A Theory of Atomic Features and Four Testable Predictions

Anonymous Authors[1]

Abstract

Features serve as a core conceptual building block in the study of language models. We formalize the hypothesis that there exists an *atomic* set of features and consider if sparse autoencoders (SAEs) are capable of recovering these features. Starting from the theory, we derive testable hypotheses including a *stability principle*: that under the hypothesis, SAEs of increasing size recover a growing set of stable features (which we expect are the atomic features). We demonstrate that this principle holds in practice on SAEs trained on two large embedding models. We also find evidence for three other testable predictions: recovery of features at each level of a hierarchy, recovery of shared features using different training data distributions, and recovery of shared features across SAEs trained on different embedding models (evidence of platonicity). Our results suggest the expressivity and fidelity of a theory of atomic features. Practically, our results suggest that scaling SAEs can provide more granularity while retaining stable high-level features.

1. Introduction

Features are a foundational and ubiquitous concept in the empirical study and practice of language models. Features are typically understood as directions in the representation space of language models that correspond to different concepts. Features are used in many tasks: including to steer and interpret model behavior (Wang et al., 2025; Zou et al., 2023; Templeton et al., 2024; Lindsey et al., 2025; Li et al., 2025; Park et al., 2023).

While features are widely used, there is debate about the "nature" of features. This debate can be most clearly understood in terms of if one believes the *atomic hypothesis*: that there exists some unique fundamental set of *atomic* features. In this view, language model representations can be decomposed into atomic features in the same way that molecules can be decomposed into atoms.[1]

There is plenty of skepticism towards this view. One principle hurdle is that it isn't clear how to even verify if a given set of features is the atomic set of features (assuming such a set exists). Furthermore, our existing tools for extracting features appear to have limitations: for example, features learned by different sparse autoencoders tend to differ (Paulo and Belrose, 2025), challenging the idea that there is one unique set of atomic features. Therefore, an alternative view is that there is no set of atomic features, and that features (extracted from SAEs or by other means) are useful but imperfect approximations of more complicated structure within language models (Engels et al., 2024; Modell et al., 2025).

Despite the fundamental nature of this debate (both in terms of high-level theorization of language models and practical implications for extracting features), we lack the ability to test these contrasting views in a principled manner. As we have described, it is not clear if it is even possible to test the atomic hypothesis.

In the present work, we introduce a formal theory of atomic features and then derive testable predictions of the theory, using a mix of theoretical derivation and simulation. A major prediction of the theory is a **stability principle for dictionary learning**. The principle states that across SAEs of increasing dictionary size, we should see a stable set of features emerge, and that the features in this stable set are atomic.

1.1. Deriving Testable Predictions from the Theory

Our approach is to adopt a formal theory of atomic features and then to derive testable predictions from the theory. Under the theory, there exists a fundamental set of M atomic features such that language model representations can be decomposed as sparse linear combinations of them.

Both a theoretical result (Theorem 1) and simulated exper-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]While this view is not often explicitly stated, it is implicit in other popular theories such as the linear representation hypothesis (Park et al., 2023; Garg et al., 2026). (See (Smith, 2024) for a discussion.)

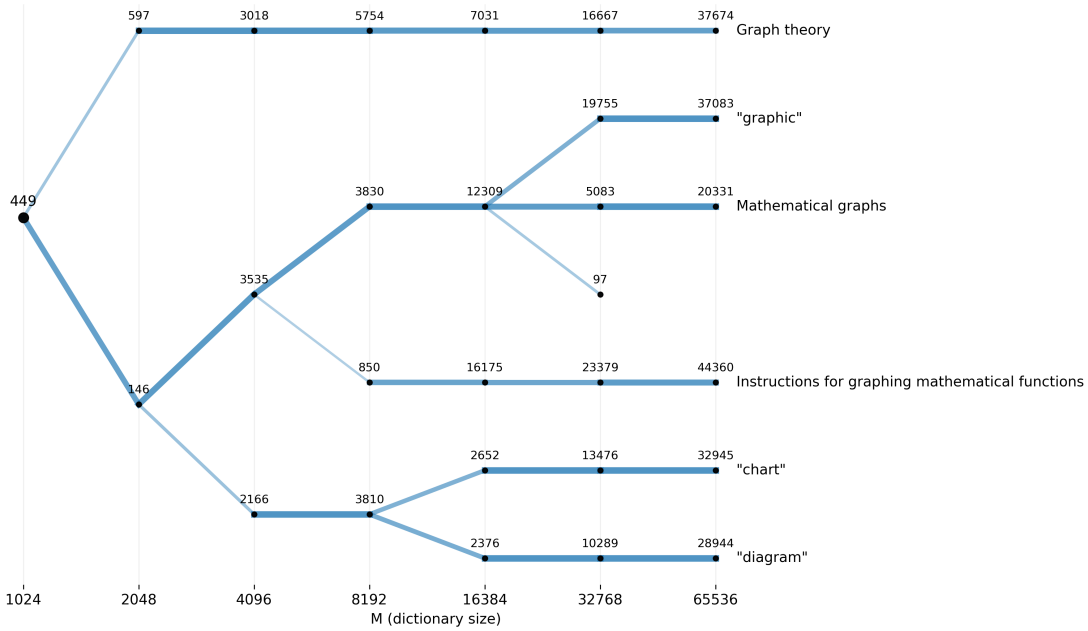


Figure 1. As dictionary size increases, features split into stable atomic components, as predicted by the atomic theory. Here, for example, the "Graph theory" feature is stable starting from D_{2048}. Notice also that "Graph theory" is a high-level concept, yet it remains stable (in Section 3, we show that the dictionaries also learn many subtopics of graph theory as features, such as Bipartite graphs, Kuratowski's theorem, and graph coloring). The feature at the root node, meanwhile, appears to be polysemantic (activating for different usages of the word "graph" as well as specific concepts in graph theory), suggesting that it is not atomic; indeed, it splits rather than remaining stable.

iments yield the following prediction: under the atomic theory, sparse dictionaries of size $m < M$ contain many of the most prevalent atomic features in the data distribution.

