blog', 'blogs', 'blog', 'blog']
↳ Low-Level Feature: 0 (Activation: -0.00017220)
Words that maximally activate this low-level feature:
['article', 'artikel', 'Artikel', '博客', 'Blog']
- ◆ Top-Level Feature 1563 (Activation: 0.0417)
Words that maximally activate this feature:
['web', 'Web', 'web', 'web', 'WEB']
↳ Low-Level Feature: 26 (Activation: 0.00013382)
Words that maximally activate this low-level feature:
['Internet', 'Internet', 'internet', 'internet', 'INTERNET']

(c) No L1

Explanations for word: 'Twitter'

- ◆ Top-Level Feature 1618 (Activation: 0.6319)
Words that maximally activate this feature:
['tweet', 'tweets', 'twitter', 'Tweet', 'Twitter']
↳ Low-Level Feature: 17 (Activation: 0.00006684)
Words that maximally activate this low-level feature:
['Blogging', 'blogging', 'Blogger', 'TWITTER', 'hashtag']
- ◆ Top-Level Feature 1045 (Activation: 0.1111)
Words that maximally activate this feature:
['social', 'Social', 'social', 'social', 'SOCIAL']
↳ Low-Level Feature: 1 (Activation: 0.02887224)
Words that maximally activate this low-level feature:
['Facebook', 'Facebook', 'facebook', 'FACEBOOK', 'sociable']
- ◆ Top-Level Feature 1245 (Activation: 0.0928)
Words that maximally activate this feature:
['Disney', 'Hasbro', 'Pfizer', 'Nestlé', 'Disney']
↳ Low-Level Feature: 16 (Activation: 0.00003986)
Words that maximally activate this low-level feature:
['FAO', 'Walgreens', 'Cisco', 'Daimler', 'UNHCR']
- ◆ Top-Level Feature 578 (Activation: 0.0844)
Words that maximally activate this feature:
['blockchain', 'Blockchain', 'TikTok', 'cryptocurrency', 'NFTs']
↳ Low-Level Feature: 12 (Activation: -0.00022477)
Words that maximally activate this low-level feature:
['crypto', 'CRYPTO', 'Crypto', 'crypto', 'Vegan']
- ◆ Top-Level Feature 1594 (Activation: 0.0561)
Words that maximally activate this feature:
['telephone', 'Telephone', 'telephone', 'phone', 'Telephone']
↳ Low-Level Feature: 25 (Activation: -0.00012464)
Words that maximally activate this low-level feature:
['Sms', 'iphone', 'telemetry', 'télé', 'Tele']

(d) No Orthogonality or L1

Figure 11: Ablation studies on the unembedding matrix show no obvious advantage from the orthogonality or ℓ_1 regularizers, though we chose to keep them for our final runs on the embeddings.

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

Explanations for word: 'python'

- ◆ Top-Level Feature 547 (Activation: 0.0955)
Words that maximally activate this feature:
['pit', 'Pit', 'Pit', 'pit', 'pits']
↳ Low-Level Feature: 9 (Activation: 0.07605679)
Words that maximally activate this low-level feature:
['py', 'Python', 'Python', 'python', 'python']
- ◆ Top-Level Feature 734 (Activation: 0.0839)
Words that maximally activate this feature:
['script', 'Script', 'scripts', 'Script', 'script']
↳ Low-Level Feature: 4 (Activation: 0.00004199)
Words that maximally activate this low-level feature:
['Script']
- ◆ Top-Level Feature 51 (Activation: 0.0826)
Words that maximally activate this feature:
['snap', 'snaps', 'Sna', 'sna', 'snap']
↳ Low-Level Feature: 16 (Activation: -0.00001486)
Words that maximally activate this low-level feature:
['Snap', 'snapping', 'SNA', 'Sni', 'sna']
- ◆ Top-Level Feature 1867 (Activation: 0.0624)
Words that maximally activate this feature:
['Checkbox', 'Database', 'TextBox', 'nongodb', 'ToDo']
↳ Low-Level Feature: 19 (Activation: 0.06658750)
Words that maximally activate this low-level feature:
['Javascript', 'javascript', 'JAVA', 'Kotlin', 'TypeScript']
- ◆ Top-Level Feature 1316 (Activation: 0.0506)
Words that maximally activate this feature:
['ph', 'Ph', 'ph', 'PH']
↳ Low-Level Feature: 17 (Activation: 0.00003440)
Words that maximally activate this low-level feature:
['Philip', 'Phillip', 'Philip', 'Phillips', 'Phillip']

(a) Baseline

Explanations for word: 'python'

- ◆ Top-Level Feature 770 (Activation: 0.4408)
Words that maximally activate this feature:
['Py', 'py', 'Py', 'py', 'PY']
↳ Low-Level Feature: 25 (Activation: -0.00062331)
Words that maximally activate this low-level feature:
['numpy', 'pandas', 'numpy', 'django', 'pygame']
- ◆ Top-Level Feature 1210 (Activation: 0.3428)
Words that maximally activate this feature:
['Java', 'Java', 'JAVA', 'java', 'PHP']
↳ Low-Level Feature: 22 (Activation: 0.00007985)
Words that maximally activate this low-level feature:
['MySQL', 'Mysql', 'mysql', 'MySQL', 'Kubernetes']
- ◆ Top-Level Feature 1293 (Activation: 0.0737)
Words that maximally activate this feature:
['tiger', 'Tiger', 'Tiger', 'tigers', 'elephant']
↳ Low-Level Feature: 21 (Activation: 0.00015797)
Words that maximally activate this low-level feature:
['Dinosaur', 'alligator', 'Turtle', 'Dragons', 'turtle']
- ◆ Top-Level Feature 1950 (Activation: 0.0728)
Words that maximally activate this feature:
['script', 'Script', 'scripts', 'Script', 'script']
↳ Low-Level Feature: 5 (Activation: -0.00019795)
Words that maximally activate this low-level feature:
['scripted', 'SCRIPT', 'Scrip', 'SCRIPT', 'cueha']
- ◆ Top-Level Feature 1246 (Activation: 0.0440)
Words that maximally activate this feature:
['UK', 'UK', 'London', 'London', 'Britain']
↳ Low-Level Feature: 19 (Activation: 0.00006086)
Words that maximally activate this low-level feature:
['england', 'london', 'british', 'ukh', 'Hertfordshire']

