ORTSAE: ORTHOGONAL SPARSE AUTOENCODERS UNCOVER ATOMIC FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) are a technique for sparse decomposition of neural network activations into human-interpretable features. However, current SAEs suffer from feature absorption, where specialized features capture instances of general features creating representation holes, and feature composition, where independent features merge into composite representations. In this work, we introduce Orthogonal SAE (OrtSAE), a novel approach aimed to mitigate these issues by enforcing orthogonality between the learned features. By implementing a new training procedure that penalizes high pairwise cosine similarity between SAE features, OrtSAE promotes the development of disentangled features while scaling linearly with the SAE size, avoiding significant computational overhead. We train OrtSAE across different models and layers and compare it with other methods. We find that OrtSAE discovers 9% more distinct features, reduces feature absorption (by 65%) and composition (by 15%), improves performance on spurious correlation removal ($+6\%$), and achieves on-par performance for other downstream tasks compared to traditional SAEs.

1 INTRODUCTION

Large Language Models (LLMs) have achieved remarkable performance in natural language processing, but their internal mechanisms remain poorly understood. Mechanistic interpretability aims to understand how neural networks function by reverse-engineering their computational processes (Olah et al., 2020). Central to this field is understanding *features*, the human-interpretable concepts represented as directions in a model's internal representation (Elhage et al., 2022; Park et al., 2023).

Early interpretability methods focused on analyzing individual neurons (Olah et al., 2020; Bills et al., 2023), but a key challenge has been that neurons are often *polysemantic*, responding to multiple unrelated concepts rather than encoding single interpretable features (Olah et al., 2020). One theory of why polysemanticity occurs is *superposition*, which posits that neural networks represent more features than they have dimensions (Elhage et al., 2022). Although this enables efficient use of model capacity, it significantly complicates interpretability research.

Sparse Autoencoders (SAEs) have emerged as a powerful approach to disentangling superposition (Bricken et al., 2023; Cunningham et al., 2023). By adding a sparsity penalty to the reconstruction loss, SAEs learn to decompose activations into a sparse latent space where each dimension aims to capture a distinct, interpretable feature (Gao et al., 2024; Marks et al., 2024). Traditional SAE variants (Bricken et al., 2023; Gao et al., 2024; Bussmann et al., 2024; Rajamanoharan et al., 2024) focused on improving reconstruction quality while maintaining sparsity. However, the standard objective can lead to two failure modes. As the number of SAE latents grows, *feature absorption* can occur (Fig. 2a), where a broad feature representation absorbs into more specific, token-aligned latents (e.g., a latent "starts with E" will activate on all tokens starting with "E", except for the token "elephant") (Chanin et al., 2024). Another issue is *feature composition* (Fig. 2b), in which independent features (e.g. representing "red" and "square") are merged into a single composite feature ("red square") (Leask et al., 2025). Both problems undermine the interpretability of SAE latents and the applicability of SAE representations for downstream tasks (Karvonen et al., 2025). To address these issues, Bussmann et al. (2025) introduced Matryoshka SAE, a hierarchical approach to organizing features at multiple levels of abstraction. However, this method introduces additional computational

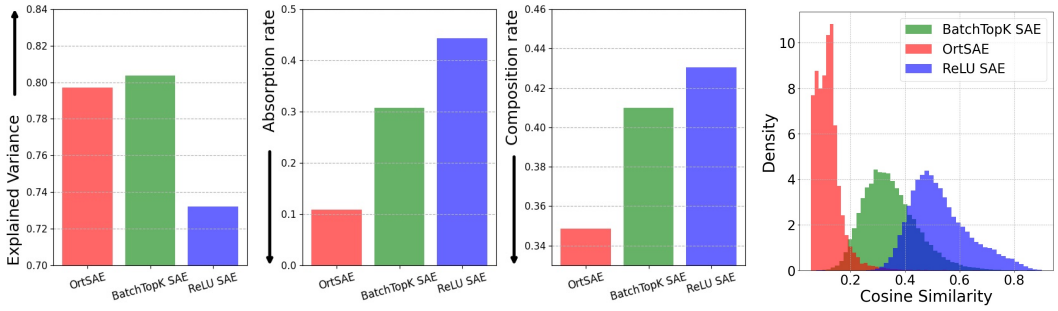


Figure 1: **Performance of OrtSAEs vs. traditional SAEs.** Bar plots display explained variance, absorption, and composition rates for three SAE variants at L0=70 sparsity. OrtSAEs show a marginally lower explained variance than BatchTopK SAEs but decreased absorption and composition, indicating better feature specificity. The density plot illustrates the distribution of pairwise cosine similarity values, computed as the maximum similarity between each decoder feature and its closest counterpart in the model, across all features at L0=70. OrtSAEs demonstrate lower pairwise cosine similarity, confirming greater decoder feature orthogonality compared to BatchTopK and ReLU SAEs.