Notice that this prediction (a *recovery principle*) is not directly testable empirically. Given a dictionary of features, it is not clear how to verify which (if any) features are atomic. A main observation we make is that this prediction does however imply a testable property of *sequences* of dictionaries of increasing size: many features in a given dictionary should also appear in all larger dictionaries

This *stability principle* sharply contrasts with other methods for extracting features like k-means clustering, in which by increasing the number of features, the features change and become more and more granular.

We now state the prediction of stability, as well as three related testable predictions of interest. In Section 2.3, we show in greater detail how we arrive at these predictions from the theory.

Stability. Given a sequence of increasingly large learned dictionaries, many features in one dictionary are present in all larger dictionaries. In other words, past a certain dictionary size, atomic features will emerge and remain stable.

Hierarchy. Higher-level concepts in hierarchies do not split in larger dictionaries. While there is worry that larger SAEs will lose these high-level hierarchical concepts (Bussmann et al., 2025; Muchane et al., 2025; Costa et al., 2025), the atomic theory predicts that in larger SAEs, these should be retained. This prediction is of high practical relevance, since it implies that large SAEs can obtain features at all desired levels of coarseness.

Distribution Invariance. SAEs trained on different data distributions learn an overlapping set of atomic features: the intersection between the m most prevalent atomic features in each data distribution.

Platonicity. Further assuming the platonic representation hypothesis (Huh et al., 2024), the atomic theory predicts that dictionaries trained on different models will share features. This gives another way to test the platonic representation hypothesis.

1.2. Overview of Experiments

We now provide an overview of our empirical tests of the four testable predictions described above. In our experiments, we take two large embedding models (gemini-embedding-2 and llama-embed-nemotron-8b) and train SAEs to obtain feature dictionaries of size $m =$

$1024, 2048, 4096, 8192, 16384, 32768, 65536$. We use a data mix of 62.3 million examples from 12 datasets used to fine-tune the nemotron model.

We find evidence that supports each of the four testable predictions described above:

Stability. We show that many features in each dictionary also appear in all larger dictionaries (Figure 4). We find, e.g., that at a threshold of 0.7 used in prior work (Paulo and Belrose, 2025), 79.0% of features in the gemini dictionary of size 4096 are contained within all four larger dictionaries. Figure 1 gives a particular example of the emergence of stable features at larger dictionary size.

Hierarchy. We show in three case studies, on New York City geography, plants and animals, and subtopics in graph theory, that there exist atomic features that capture multiple levels of a hierarchical concepts, within a given SAE. For example, we find that larger SAEs retain features for New York City as a whole (in addition to more granular borough features), for animals as a whole (in addition to features for granular animal groups), and for the concept of graph theory as a whole (in addition to features for subtopics like graph coloring and Dijkstra's algorithm).

Distribution invariance. We consider two splits of our data mix (one that splits datasets using Wikipedia from those that do not; one that is a more random split of datasets) and compare SAEs trained on each split. We find that many features are shared, and that these shared features are also more likely to be atomic (as measured using a stability proxy).

Platonicity. Comparing the activation patterns of SAEs trained using gemini and nemotron, we find that many features are highly correlated. These shared (i.e., platonic) features are also more likely to be atomic (as measured using a stability proxy).

1.3. Discussion

Overall, our empirical results are highly-consistent with the theoretical predictions we derived from a theory of atomic features.

Contribution to the theory of language models. On the theoretical side, our results provide evidence for an "atomic representation hypothesis" and that SAEs are capable of distilling these atomic features. At a higher level, our approach suggests promise in taking a traditional scientific approach of adopting ambitious theories and deriving and testing predictions from the theory.

Contribution to the practical use of SAEs. Practically, our results imply that scaling dictionary size can produce more granularity in features while simultaneously preserving high-level features. Moreover, increasing dictionary size should result in more monosemantic and "fundamental" features. This also highlights a practical divide between SAEs and clustering methods as approaches to extract interpretable features.

2. Deriving Testable Predictions of the Atomic Theory

We assume the theory of atomic features as our premise, and then analyze it to derive testable predictions. In particular, we analyze the behavior of sparse autoencoders optimized on data generated according to the atomic theory. In this section, we do so theoretically and in simulation.

Formalizing the atomic theory. In the atomic theory, there exists some set of *atomic* features $a_1, \cdots, a_M \in \mathbb{R}^d$ such that language model representations $x \in \mathbb{R}^d$ can be uniquely decomposed as

$$x \approx \sum_{i=1}^{M} z_i a_i, \tag{1}$$

where typically we assume $z \in \mathbb{R}^M$ is sparse (allowing for M to be much larger than d).

In Section 2.1, we give a theoretical result that establishes a strong property of SAEs under the atomic theory: that under pairwise independence assumptions on features, SAEs exactly recover the most prevalent atomic features under the data distribution. In Section 2.2, we then consider richer synthetic data settings, beyond pairwise independence. We observe the same pattern: SAEs learn many of the most prevalent atomic features. Then, in Section 2.3, we show how this observation leads to testable predictions given real data, which we test in the following section.

2.1. Exact Atom Recovery from Dictionary Learning

Consider a set of atomic features consisting of orthonormal vectors $a_1, \cdots, a_M \in \mathbb{R}^M$. We consider a simple class of data generating processes from this "atomic basis." This set of data generating processes are parameterized by a distribution $\mathcal{D}$ over supports $S \subseteq [M]$ of size k (i.e., data is k-sparse in the atoms). Then we consider samples of the form

$$x = \sum_{j \in S} z_j a_j,$$

where $S \subseteq [M]$ is a random support of size k and such that conditional on S, the active coefficients $\{z_j : j \in S\}$ are pairwise independent. Finally, we assume a set of regularity

conditions: the active coefficients satisfy $\mathbb{E}[z_j \mid S] = 0$ and $\mathbb{E}[z_j^2 \mid S] = 1$ for every $j \in S$; and for every $T \subseteq [M]$ with $|T| = k$, one has $\Pr(S = T) > 0$, and conditional on $S = T$ and the coefficient vector $(z_j)_{j \in T}$ has a density that is positive on some nonempty open subset of $\mathbb{R}^k$ (the atomic features co-occur). Of these assumptions, that of zero-mean pairwise independence conditional on support is the relatively strong one; it is shared with Arora et al. (2015), and we relax it in simulation.