(b) No Orthogonality

Explanations for word: 'python'

- ◆ Top-Level Feature 1294 (Activation: 0.4444)
Words that maximally activate this feature:
['Java', 'JAVA', 'Java', 'Python', 'python']
↳ Low-Level Feature: 5 (Activation: -0.00015276)
Words that maximally activate this low-level feature:
['Postgres', 'postgres', 'PostgreSQL', 'MySQL', 'MySQL']
- ◆ Top-Level Feature 2039 (Activation: 0.1708)
Words that maximally activate this feature:
['snake', 'Snake', 'snake', 'Snake', 'snakes']
↳ Low-Level Feature: 29 (Activation: 0.00013670)
Words that maximally activate this low-level feature:
['viper', 'viper', 'venomous', 'cobra', 'scorpion']
- ◆ Top-Level Feature 823 (Activation: 0.1585)
Words that maximally activate this feature:
['pi', 'Pi', 'pi', 'PI', 'PI']
↳ Low-Level Feature: 16 (Activation: 0.00014878)
Words that maximally activate this low-level feature:
['piety', 'PIC', 'pi', 'Pi', 'Pyramid']
- ◆ Top-Level Feature 1622 (Activation: 0.1320)
Words that maximally activate this feature:
['py', 'PY', 'PY', 'py', 'ry']
↳ Low-Level Feature: 23 (Activation: -0.00012211)
Words that maximally activate this low-level feature:
['Blythe', 'aly', 'ovy', 'dy', 'Nya']
- ◆ Top-Level Feature 1774 (Activation: 0.0688)
Words that maximally activate this feature:
['pd', 'dm', 'dt', 'md', 'ml']
↳ Low-Level Feature: 29 (Activation: 0.00010894)
Words that maximally activate this low-level feature:
['ctx', 'dll', 'dst', 'iv', 'mb']

(c) No L1

Explanations for word: 'python'

- ◆ Top-Level Feature 251 (Activation: 0.0985)
Words that maximally activate this feature:
['hd', 'fb', 'cb', 'gps', 'pc']
↳ Low-Level Feature: 9 (Activation: 0.00009679)
Words that maximally activate this low-level feature:
['pc', 'ai', 'ford', 'ajax', 'unicode']
- ◆ Top-Level Feature 798 (Activation: 0.0771)
Words that maximally activate this feature:
['pi', 'Pi', 'pi', 'PI', 'PI']
↳ Low-Level Feature: 26 (Activation: 0.23805813)
Words that maximally activate this low-level feature:
['Python', 'piercing', 'pim', 'opi', 'Piers']
- ◆ Top-Level Feature 1866 (Activation: 0.0351)
Words that maximally activate this feature:
['tiger', 'Tiger', 'Lion', 'Lion', 'wolf']
↳ Low-Level Feature: 29 (Activation: 0.00007088)
Words that maximally activate this low-level feature:
['turtle', 'penguin', 'Gecko', 'dolphin', 'hedgehog']
- ◆ Top-Level Feature 594 (Activation: 0.0280)
Words that maximally activate this feature:
['script', 'Script', 'scripts', 'Script', 'script']
↳ Low-Level Feature: 27 (Activation: 0.00010754)
Words that maximally activate this low-level feature:
['javascript', 'scripture', 'manuscrit', 'kìch', 'скри']
- ◆ Top-Level Feature 1499 (Activation: 0.0255)
Words that maximally activate this feature:
['plastic', 'Plastic', 'plastic', 'plastics', 'Plas']
↳ Low-Level Feature: 25 (Activation: 0.01944395)
Words that maximally activate this low-level feature:
['YAML', 'YAML', 'jackson', 'لاست', 'JSON']

(d) No Orthogonality or L1

Figure 12: Ablation studies on the unembedding matrix show no obvious advantage from the orthogonality or ℓ_1 regularizers, though we chose to keep them for our final runs on the embeddings.

1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

Explanations for word: 'Bayesian'

- ◆ Top-Level Feature 901 (Activation: 0.1695)
Words that maximally activate this feature:
['Jacobian', 'differential', 'bilinear', 'piecewise', 'variational']
↳ Low-Level Feature: 31 (Activation: 0.00010277)
Words that maximally activate this low-level feature:
['singularities', 'oscillatory', 'moduli', 'sinusoidal', 'trigonometric']
- ◆ Top-Level Feature 2010 (Activation: 0.1214)
Words that maximally activate this feature:
['probability', 'probabilities', 'Probability', 'probability', 'Probab']
↳ Low-Level Feature: 15 (Activation: 0.00001387)
Words that maximally activate this low-level feature:
['prospects', 'Probable', 'probable', 'Probable', 'probable']
- ◆ Top-Level Feature 870 (Activation: 0.0941)
Words that maximally activate this feature:
['Catholic', 'Hindu', 'Catholic', 'Muslim', 'Christian']
↳ Low-Level Feature: 31 (Activation: -0.00001035)
Words that maximally activate this low-level feature:
['Aryan', 'Huguenot', 'Aryan', 'Gothic', 'Gregorian']
- ◆ Top-Level Feature 1037 (Activation: 0.0081)
Words that maximally activate this feature:
['ba', 'Ba', 'Ba', 'ba', 'BA']
↳ Low-Level Feature: 30 (Activation: 0.07795684)
Words that maximally activate this low-level feature:
['Ba', 'bay', 'Bay', 'Bay', 'BA']
- ◆ Top-Level Feature 404 (Activation: 0.0781)
Words that maximally activate this feature:
['statistics', 'Statistics', 'statistical', 'statistic', 'stats']
↳ Low-Level Feature: 12 (Activation: 0.00004084)
Words that maximally activate this low-level feature:
['thong', 'stat', 'Estad', 'statistics', 'Statistical']

(a) Baseline

Explanations for word: 'Bayesian'

- ◆ Top-Level Feature 1054 (Activation: 0.0860)
Words that maximally activate this feature:
['Ba', 'ba', 'Ba', 'ba', 'BA']
↳ Low-Level Feature: 4 (Activation: 0.11283194)
Words that maximally activate this low-level feature:
['BAY', 'bays', 'Bays', 'Bayesian', 'baie']
- ◆ Top-Level Feature 1852 (Activation: 0.0789)
Words that maximally activate this feature:
['mathematical', 'Mathematical', 'mathematics', 'math', 'Mathematical']
↳ Low-Level Feature: 13 (Activation: -0.00007937)
Words that maximally activate this low-level feature:
['maths', 'Mathematics', 'Calculus', 'maths', 'Maths']
- ◆ Top-Level Feature 1934 (Activation: 0.0594)
Words that maximally activate this feature:
['Einstein', 'Descartes', 'Spinoza', 'Mozart', 'Einstein']
↳ Low-Level Feature: 28 (Activation: 0.00009781)
Words that maximally activate this low-level feature:
['Shakespeare', 'Shakespeare', 'Macbeth', 'Othello', 'Napoleon']
- ◆ Top-Level Feature 1210 (Activation: 0.0559)
Words that maximally activate this feature:
['Java', 'Java', 'JAVA', 'Java', 'PHP']
↳ Low-Level Feature: 22 (Activation: -0.00002095)
Words that maximally activate this low-level feature:
['MySQL', 'Mysql', 'mysql', 'MySQL', 'Kubernetes']
- ◆ Top-Level Feature 64 (Activation: 0.0555)
Words that maximally activate this feature:
['adiabatic', 'phonon', 'anisotropic', 'oscillatory', 'linearized']
↳ Low-Level Feature: 4 (Activation: -0.00312077)
Words that maximally activate this low-level feature:
['dielectric', 'depolar', 'sintering', 'conformal', 'sinter']

(b) No Orthogonality

Explanations for word: 'Bayesian'

- ◆ Top-Level Feature 1951 (Activation: 0.1547)
Words that maximally activate this feature:
['ay', 'AY', 'Ay', 'ay', 'Ay']
↳ Low-Level Feature: 3 (Activation: 0.00001630)
Words that maximally activate this low-level feature:
['zey', 'vey', 'Mey', 'aley', 'Vey']
- ◆ Top-Level Feature 468 (Activation: 0.1542)
Words that maximally activate this feature:
['probability', 'probabilities', 'Probability', 'probability', 'Probab']
↳ Low-Level Feature: 30 (Activation: 0.00002666)
Words that maximally activate this low-level feature:
['Probability', 'Probability', 'unlikely', 'Probab', 'probabilities']
- ◆ Top-Level Feature 1561 (Activation: 0.1399)
Words that maximally activate this feature:
['Ba', 'ba', 'Ba', 'ba', 'BA']
↳ Low-Level Feature: 4 (Activation: 0.00008041)
Words that maximally activate this low-level feature:
['Bail', 'Bail', 'bail', 'bail', 'bailout']
- ◆ Top-Level Feature 204 (Activation: 0.0992)
Words that maximally activate this feature:
['anisotropic', 'adiabatic', 'oscillatory', 'phonon', 'isothermal']
↳ Low-Level Feature: 24 (Activation: 0.00013688)
Words that maximally activate this low-level feature:
['cartesian', 'eigenvectors', 'trigonometric', 'Jacobian', 'discriminant']
- ◆ Top-Level Feature 1837 (Activation: 0.0812)
Words that maximally activate this feature:
['statistics', 'Statistics', 'statistic', 'statistical', 'stats']
↳ Low-Level Feature: 24 (Activation: -0.00002423)
Words that maximally activate this low-level feature:
['analytics', 'statut', 'statistic', 'stat', 'Estad']

(c) No L1

Explanations for word: 'Bayesian'

- ◆ Top-Level Feature 217 (Activation: 0.1110)
Words that maximally activate this feature:
['statistics', 'Statistics', 'statistic', 'statistical', 'Statistics']
↳ Low-Level Feature: 27 (Activation: 0.00000954)
Words that maximally activate this low-level feature:
['Estat', 'statysty', 'Statisti', 'Estad', 'Statistik']
- ◆ Top-Level Feature 312 (Activation: 0.0925)
Words that maximally activate this feature:
['AY', 'ay', 'ray', 'Ay', 'ay']
↳ Low-Level Feature: 14 (Activation: 0.08490946)
Words that maximally activate this low-level feature:
['bay', 'Bay', 'Bay', 'BAY', 'BAY']
- ◆ Top-Level Feature 929 (Activation: 0.0588)
Words that maximally activate this feature:
['Spinoza', 'Chaucer', 'Machiavelli', 'Proust', 'Baudelaire']
↳ Low-Level Feature: 21 (Activation: 0.00013820)
Words that maximally activate this low-level feature:
['Newton', 'Gauss', 'Euler', 'Euler', 'Einstein']
- ◆ Top-Level Feature 955 (Activation: 0.0570)
Words that maximally activate this feature:
['Brazilian', 'Mexican', 'Italian', 'Turkish', 'Vietnamese']
↳ Low-Level Feature: 5 (Activation: -0.00007113)
Words that maximally activate this low-level feature:
['Italian', 'Italian', 'Italians', 'Italian', 'Italian']
- ◆ Top-Level Feature 810 (Activation: 0.0470)
Words that maximally activate this feature:
['math', 'mathematics', 'Math', 'maths', 'Math']
↳ Low-Level Feature: 5 (Activation: 0.00011147)
Words that maximally activate this low-level feature:
['algebra', 'arithmetic', 'algebraic', '数学', 'Algebra']

(d) No Orthogonality or L1

Figure 13: Ablation studies on the unembedding matrix show no obvious advantage from the orthogonality or ℓ_1 regularizers, though we chose to keep them for our final runs on the embeddings.