For a dictionary $B = [b_1, \ldots, b_m] \in \mathbb{R}^{d \times m}$ with unit-norm columns, the dictionary loss is

$$\mathcal{L}(B) := \mathbb{E}_{x \sim \mathcal{D}} \left[\min_{\|z\|_0 \leq k} \|x - Bz\|_2^2 \right].$$

Theorem 2.1 (Recovery of most frequent atoms)**.** *Write $p_j := \Pr(j \in S)$ for the marginal frequency of atom a_j. Index the features such that $p_1 \geq p_2 \geq \cdots \geq p_M$ and assume that $p_m > p_{m+1}$. Then every global minimizer $B^\star = [b_1^\star, \ldots, b_m^\star]$ of $\mathcal{L}(B)$ satisfies*

$$\{b_1^\star, \ldots, b_m^\star\} = \{\pm a_1, \ldots, \pm a_m\}.$$

That is, the m most frequent atoms are recovered up to permutation and sign.

The result predicts that dictionaries of size $m < M$ will recover the most prevalent atomic features. We give the proof in Section B.

The theorem makes several assumptions that deviate from practice: first, it requires that conditional on support, active coefficients are independent and zero-mean (this is unrealistic particularly because some atomic features are likely to be correlated with each other); second, it assumes that the true dictionary consists of entirely orthogonal vectors, whereas it is generally hypothesized that feature vectors exist in superposition (i.e., there are more feature vectors than dimensions). In the simulations in the section below, we test distributions that break each of these assumptions.

2.2. Studying the Theory in Simulation

Again, we will consider samples of the form $x = \sum_{j \in S} z_j a_j$, where $S \subseteq [M]$ is a random support. However, we will now allow the set of atomic features $a_1, a_2, \cdots, a_M$ to be uniformly unit vectors in $\mathbb{R}^d$. We will also consider more realistic distributions of the feature value z_j conditional on j being in the support S. We consider two instantiations of the data distribution:

Simulation Setup (a): Independent Atomic Features. In the first setup, features in the support are chosen independently proportional to a power law: $\Pr[j \in S] \propto j^{-\alpha}$. We sample such that $|S| = k$ is fixed. Conditional on $j \in S$, we let z_j be sampled independently according to the Gaussian distribution $\mathcal{N}(\mu, \sigma)$. We set $\alpha = 0.5, \mu = 1, \sigma = 0.25$ and $d = 128, M = 64, k = 4$.

Simulation Setup (b): Hierarchical Correlated Atomic Features. Now, there are a set of $M/2$ parent features, each of which are associated with a child (totalling of M features). We first choose a random subset S_1 of four parents, chosen independently as above according to the power law. For each parent in the support, we add the child feature into S_2 with probability 0.5 and set $S = S_1 \cup S_2$. Then, conditional on $j \in S$, we let z_j be sampled independently according to the Gaussian distribution $\mathcal{N}(\mu, \sigma)$. Again we set $\alpha = 0.5, \mu = 1, \sigma = 0.25$ and $d = 128, M = 64, k = 4$.

In both cases, we train SAEs with $m = 4, 8, 16, 32$. In (a) we set $k = 4$ and in (b) we set $k = 6$ (the expected sizes of the supports). We use batch size 1024 and auxiliary coefficient $1/16$. Details about the SAE architecture and training procedure are given in Section A.

In Figure 2, we then plot the dot product between learned features (decoder directions) (x-axis) and the true atomic features (y-axis). In (a), rows (true atomic features) are sorted according to prevalence of the feature. In (b), rows corresponding to parent and child features are adjacent and then sorted according to the prevalence of the parent feature. We sort columns for visual comprehension. These results show that large proportion of learned features recover atomic features, and that these atomic features tend to be the most prevalent features. (b) also shows a more specific pattern: parent and child features tend to be mixed together in smaller dictionaries (notice columns that are active on two adjacent rows), but then separate into the individual atomic features in the larger dictionaries. We also test the second setup where we instead set $M = 256$, keeping all other parameters the same, testing a setting where $M > d$ (i.e., when there is superposition). We plot the corresponding results in Figure 6, which shows patterns that are nearly identical to (b).

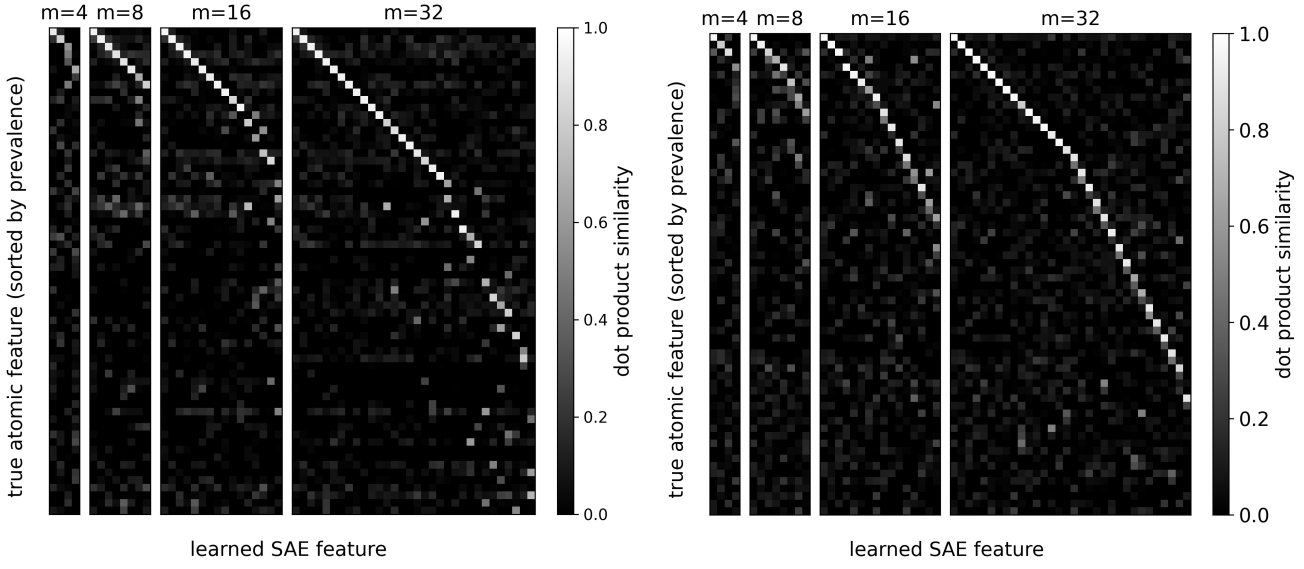
2.3. Testable Predictions

Both the theoretical result and simulations point to the following prediction about dictionary learning:

> (P1: Recovery) Dictionaries of $m < M$ features will recover many of the most prevalent atomic features.