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

Explanations for word: 'puppy'

- ◆ Top-Level Feature 732 (Activation: 0.3964)
Words that maximally activate this feature:
[' dog', ' Dog', 'Dog', 'dog', ' dogs']
↳ Low-Level Feature: 31 (Activation: -0.00061521)
Words that maximally activate this low-level feature:
[' barks', ' coBaki', ' woof', ' coBaka', ' Labrador']
- ◆ Top-Level Feature 956 (Activation: 0.0883)
Words that maximally activate this feature:
[' pig', ' goat', ' pigs', ' monkey', ' Goat']
↳ Low-Level Feature: 12 (Activation: 0.0000925)
Words that maximally activate this low-level feature:
[' horse', ' Horse', ' Horse', ' pony', ' Pony']
- ◆ Top-Level Feature 117 (Activation: 0.0525)
Words that maximally activate this feature:
[' boy', ' Boy', ' boy', ' Boy', ' BOY']
↳ Low-Level Feature: 0 (Activation: 0.21417145)
Words that maximally activate this low-level feature:
[' menina', ' boya', ' ガール', ' BOYS', ' GIRLS']
- ◆ Top-Level Feature 487 (Activation: 0.0457)
Words that maximally activate this feature:
[' pet', ' Pet', 'pet', ' Pet', ' PET']
↳ Low-Level Feature: 4 (Activation: 0.00009525)
Words that maximally activate this low-level feature:
[' PETER', ' Pedro', ' Pedro', ' petrol', ' Petrol']
- ◆ Top-Level Feature 1809 (Activation: 0.0437)
Words that maximally activate this feature:
[' patient', ' shopper', ' attendee', ' subscriber', ' sufferer']
↳ Low-Level Feature: 8 (Activation: 0.00009030)
Words that maximally activate this low-level feature:
[' Passenger', ' passenger', ' passenger', ' Passenger', ' guest']

(a) Baseline

Explanations for word: 'puppy'

- ◆ Top-Level Feature 1986 (Activation: 0.1022)
Words that maximally activate this feature:
[' 今後も', ' お花', ' 一歩', ' 本当の', ' これからも']
↳ Low-Level Feature: 6 (Activation: 0.04209925)
Words that maximally activate this low-level feature:
[' 腹', ' 手作り', ' 今後も', ' 痛い', ' 一歩']
- ◆ Top-Level Feature 1442 (Activation: 0.1020)
Words that maximally activate this feature:
[' сям', ' unlicensed', ' меша', ' ohn', ' nguyêñ']
↳ Low-Level Feature: 23 (Activation: 0.06209645)
Words that maximally activate this low-level feature:
[' founder', ' nguyêñ', ' ohn', ' deflected', ' deflection']
- ◆ Top-Level Feature 967 (Activation: 0.0865)
Words that maximally activate this feature:
[' baixo', ' partidas', ' iken', ' oko', ' 兵']
↳ Low-Level Feature: 21 (Activation: 0.03897284)
Words that maximally activate this low-level feature:
[' Presenter', ' に限', ' dorowopa', ' Englishman', ' to']
- ◆ Top-Level Feature 402 (Activation: 0.0865)
Words that maximally activate this feature:
[' せ', ' possess', ' RESERVE', ' Kimmel', ' Muslims']
↳ Low-Level Feature: 30 (Activation: 0.03146359)
Words that maximally activate this low-level feature:
[' upkeep', ' upkeep', ' Kimmel']
- ◆ Top-Level Feature 2036 (Activation: 0.0858)
Words that maximally activate this feature:
[' ʻuk, ' ujin', ' membunuh', ' diplom', ' ʻim']
↳ Low-Level Feature: 27 (Activation: 0.02508827)
Words that maximally activate this low-level feature:
[' Uttarakhand', ' denial', ' heavier', ' neutrino', ' CAS']

(b) No Whitening

Figure 14: The causal inner product is necessary for the model to learn meaningful features.

Explanations for word: 'Queen'

- ◆ Top-Level Feature 951 (Activation: 0.5322)
Words that maximally activate this feature:
[' King', ' king', ' King', ' kings', ' KING']
↳ Low-Level Feature: 24 (Activation: -0.00059395)
Words that maximally activate this low-level feature:
[' crowns', ' könig', ' crowns', ' crown', ' roy']
- ◆ Top-Level Feature 1724 (Activation: 0.2876)
Words that maximally activate this feature:
[' lady', ' woman', ' Lady', ' lady', ' Lady']
↳ Low-Level Feature: 15 (Activation: 0.00006847)
Words that maximally activate this low-level feature:
[' girlfriend', ' girlfriend', ' girlfriend', ' girlfriends', ' madam']
- ◆ Top-Level Feature 1247 (Activation: 0.2281)
Words that maximally activate this feature:
[' Qu', ' qu', ' Qu', ' QU', ' qu']
↳ Low-Level Feature: 13 (Activation: 0.00012654)
Words that maximally activate this low-level feature:
[' QUEUE', ' Que', ' Que', ' QUE', ' QUE']
- ◆ Top-Level Feature 1914 (Activation: 0.1337)
Words that maximally activate this feature:
[' Officer', ' Engineer', ' Professor', ' Trainer', ' Surgeon']
↳ Low-Level Feature: 29 (Activation: 0.00004217)
Words that maximally activate this low-level feature:
[' Presidents', ' Blogger', ' g', ' Maestro', ' Appellant']
- ◆ Top-Level Feature 939 (Activation: 0.0612)
Words that maximally activate this feature:
[' chairman', ' Chairman', ' chairman', ' Chairman', ' CEO']
↳ Low-Level Feature: 8 (Activation: 0.00004819)
Words that maximally activate this low-level feature:
[' President', ' president', ' President', ' president', ' PRESIDENT']

(a) Baseline

Explanations for word: 'Queen'