However, notice that this prediction is not testable given any single learned dictionary; there is not an obvious way to tell which (if any) of the learned features are atomic. However, it does imply a testable prediction about sequences of dictionaries of increasing size:

> (P2: Stability) Many features recovered in a dictio-



(a) Independent Atomic Features

(b) Hierarchical Correlated Atomic Features

Figure 2. Two simulations of dictionary learning by SAEs under the atomic theory. Plots give dot product between learned features (x-axis) and the true atomic features (y-axis, sorted by prevalence). We see two regularities: learned dictionaries tend to recover many atomic features, and these typically are the most prevalent atomic features, while less prevalent features may be learned as mixtures. These patterns occur even outside the assumptions of Theorem 2.1. This suggests a testable prediction: that many features in smaller dictionaries also appear in larger dictionaries (the stability principle).

nary will also be present in all larger dictionaries.

We call this the *stability principle*. The stability principle implies that features learned by dictionaries of increasing size do not split indefinitely. Then, if we believe that each concept in a hierarchy is atomic (one reason, intuitively, being that it is more efficient to represent a parent concept as a single atom rather than as a union of many child atoms), this suggests the following further prediction:

> (P3: Hierarchy) Features representing top-level concepts in hierarchies persist in larger dictionaries.

Such a prediction pushes back on prevailing wisdom that features corresponding to higher-level nodes necessarily split in larger dictionaries (Muchane et al., 2025; Bussmann et al., 2025; Costa et al., 2025). Practically, this implies that larger dictionaries can recover features at all desired levels of coarseness (in contrast to clustering). P1 also yields a prediction about dictionaries trained on different data distributions:

> (P4: Distribution Invariance) Dictionaries trained on different data distributions contain shared features comprising the atomic features that are prevalent in both data distributions.

Finally, taking P1 together with the Platonic representation hypothesis suggests the following:

> (P5: Platonicity) Dictionaries trained on different models contain many shared features.

To summarize, from the atomic theory, we derived P1 from theoretical analysis and simulation. While P1 itself is not directly testable, it yields several testable (and interesting) predictions, P2-5.

3. Experiments

We take two large embedding models, gemini-embedding-2 and llama-embed-nemotron-8b, which have dimension $d = 3072, 4096$ respectively. We train top-k SAEs with latent dimension $m = 1024, 2048, 4096, 8192, 16384, 32768, 65536$ and with $k = 16, 16, 32, 32, 32, 64, 128$ respectively. The data mix consists of 12 datasets used to train llama-embed-nemotron-8b, totalling 62.3 million examples. We train each of these SAEs on 1 million batches of size 2048. At the given choices of sparsity k, this led to essentially 0 dead neurons by the end of training when using an auxiliary training loss coefficient of 0.25. Section A contains details about the data and SAE training procedure.

Using this setup, we test the four predictions made in the previous section. First, we establish a couple useful definitions. We refer to the learned dictionaries from SAEs of

Figure 3. Larger gemini SAEs learn stable hierarchical features of NYC geography: New York City (as a whole), Brooklyn, the Bronx, Queens, Staten Island, and outer boroughs (i.e., not Manhattan). Value in parentheses give dot product to the D_{65536} feature.

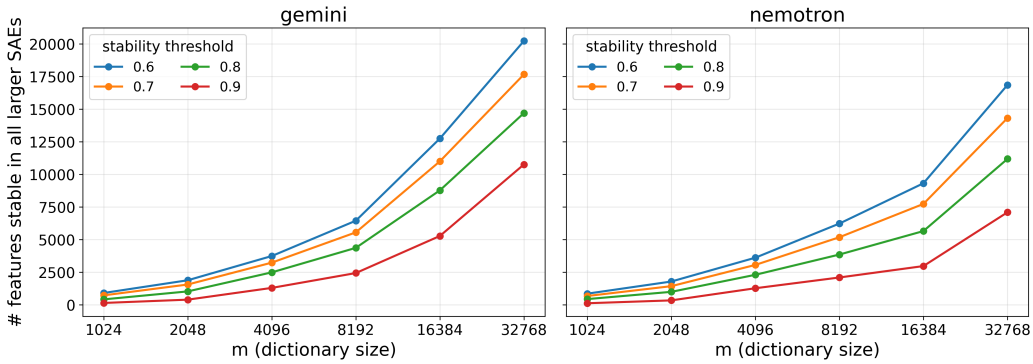


Figure 4. Empirically, many features in a dictionary appear in all larger SAEs (up to 65536), demonstrating stability.

size m as D_m, the set of m decoder directions in $\mathbb{R}^d$. The underlying model (gemini or nemotron) will be clear from context. We say that a feature $b \in \mathbb{R}^d$ appears in D_m at threshold τ if there exists $b' \in D_m$ such that $\langle b, b' \rangle > \tau$.

P2: Stability

Here we test whether many features in smaller dictionaries occur in all larger dictionaries. Figure 4 plots the number of features in D_m that appear in all larger dictionaries at a given threshold. For example, for the gemini models, at threshold 0.7, we find that $3234/4096 = 79.0\%$ of features in D_{4096} appear in all of $D_{8192}, D_{16384}, D_{32768}$, and D_{65536}. For the nemotron models, $3058/4096 = 74.7\%$ of features in D_{4096} appear in all larger dictionaries. Overall, many features in each dictionary appear in all larger dictionaries (even at a threshold of 0.9), supporting P2.

P3: Hierarchy

We now test if high-level hierarchical concepts persist in large dictionaries. We consider three case studies in which we expect there to be hierarchical structure, and study whether or not different nodes in the hierarchy are represented by atomic features in the SAEs we train. In each case, we find the presence of atomic features that represent high-level parent concepts.

Plants and animals. We take a list of plants and animals (broken up into birds, insects, fish, mammals, reptiles, amphibians) used by Park et al. (2024). We find features in D_{65536} in gemini that cleanly identifies each of these categories. Each category has a feature that classifies it with AUC ¿ 0.79 and in most cases much higher (Table 1). We also find that all such features appear in all smaller dictionaries aside from D_{1024} at threshold 0.8. Importantly, we find that there are separate atomic features for the parent concept of animal and for the child concepts of birds, insects, fish, mammals, reptiles, and amphibians.

category	neuron id	auc
mammal	49088	0.9684
bird	59640	0.9932
reptile	20588	0.7958
fish	44918	0.9463
amphibian	56211	0.8609
insect	36807	0.9853
plant	26399	1.0000
animal	37673	0.9990

Table 1. Plant and animal categories have clean features in D_{65536}. The top-level animal and plant features each persist in the dictionary (i.e., did not split).

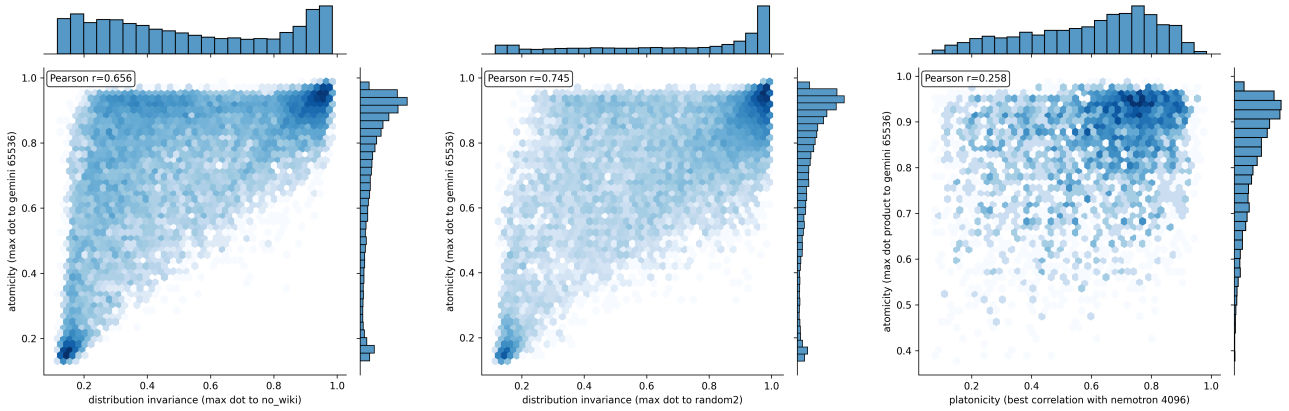
New York City geography. Figure 3 shows maps of New York City broken into neighborhoods. Each neighborhood is colored by the activation of the feature on the neighborhood name (e.g., "Nolita," "Astoria," "Long Island City"). We find that a high-level New York City neuron appears in all dictionaries starting from D_{4096} in gemini at threshold 0.94, suggesting that it is atomic. Furthermore, atomic child features also emerge: including for Queens, Brooklyn, Staten Island, the Bronx, as well as a child feature for the outer boroughs (everything except Manhattan).

Graph theory and its subtopics. We find a Graph Theory feature that appears in all dictionaries starting at D_{2048} in gemini at threshold 0.8 (Figure 1). We also find many child features (features that fire at least 40% of the time when the parent feature fires), including: Bipartite graphs, Planar graphs, Directed graphs, Dijkstra's algorithm, regular graphs, Graph planarity and Kuratowski's Theorem, "edge", Graph connectivity, "Hamiltonian", Number of edges in a graph, Graph degree sequences, Minimum Spanning Trees, Graph theory vertex degrees and edges, and graph coloring.

P4: Distribution Invariance

We now test if SAEs trained on different data distributions recover shared features. A prediction of Theorem 1 is that SAEs trained on different data distributions learn an overlapping set of atomic features: the intersection between the m most prevalent atomic features in each data distribution.

We test this hypothesis on two different data splits: first, we



(a) wiki v. no wiki data split *(b)* random1 v. random 2 data split *(c)* gemini v. nemotron

Figure 5. Comparisons between features derived from different data distributions and from different models. (a) and (b) show that there are many features that are distribution invariant (learned by SAEs with different data), and that these features are more atomic. (c) shows that many features are platonic (learned by SAEs trained on different models) and that these features are also more atomic.

split our underlying datasets between those that include Wikipedia and those that don't include Wikipedia; second, we take a random split (where both include parts of Wikipedia as underlying data). Therefore, we expect the second pair of data distributions to be more similar. We then train SAEs on gemini embeddings to obtain dictionaries of size 16384 on each of these data splits. (Section A gives exact data used in each split.)

Figure 5(a) and (b) plots on the x-axis a measure of distribution invariance (the maximum dot product between each feature in one dictionary to a feature in the other dictionary) and on the y-axis a measure of atomicity (the maximum dot product to D_{65536} in gemini, measuring if the feature remains in larger dictionaries). In both cases, we find that (1) many features are distribution invariant (i.e., appear in the other dictionary at a high threshold) and (2) features that are distribution invariant are also more likely to be atomic. This supports P4. We also note that there is greater distribution invariance in (b), where the data splits have more overlap (due to shared use of Wikipedia). This is consistent with the prediction P4, which also states that the overlapping set of atomic features recovered are those that are prevalent in both data distributions.

P5: Platonicity

We now test whether dictionaries learned from different embedding models recover the same atomic features, studying the platonic representation hypothesis (Huh et al., 2024). To do this, we take D_{4096} for both gemini and nemotron and compute activations for each feature in both dictionaries on a sample of 114,126 texts from our data mix. We restrict our analysis to features that are active on at least 50 texts (3991 and 4004 such features, respectively). Then, for each feature in the gemini dictionary, we compute the feature in the nemotron dictionary with the highest Pearson correlation in the activation pattern. Figure 5(c) then plots this maximum Pearson correlation on the x-axis and a measure of atomicity on the y-axis given by the maximum dot product with D_{65536} for gemini (this gives a measure of stability of the feature). Examples of features that are shared by both dictionaries include "Polish villages," "Baseball," "Dream interpretations," "Pokemon," "Eurovision Song Contest," and "Sri Lanka." A larger list is given in Table 2. We find that (1) many gemini features have a highly-correlated match in the nemotron dictionary, and (2) these features tend to also be more atomic. This provides some preliminary evidence that atomic features may also be platonic—that is, shared across different models.