- ◆ Top-Level Feature 1707 (Activation: 0.1521)
Words that maximally activate this feature:
[' 2', ' M', ' v', ' ného', ' r']
↳ Low-Level Feature: 21 (Activation: 0.14752719)
Words that maximally activate this low-level feature:
[' mlhões', ' かけた', ' ьно', ' ToList', ' desses']
- ◆ Top-Level Feature 1473 (Activation: 0.1479)
Words that maximally activate this feature:
[' \r', ' 2', ' 9', ' =', ' 3']
↳ Low-Level Feature: 14 (Activation: 0.19582644)
Words that maximally activate this low-level feature:
[' \r', ' =', ' 4', ' \r', ' 3']
- ◆ Top-Level Feature 1002 (Activation: 0.1417)
Words that maximally activate this feature:
[' ʻH', ' Ersten', ' meg', ' öt', ' vs']
↳ Low-Level Feature: 2 (Activation: 0.34217829)
Words that maximally activate this low-level feature:
[' ʻH', ' amal', ' öt', ' ʻstra']
- ◆ Top-Level Feature 320 (Activation: 0.1412)
Words that maximally activate this feature:
[' honda', ' casu', ' Thòì', ' esting', ' poi']
↳ Low-Level Feature: 0 (Activation: 0.07256843)
Words that maximally activate this low-level feature:
[' Array', ' Options', ' King', ' poi', ' historians']
- ◆ Top-Level Feature 1829 (Activation: 0.1406)
Words that maximally activate this feature:
[' ', ' ', ' ', ' とりあえず', ' 明日', ' nny']
↳ Low-Level Feature: 9 (Activation: 0.05275402)
Words that maximally activate this low-level feature:
[' Registro', ' perpetuated', ' 心', ' \r', ' 漆']

(b) No Whitening

Figure 15: The causal inner product is necessary for the model to learn meaningful features.

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241

```

Explanations for word: 'Twitter'
=====
◆ Top-Level Feature 1959 (Activation: 0.5850)
Words that maximally activate this feature:
['tweet', 'tweets', 'Tweet', 'tweet', 'tweeting']
└─ Low-Level Feature: 24 (Activation: 0.0001509)
  Words that maximally activate this low-level feature:
  ['tweeting', 'twitter', 'Twitter', 'TWITTER', 'twitter']

◆ Top-Level Feature 1939 (Activation: 0.3562)
Words that maximally activate this feature:
['YouTube', 'Youtube', 'youtube', 'Google', 'YouTube']
└─ Low-Level Feature: 7 (Activation: -0.00020170)
  Words that maximally activate this low-level feature:
  ['goog', 'Flickr', 'yahoo', 'Wikipedia', 'Wiki']

◆ Top-Level Feature 506 (Activation: 0.1397)
Words that maximally activate this feature:
['social', 'Social', 'social', 'social', 'SOCIAL']
└─ Low-Level Feature: 3 (Activation: -0.00001523)
  Words that maximally activate this low-level feature:
  ['socialists', 'socialist', 'Socialist', 'Socialists', 'socialism']

◆ Top-Level Feature 1136 (Activation: 0.0638)
Words that maximally activate this feature:
['blockchain', 'Blockchain', 'NFTs', 'TikTok', 'Blockchain']
└─ Low-Level Feature: 27 (Activation: 0.00007173)
  Words that maximally activate this low-level feature:
  ['新冠', 'Trump', 'Trump', 'Biden', 'https']

◆ Top-Level Feature 2046 (Activation: 0.0511)
Words that maximally activate this feature:
['Tuesday', 'Wednesday', 'Thursday', 'Monday', 'Friday']
└─ Low-Level Feature: 22 (Activation: 0.00005836)
  Words that maximally activate this low-level feature:
  ['Sunday', 'sunday', 'sunday', 'SUNDAY', 'SUNDAY']
    
```

(a) Baseline

```

Explanations for word: 'Twitter'
=====
◆ Top-Level Feature 1902 (Activation: 0.2157)
Words that maximally activate this feature:
['gennaio', 'giugno', 'dicembre', 'settembre', '/>\r']
└─ Low-Level Feature: 11 (Activation: 0.25783429)
  Words that maximally activate this low-level feature:
  ['gennaio', 'giugno', 'dicembre', 'settembre', '/>\r']

◆ Top-Level Feature 799 (Activation: 0.1427)
Words that maximally activate this feature:
['}', '}', '>', 'coders', 'dtype']
└─ Low-Level Feature: 24 (Activation: 0.29527846)
  Words that maximally activate this low-level feature:
  ['}', 'coders', '}', 'legger', 'mec']

◆ Top-Level Feature 1473 (Activation: 0.1222)
Words that maximally activate this feature:
['\r', '2', '9', '=(', '3']
└─ Low-Level Feature: 14 (Activation: 0.14846489)
  Words that maximally activate this low-level feature:
  ['\r', '=', '4', '\r', '3']

◆ Top-Level Feature 1356 (Activation: 0.1189)
Words that maximally activate this feature:
['ú', 'zò', 'czego', 'futures', 'í']
└─ Low-Level Feature: 25 (Activation: 0.05206553)
  Words that maximally activate this low-level feature:
  ['Baseball', 'zò', 'बल्ल', 'UC', 'sacrament']

◆ Top-Level Feature 907 (Activation: 0.1167)
Words that maximally activate this feature:
['ullah', 'threatening']
└─ Low-Level Feature: 24 (Activation: 0.12262511)
  Words that maximally activate this low-level feature:
  ['threatening', 'ullah']
    
```