4. Conclusion

In this paper, we formalized a theory of atomic features, and show how the theory can be used to derive testable predictions. We then provide empirical evidence in support of these predictions, using 14 SAEs trained on two large embedding models. Therefore, we demonstrate the promise of the atomic theory as a way to understand language models. Our results suggest scaling SAEs and identifying stable features as a concrete approach to identifying atomic features. Furthermore, scaling SAEs yields features at multiple levels of granularity (i.e., at different levels of a hierarchy), putting it in contrast with other unsupervised methods like clustering as an approach to extract interpretable features.

References

S. Arora, R. Ge, T. Ma, and A. Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Conference on learning*

theory, pages 113–149. PMLR, 2015.

V. Boteva, D. Gholipour, A. Sokolov, and S. Riezler. A full-text learning to rank dataset for medical information retrieval. 2016. URL http://www.cl.uni-heidelberg.de/~riezler/publications/papers/ECIR2016.pdf.

B. Bussmann, N. Nabeshima, A. Karvonen, and N. Nanda. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld. Specter: Document-level representation learning using citation-informed transformers. In *ACL*, 2020.

V. Costa, T. Fel, E. S. Lubana, B. Tolooshams, and D. Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. *arXiv preprint arXiv:2506.03093*, 2025.

J. Engels, E. J. Michaud, I. Liao, W. Gurnee, and M. Tegmark. Not all language model features are one-dimensionally linear. *arXiv preprint arXiv:2405.14860*, 2024.

N. Garg, J. Kleinberg, and K. Peng. How many features can a language model store under the linear representation hypothesis? *arXiv preprint arXiv:2602.11246*, 2026.

M. Huh, B. Cheung, T. Wang, and P. Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, art. arXiv:1705.03551, 2017.

D. Khashabi, A. Ng, T. Khot, A. Sabharwal, H. Hajishirzi, and C. Callison-Burch. Gooaq: Open question answering with diverse answer types. *arXiv preprint*, 2021.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

P. Lewis, Y. Wu, L. Liu, P. Minervini, H. Küttler, A. Piktus, P. Stenetorp, and S. Riedel. Paq: 65 million probably-asked questions and what you can do with them. 2021.

B. Z. Li, Z. C. Guo, and J. Andreas. (how) do language models track state? *arXiv preprint arXiv:2503.02854*, 2025.

J. Lindsey, W. Gurnee, E. Ameisen, B. Chen, A. Pearce, N. L. Turner, C. Citro, D. Abrahams, S. Carter, B. Hosmer, J. Marcus, M. Sklar, A. Templeton, T. Bricken, C. McDougall, H. Cunningham, T. Henighan, A. Jermyn, A. Jones, A. Persic, Z. Qi, T. B. Thompson, S. Zimmerman, K. Rivoire, T. Conerly, C. Olah, and J. Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

A. Modell, P. Rubin-Delanchy, and N. Whiteley. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*, 2025.

M. Muchane, S. Richardson, K. Park, and V. Veitch. Incorporating hierarchical semantics in sparse autoencoder architectures. *arXiv preprint arXiv:2506.01197*, 2025.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016. URL http://arxiv.org/abs/1611.09268.

K. Park, Y. J. Choe, and V. Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

K. Park, Y. J. Choe, Y. Jiang, and V. Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024.

G. Paulo and N. Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In J. Su, K. Duh, and X. Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen. CARER: Contextualized affect representations for emotion recognition. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1404. URL https://aclanthology.org/D18-1404.

L. Smith. The 'strong' feature hypothesis could be wrong. *AI Alignment Forum*, 2024.

A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. transformer circuits thread, 2024.

N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018.

M. Wang, T. D. la Tour, O. Watkins, A. Makelov, R. A. Chi, S. Miserendino, J. Heidecke, T. Patwardhan, and D. Mossing. Persona features control emergent misalignment. *arXiv preprint arXiv:2506.19823*, 2025.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of*

the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2369–2380, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259.

X. Yue, T. Zheng, G. Zhang, and W. Chen. Mammoth2: Scaling instructions from the web. *Advances in Neural Information Processing Systems*, 2024.

A. Zou, L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, A.-K. Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A. Data and SAE training details

Dataset details. Total number of rows: 62,255,407

- emotion (https://huggingface.co/datasets/mteb/emotion) (Saravia et al., 2018) English Twitter filtered for emotions, 19,930
- fever (https://huggingface.co/datasets/BeIR/fever) (Thorne et al., 2018; Thakur et al., 2021) Wikipedia, 11,079,420
- gooaq (https://huggingface.co/datasets/sentence-transformers/gooaq) (Khashabi et al., 2021) Google searches (from autocomplete), 6,024,992
- hotpotqa (https://huggingface.co/datasets/mteb/hotpotqa) (Yang et al., 2018) crowdworker questions + answers from Wikipedia, 10,564,510
- msmarco (https://huggingface.co/datasets/mteb/msmarco) (Nguyen et al., 2016) bing questions + results, 9,351,785
- natural-questions (https://huggingface.co/datasets/sentence-transformers/natural-questions) google search questions, Wikipedia answers (Kwiatkowski et al., 2019), 200,462
- nfcorpus (https://huggingface.co/datasets/mteb/nfcorpus) Queries from nutritionfacts.org and responses extracted from medical documents from PubMed (Boteva et al., 2016), 10,503
- paq (https://huggingface.co/datasets/sentence-transformers/paq) (Lewis et al., 2021) Wikipedia answers and generated questions using squad, natural questions, triviaqa), 20,000,000 (we take a sample of 10M question-answer pairs)
- scifact (https://huggingface.co/datasets/mteb/scifact) (Cohan et al., 2020) scientific claims and supporting paper titles/abstracts, 11,475
- squad (https://huggingface.co/datasets/sentence-transformers/squad) (Rajpurkar et al., 2016) crowdworker-posed questions and wikipedia answers, 175,198
- trivia-qa (https://huggingface.co/datasets/sentence-transformers/trivia-qa) (Joshi et al., 2017) questions and answers from 14 trivia and quiz-league websites, as

well as supporting evidence from web search results and Wikipedia articles, 146,692

- WebInstructSub (https://huggingface.co/datasets/TIGER-Lab/WebInstructSub?library=datasets) web crawl data for math/science/engineering (mostly from stackexchange and socratic) (Yue et al., 2024), 4,670,440

Of these, fever, hotpotqa, natural-questions, paq, squad, and trivia-qa use Wikipedia. These total 42,166,282. So the others have 62,255,407 - 42,166,282 = 20,089,125.

SAE details. A top-k sparse autoencoder takes an input x and computes

$$z = \mathrm{ReLU}(\mathrm{TopK}(W_{\mathrm{enc}}(x - b_{\mathrm{pre}}) + b_{\mathrm{enc}})), \quad (2)$$

$$\hat{x} = W_{\mathrm{dec}} z + b_{\mathrm{dec}}, \quad (3)$$

where $b_{\mathrm{pre}} \in \mathbb{R}^d, W_{\mathrm{enc}} \in \mathbb{R}^{m \times d}, b_{\mathrm{enc}} \in \mathbb{R}^m, W_{\mathrm{dec}} \in \mathbb{R}^{d \times m}, b_{\mathrm{dec}} \in \mathbb{R}^d$, and TopK sets all activations except the top k to zero. The loss on one input is

$$\mathcal{L} = ||x - \hat{x}||_2^2. \quad (4)$$

We also use an auxiliary loss to avoid dead neurons. The auxiliary loss term provides nonzero gradients for up to k_{aux} neurons which were not active in the last several forward passes. We fit these neurons to the residual reconstruction error left by the TopK neurons. That is, let z^{AuxK} consist of the activations of the top k_{aux} dead neurons in z; then

$$\mathcal{L}_{\mathrm{aux}} = ||W_{\mathrm{dec}} z^{\mathrm{AuxK}} - (x - \hat{x}_i)||_2^2,$$

and the full loss, then, is $\mathcal{L} = \mathcal{L}_{\mathrm{TopK}} + w_{\mathrm{aux}} \mathcal{L}_{\mathrm{AuxK}}$.

In our experiments we set $w_{\mathrm{aux}} = 0.25$ throughout. We use the top-k implementation from https://github.com/bartbussmann/BatchTopK/tree/main with otherwise default parameters.

Compute details. Each SAE was trained on an NVIDIA A100 (80GB) or H100. In all main experiment SAEs, we trained the model for 1M batches of size 2048. For synthetic models, we trained on 1M examples in batches of 1024. In both cases, neurons were considered to be dead if they did not activate for 512 batches.

Data splits for distribution invariance experiments.

- **random 1**: emotion, fever, gooaq, natural-questions, paq, scifact
- **random 2**: hotpotqa, msmarco, nfcorpus, squad, trivia-qa, WebInstructSub
- **wiki**: fever, hotpotqa, natural-questions, paq, squad, trivia-qa
- **no wiki**: emotion, gooaq, msmarco, nfcorpus, scifact, WebInstructSub

Code. We provide scripts for running our main analysis at this anonymous repo: https://anonymous.4open.science/r/atomic_features_neurips-4AF6/README.md. We plan on releasing the full underlying data and models needed to run the scripts.

B. Proof of Theorem 1

We first show that $\mathrm{span}(\{b_1^*, b_2^*, \cdots, b_m^*\}) = \mathrm{span}(\{a_1, a_2, \cdots, a_m\})$. Let P_U be the projection operator onto the subspace U. Then we have that

$$\mathcal{L}(B) \geq \mathbb{E}_{x \sim \mathcal{D}} \left[\|x - P_B x\|_2^2 \right]. \quad (5)$$

We first show that

$$\mathbb{E}_{x \sim \mathcal{D}} \left[\|x - P_U x\|_2^2 \right] \quad (6)$$

is strictly minimized by taking U to be $A_m := \mathrm{span}(\{a_1, a_2, \cdots, a_m\})$. This follows directly from the SVD argument, since the coefficients z_i are zero-mean, unit variance, and pairwise independent. Then defining $\mathcal{L} = \mathbb{E}_{x \sim \mathcal{D}} \left[\|x - P_{A_m} x\|_2^2 \right]$, we see that we can achieve this loss from the dictionary $\{a_1, a_2, \cdots, a_m\}$. It now remains to show that any optimal dictionary must consist of these specific vectors up to sign.

Now that we know that $\mathrm{span}(\{b_1^*, b_2^*, \cdots, b_m^*\}) = \mathrm{span}(\{a_1, a_2, \cdots, a_m\})$, we show that B^* contains exactly the directions $a_1, a_2, \cdots, a_m$ (up to sign). Define $R_{B^*} = \{B^* z : \|z\|_0 \leq k\}$. First observe that $P_{A_m} x \in R_{B^*}$ almost surely, otherwise the loss would exceed $\mathcal{L}^*$. Now write

$$R_{B^*} = \bigcup_{\substack{J \subseteq [m] \\ |J| \leq k}} F_J, \quad (7)$$

where we set $F_J := \mathrm{span}(\{b_j^* : j \in J\})$. Also define $E_T := \mathrm{span}(\{a_t : t \in T\})$. Then for every $T \subseteq [m]$ such that $|T| = k$, we have that x has positive density on an open subset of E_T. Therefore, we have that $E_T \cap F_J$ must have dimension k for at least one choice of J, since otherwise $E_T \cap R_{B^*}$ has measure 0, contradicting the fact that $P_{A_m} x \in R_{B^*}$ almost surely. This implies that for each $T \subseteq [m]$ with $|T| = k$, there exists $J \subseteq [m]$ with $|J| = k$ such that $E_T = F_J$. Now for each $i \in [m]$, observe that

$$\mathrm{span}(\{b_i^*\}) = \bigcap_{\substack{J \subseteq [m] \\ |J| = k, i \in J}} F_J \quad (8)$$

is a one-dimensional subspace that is the intersection of subspaces E_T. Therefore, $\mathrm{span}(\{b_i^*\}) = \mathrm{span}(\{a_j\})$ for some choice of $j \in [m]$. Since $b_1^*, b_2^*, \cdots, b_m^*$ span a subset of dimension m, this implies that each b_i^* is equal to a unique choice of a_j up to sign, showing the result.