(b) No Whitening

Figure 18: The causal inner product is necessary for the model to learn meaningful features.

```

Explanations for word: 'python'
=====
◆ Top-Level Feature 547 (Activation: 0.0955)
Words that maximally activate this feature:
['pit', 'Pit', 'Pit', 'pit', 'pits']
└─ Low-Level Feature: 9 (Activation: 0.07605679)
  Words that maximally activate this low-level feature:
  ['py', 'Python', 'Python', 'python', 'python']

◆ Top-Level Feature 734 (Activation: 0.0839)
Words that maximally activate this feature:
['script', 'Script', 'scripts', 'Script', 'script']
└─ Low-Level Feature: 4 (Activation: 0.00004199)
  Words that maximally activate this low-level feature:
  ['Script']

◆ Top-Level Feature 51 (Activation: 0.0626)
Words that maximally activate this feature:
['snap', 'snaps', 'Sna', 'sna', 'snap']
└─ Low-Level Feature: 16 (Activation: -0.00001486)
  Words that maximally activate this low-level feature:
  ['Snap', 'snapping', 'SNA', 'Sni', 'sna']

◆ Top-Level Feature 1867 (Activation: 0.0624)
Words that maximally activate this feature:
['checkbox', 'DataBase', 'TextBox', 'mongodb', 'ToDo']
└─ Low-Level Feature: 19 (Activation: 0.06658750)
  Words that maximally activate this low-level feature:
  ['JavaScript', 'javascript', 'JAVA', 'Kotlin', 'TypeScript']

◆ Top-Level Feature 1316 (Activation: 0.0506)
Words that maximally activate this feature:
['ph', 'Ph', 'ph', 'PH', 'PH']
└─ Low-Level Feature: 17 (Activation: 0.00003440)
  Words that maximally activate this low-level feature:
  ['Philip', 'Phillip', 'Philip', 'Phillips', 'Phillip']
    
```

(a) Baseline

```

Explanations for word: 'python'
=====
◆ Top-Level Feature 1902 (Activation: 0.2456)
Words that maximally activate this feature:
['gennaio', 'giugno', 'dicembre', 'settembre', '/>\r']
└─ Low-Level Feature: 24 (Activation: 0.29844666)
  Words that maximally activate this low-level feature:
  ['みました', 'ške', 'へ', 'iology', '=/'"]

◆ Top-Level Feature 996 (Activation: 0.1957)
Words that maximally activate this feature:
['ayí', 'эм', 'm', 'んだろ', 'l']
└─ Low-Level Feature: 22 (Activation: 0.05954496)
  Words that maximally activate this low-level feature:
  ['cida', 'raps', 'dig', 'kenn', 'AEC']

◆ Top-Level Feature 696 (Activation: 0.1775)
Words that maximally activate this feature:
['的不同', '習', 'pm', 'b', 'firefox']
└─ Low-Level Feature: 20 (Activation: 0.17208365)
  Words that maximally activate this low-level feature:
  ['習', '習', 'vuelto', 'b', '的不同']

◆ Top-Level Feature 330 (Activation: 0.1550)
Words that maximally activate this feature:
['ó', 'uja', 'strlen', 'utm', 'vaí']
└─ Low-Level Feature: 15 (Activation: 0.25827643)
  Words that maximally activate this low-level feature:
  ['ców', 'categorical', 'чики', 'и', 'vaí']

◆ Top-Level Feature 145 (Activation: 0.1477)
Words that maximally activate this feature:
['.', 'дний', '場', '体が', '嶋']
└─ Low-Level Feature: 12 (Activation: 0.10202558)
  Words that maximally activate this low-level feature:
  ['.', 'K', 'ffed', 'gm', 'regresa']
    
```

(b) No Whitening

Figure 19: The causal inner product is necessary for the model to learn meaningful features.

C MORE EXAMPLE FEATURES

1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349

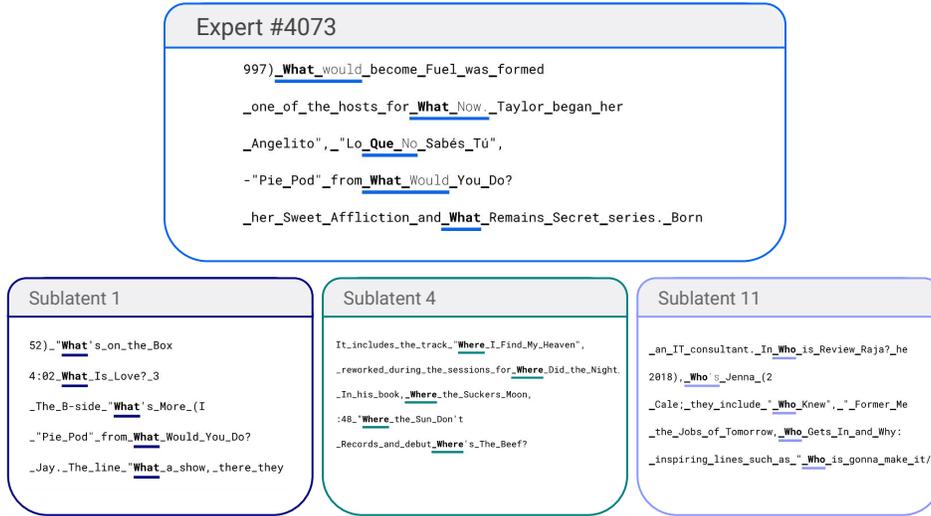
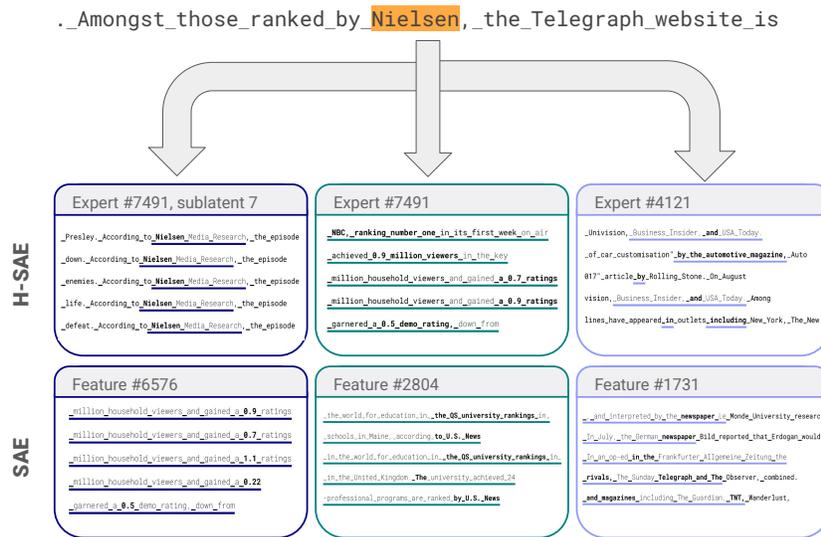


Figure 21: An example of a hierarchical feature from the 8K highlevel x 16 sublatent H-SAE model. A ‘question-word’ high level feature with “What”, “Where” and “Who” sublatents.



1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403

Figure 22: A comparison decomposition of the same context/embedding in the 8k x 16 H-SAE architecture and 8k standard SAE. Of note in the H-SAE is Expert #7491 which not only has the meaningful sublatent visualized here but also additional sublatents for various parts of writing about rating like demographics, specific ratings agencies like Nielsen, and mediums like TV episodes. Expert #4121 is a “magazines”/“news website” feature, promoting the following tokens heavily if added to the residual stream: `_Forbes`, `_Bloomberg`, `_CNN`, `_BBC`, `_Daily`, `_Newsweek`, `_Reuters`, `_Huffington`, `_magazine`, `_The`. We see a similar decomposition on the SAE, with a newspaper feature and “media ratings” feature. However, due to the lack of granular features, the specific ratings agency ‘Nielsen’ is not represented (rather the top-level ratings feature promotes the token ‘Nielsen’ more heavily in the SAE). The SAE uses a more generic “ratings”/“rankings” feature that shows signs of absorption with a “University Rankings” feature.

1404 D COMPARISONS WITH MATROYSHKA-SAE

1405

1406 D.1 SYNTHETIC BENCHMARK

1407

1408 The M-SAE paper introduces a synthetic benchmark that consists of 20 features organized hierarchi-
1409 cally into 11 “parent features” (which can co-activate) and 9 child features split evenly among 3 of the
1410 parents (so 8 “parent” features have no children). We adapt this to 28 overall, with 12 parents, and 16
1411 children split among 4 parents. Following the Matrysoka paper, we train a M-SAE with 28 features.
1412 We also train a 12x4 H-SAE (notice that this model is misspecified for the data because we allow
1413 all parents to have children). The M-SAE paper evaluates the reconstruction by cosine similarity
1414 between the optimal alignment of the learned and ground truth features. We find the M-SAE achieves
1415 a mean cosine similarity of 0.502 and the H-SAE 0.526 (despite the H-SAE being misspecified)

1416 D.2 UNEMBEDDING DECOMPOSITION

1417

1418 We train an Hierarchical-SAE and Matroyshka-SAE for 10,000 steps on the unembeddings trans-
1419 formed by the causal inner product. The H-SAE is a top-k=5, 2k x 32 model and the M-SAE is
1420 matched with top-k=10 and 5 groups out of 65k total features (i.e. both models have the same number
1421 of overall latents and can select the same number of latents per reconstruction). The features are
1422 fairly similar, with the H-SAE often finding sparser representations even within its top-k (by having
1423 near-zero activations on irrelevant features), finding high-level features the M-SAE does not, and
1424 fewer uninterpretable features. The lower maximum activation may also be an advantage of the
1425 H-SAE, as it spreads its activation weight out over more composable features. These factors also
1426 combine with the reconstruction loss and computational advantage the H-SAE has over both an
1427 M-SAE and standard SAE (e.g. the H-SAE computes 3% as many feature activations during the
1428 encoding stage as compared to the M-SAE). Comparison decompositions follow:

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

Decomposition of ' Chicago'					
Feature Num		Activation		Top Tokens	
H-SAE	M-SAE	H-SAE	M-SAE	H-SAE	M-SAE
1927	1368	0.4056	0.5899	[' Toronto', ' Chicago', ' Atlanta', ' Mumbai', ' Denver']	[' Chicago', 'Chicago', ' CHICAGO', ' chicago', 'chicago']
1927.5	737	1E-3	0.4034	[' Madrid', 'Madrid', ' Barcelona', 'Barcelona', ' Tucson']	[' Seattle', ' Atlanta', ' Detroit', ' Denver', ' Chicago']
859	44230	0.1282	0.0462	[' American', ' America', ' american', 'American', ' Americans']	['chicago', 'நஞ்சு', ' chicago', 'நந்த', 'Chicago']
859.5	37392	3E-4	0.0164	[' USA', 'USA', ' الولايات', ' EUA', ' CWA']	[' Missouri', ' Wisconsin', ' Indianapolis', ' Kansas', ' Tennessee']
175	15282	0.1170	0.0403	[' Texas', ' Alabama', ' Florida', ' Louisiana', 'Texas']	[' Earth', ' West', ' South', ' King', ' Bay']
175.19	3961	0.0522	0.0525	[' iowa', ' Iowa', ' Kansas', 'Iowa', 'Kansas']	[' Pennsylvania', ' Michigan', ' Illinois', ' Wisconsin', ' Ohio']
98	4981	0.1094	0.0329	[' Chrom', ' chrom', 'chrom', ' Chrome', 'Chrom']	[' NYC', ' NY', ' NYC', ' NY', ' nyc']
98.0	34993	-2E-4	0.0069	[' Chromosome', 'Chromosome', ' browser', ' thermo', ' nhiêm']	[' MEXICO', 'mexico', ' mexico', 'Mexico', ' Mexico']
1512	64996	0.0887	0.0067	[' il', 'il', ' Il', ' IL', 'IL']	['đi', ' giò', 'đi', 'Amore', ' '}}{\'}\']
1512.8	34031	1.6E-3	0.0066	[' ila', ' Illusion', ' ilo', 'ilation', 'ыл']	[' rör', ' haberse', ' haber', ' skydd', 'cc']