C. Additional Figures

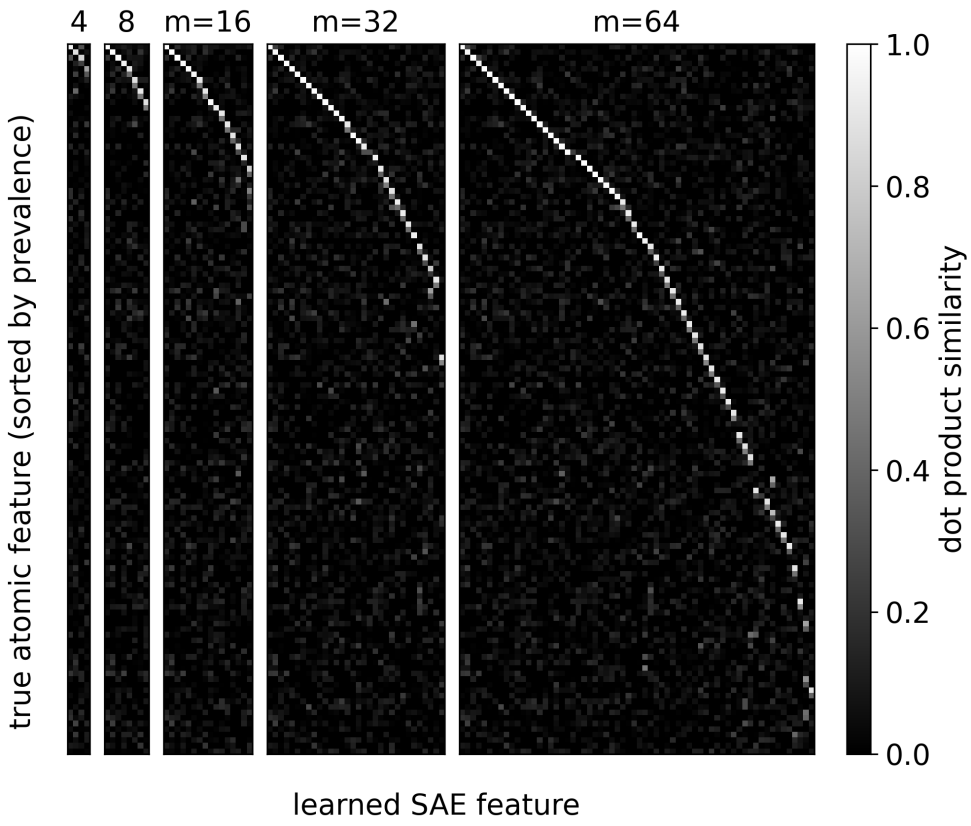


Figure 6. Synthetic superposition

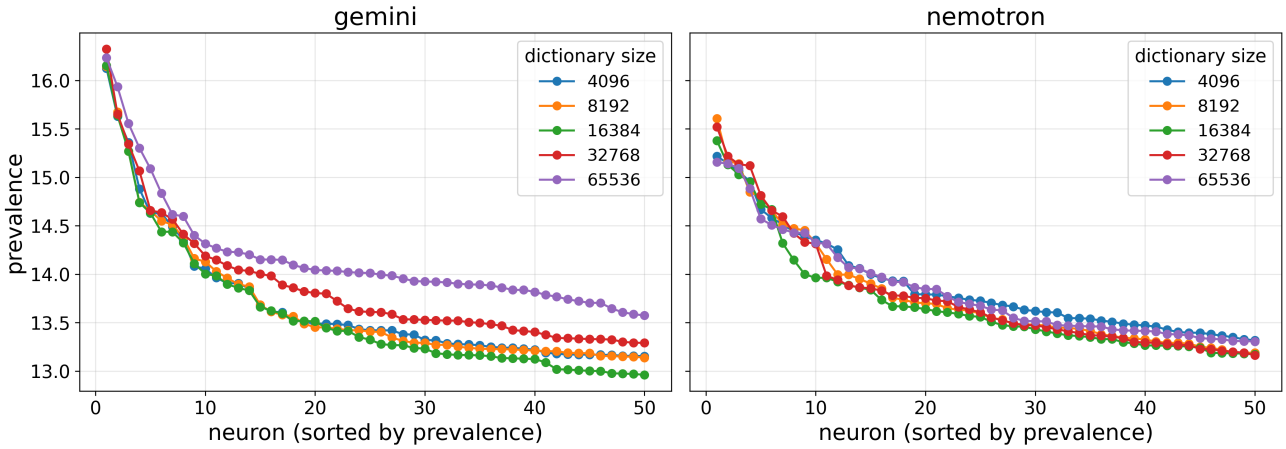


Figure 7. Large SAEs retain large features. Here, prevalence is defined by $\log(\#\text{ texts in data mix with activation} > 0.1)$.

gem. id	nemo id	corr.	gemini neuron	nemotron neuron
2434	1706	0.9848	Beginning of sequence	BOS token
1924	2058	0.9765	Iranian villages	Iranian village descriptions (Wikipedia format)
3290	1618	0.9715	Beetles in the family Cerambycidae	Cerambycidae
1405	561	0.9581	Polish village descriptions	Wikipedia descriptions of Polish villages
3957	1850	0.9562	Cricket	cricket
3239	2103	0.9556	Rugby	Rugby union
1827	3577	0.9549	morning-after pill	Emergency contraception (the morning-after pill)
1588	2776	0.9543	moths	Moths
393	2055	0.9538	Tennis tournaments and organizations	tennis
1104	666	0.9526	Pokémon	Pokémon
1568	1750	0.9525	Baseball	baseball
1458	499	0.9513	US townships	US townships
2457	3580	0.9512	sea snails	sea snail species descriptions
3441	1148	0.9506	Calories burned during physical activity	Calorie burn estimates based on weight and physical activity
990	3249	0.9461	Eurovision Song Contest	Eurovision Song Contest
4	3550	0.9450	Sri Lanka	Sri Lanka
3416	3115	0.9449	Radio station descriptions	Radio stations
321	2854	0.9439	List items	List items
750	1415	0.9418	Dream interpretations	Dream interpretation
2900	1083	0.9404	Kerala-related entities and terms	Kerala

Table 2. Features with high correlation in activation patterns across the D_{4096} dictionary for gemini and nemotron. Interpretations of the features (neurons) are generated using gemini-3-flash-preview from a sample of 25 activating texts.