Decomposition of ' puppy'					
Feature Num		Activation		Top Tokens	
H-SAE	M-SAE	H-SAE	M-SAE	H-SAE	M-SAE
732	282	0.3964	0.6628	[' dog', ' Dog', 'Dog', 'dog', ' dogs']	[' puppy', ' Puppy', 'Puppy', 'puppy', ' puppies']
732.31	177	-0.0006	0.3691	[' barks', ' собаки', ' woof', ' собака', ' Labrador']	[' dog', ' dogs', ' Dog', 'dog', 'Dog']
956	27992	0.0883	0.0356	[' pig', ' goat', ' pigs', ' monkey', ' Goat']	[' brother', ' mother', ' father', ' sister', ' son']
956.12	3008	9E-5	0.0633	[' horse', 'Horse', ' Horse', ' pony', ' Pony']	[' infant', ' Infant', ' infants', ' baby', ' babies']
117	13729	0.0525	0.0361	[' boy', ' Boy', 'boy', 'Boy', ' BOY']	[' wood', ' fish', ' glass', ' milk', ' snow']
117.0	59925	0.2141	0.0068	[' menina', ' boya', ' ガール', 'BOYS', 'GIRLS']	['\uf075', '\uf075', ' zamanda', '玩笑', '\uf06e']
487	46772	0.0457	0.0067	[' pet', ' Pet', 'pet', 'Pet', ' PET']	['ယောဇ', 'မာဂျ', 'u', ' oxpa', 'ဖျီယု']
487.4	6188	1E-4	0.0264	[' PETER', ' Pedro', 'Pedro', 'petrol', ' Petrol']	[' Teddy', 'Teddy', 'teddy', ' teddy', ' Ted']
1809	36597	0.0437	0.0065	[' patient', ' shopper', ' attendee', ' subscriber', ' sufferer']	['ନ', 'ふる', 'ຈນ', 'inau', '獸']
1809.8	63546	9E-4	0.0065	[' Passenger', ' passenger', 'passenger', 'Passenger', ' guest']	['MDL', ' MDL', 'md1', 'NOV', 'MRP']

Decomposition of 'Queen'

Feature Num		Activation		Top Tokens	
H-SAE	M-SAE	H-SAE	M-SAE	H-SAE	M-SAE
951	99	0.5322	0.8691	[' King', ' king', 'King', ' kings', ' KING']	[' Queen', 'Queen', ' queen', ' QUEEN', ' queens']
951.24	2304	-9E-3	0.1364	[' crowns', 'könig', ' crowns', ' crown', ' roy']	[' Coach', ' Owner', ' Officer', ' Manager', ' Administrator']
1724	15282	0.2876	0.0816	[' lady', ' woman', ' Lady', 'lady', 'Lady']	[' Earth', ' West', ' South', ' King', ' Bay']
1724.15	2304	7E-4	0.0109	['girlfriend', ' girlfriend', ' Girlfriend', ' girlfriends', ' madam']	[' Staten', '諫', 'Axes', ' Rhode', 'Rhode']
1247	43216	0.2281	0.0076	[' Qu', ' qu', 'Qu', ' QU', 'qu']	['goers', ' Watcher', 'qc', 'QC', ' VIEWS']
1247.13	30719	1.3E-3	0.0147	['QUEUE', ' Que', 'Que', ' QUE', 'QUE']	['Jerusalem', ' Jerusalem', ' القدس', 'lande', ' Haifa']
1914	53291	0.1337	0.0073	[' Officer', ' Engineer', ' Professor', ' Trainer', ' Surgeon']	['미', ' 미', 'ㄹ', 'ㄹ', 'ㅅ']
1914.29	34322	4E-4	0.0073	[' Presidents', ' Blogger', ' 眞', ' Maestro', ' Appellant']	['寻求', ' recibido', ' kuitenkin', 'medal', '淘']
939	45011	0.0612	0.0073	[' chairman', ' Chairman', 'chairman', 'Chairman', ' CEO']	[' 伸', '伸', ' Bronson', '肘', 'レッド']
939.8	59557	5E-3	0.0071	[' President', ' president', 'President', 'president', ' PRESIDENT']	[' VTT', ' Norsk', ' Matti', ' formant', ' Marat']

Decomposition of 'Bayesian'

Feature Num		Activation		Top Tokens	
H-SAE	M-SAE	H-SAE	M-SAE	H-SAE	M-SAE
901	8866	0.1695	0.4217	[' Jacobian', ' Dirichlet', ' bilinear', ' piecewise', ' variational']	['Bayesian', ' Bayesian', ' Bayes', 'Bayes', ' bayonet']
901.32	9192	1E-3	0.1400	[' singularities', ' oscillatory', ' moduli', ' sinusoidal', ' trigonometric']	['Probability', ' Probability', 'probability', ' probability', ' 概率']
2010	2552	0.1214	0.2152	[' probability', ' probabilities', ' Probability', 'probability', ' Probab']	[' Dirichlet', ' Jacobian', ' Cauchy', ' Laplace', ' Poincaré']
2010.15	3286	1E-4	0.1930	[' prospects', 'Probable', ' probable', ' Probable', 'probable']	['ay', 'AY', ' Ay', ' ay', 'Ay']
870	858	0.0941	0.1309	[' Catholic', ' Hindu', 'Catholic', ' Muslim', ' Christian']	[' ba', ' Ba', 'Ba', 'ba', ' BA']
870.31	1395	-1E-4	0.686	[' Aryan', ' Huguen', ' Aryan', 'Gothic', ' Gregorian']	[' probability', ' likelihood', ' chances', ' probabilities', ' Probability']
1837	1367	0.0801	0.0593	[' ba', ' Ba', 'Ba', 'ba', ' BA']	[' Catholic', 'Catholic', ' Catholics', ' catholic', ' Muslim']
1837.30	5299	0.0780	0.0459	[' Bá', ' bay', ' Bay', 'Bay', ' BAY']	[' statistics', ' statistical', ' Statistics', 'Statistics', 'statistics']
404	57380	0.0781	0.0099	[' statistics', ' Statistics', ' statistical', ' statistic', ' stats']	['広場', ' priors', '样本', 'Bayesian', ' YC']
404.12	39412	4E-4	0.0085	[' thống', ' stat', 'Estad', ' statistics', ' Statistical']	[' epistem', ' methodological', ' Epis', ' EPISODE', ' evangel']