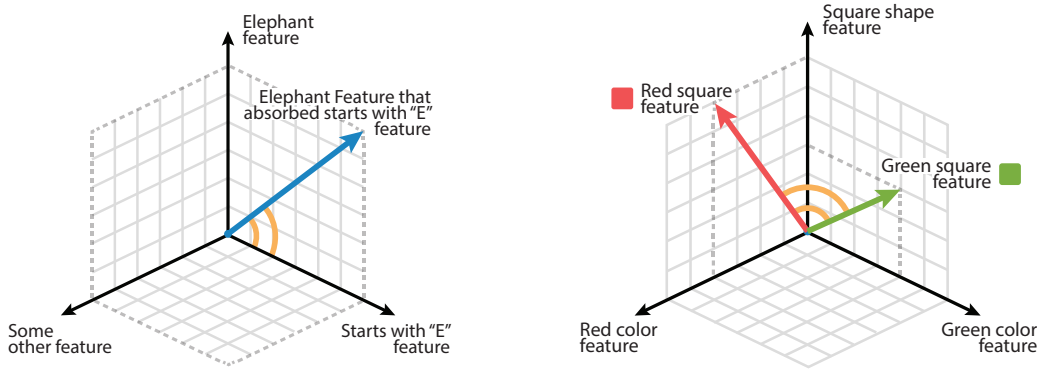
overhead and suffers from feature hedging (Chanin et al., 2025), a problem where correlated features merge at higher levels, reducing interpretability. This highlights the need for alternative approaches.

Feature absorption and composition lead to redundant representations where multiple latents capture overlapping concepts, which results in high cosine similarities between them. This suggests that enforcing orthogonality between SAE latents could provide a principled approach to mitigate these issues. Therefore, we propose **OrtSAE**, a novel approach to SAE training that promotes the emergence of more atomic features (Sec. 3.3). At each training step, we penalize high cosine similarities between SAE latents by introducing an additional orthogonality penalty. To optimize computation, we implement a chunk-wise strategy that divides SAE latents into smaller blocks, computes the penalty separately, and aggregates the results. This reduces the complexity from quadratic to linear with respect to the number of latents and introduces a negligible computational overhead. Importantly, this penalty scales efficiently without altering the core SAE architecture.

We train OrtSAE on the Gemma-2-2B (Team et al., 2024) and Llama-3-8B (Dubey et al., 2024) and compare it against traditional SAEs and Matryoshka SAE (Bussmann et al., 2025). Experimental results demonstrate that our objective reduces feature absorption and composition across a wide range of sparsity levels (Sec. 4.3). For example, at a L0 of 70 (Fig. 1), OrtSAE discovers 9% more distinct features, reduces feature absorption by 65%, and feature composition by 15% compared to traditional SAEs. On SAEBench (Karvonen et al., 2025), our method improves performance on spurious correlation removal by 6% while maintaining on-par performance for other downstream tasks (Sec. 4.4). Through qualitative experiments, we show that OrtSAE features efficiently decompose composite features learned by other SAEs into more atomic components (Sec. 4.3).

Our paper makes the following contributions:

- We propose OrtSAE, a novel approach to SAE training that directly addresses the issues of feature absorption and composition, without requiring complex architectural changes or significant computational overhead (Sec. 3.3).
- Comparison of OrtSAE with traditional SAEs shows that our method produces more distinct features, reduces absorption and composition rates (Sec. 4.3).
- Experimental results on SAEBench demonstrate that our method performs on-par with other SAE architectures, and outperforms them on spurious correlation removal (Sec. 4.4).



(a) Example of feature absorption

(b) Example of feature composition

Figure 2: **Illustration of feature absorption and feature composition**: (a) In feature absorption, specific features like "elephant" absorb broader features like "starts with E". (b) In feature composition, independent concepts like "red color" and "square form" are merged into composite features.

2 Related work

Sparse Autoencoders (SAEs) have gained significant traction for interpreting LLMs, addressing the challenge that LLMs often function as "black boxes". A key issue is polysemanticity, where individual neurons respond to multiple unrelated concepts (Bricken et al., 2023). SAEs aim to resolve this by decomposing dense LLM activation vectors into a sparse set of monosemantic features, each representing a single concept (Cunningham et al., 2023; Huben et al., 2025). Pioneering work Bricken et al. (2023) and Cunningham et al. (2023) demonstrated the effectiveness of this approach on small transformers, finding interpretable features such as DNA sequences or legal text. Subsequent efforts scaled SAEs to larger models like Claude 3 Sonnet (Templeton et al., 2024) and GPT-4 (Gao et al., 2024), as well as open-source models (Lieberum et al., 2024).

Limitations in basic SAEs, typically using ReLU activation with an L1 penalty (Bricken et al., 2023), such as L1-induced shrinkage (underestimation of feature strength) and difficulty in precise L0 control (Gao et al., 2024; Templeton et al., 2024) have driven architectural innovation. JumpReLU SAEs (Rajamanoharan et al., 2024), use a learned threshold within the activation function for direct L0 optimization. TopK SAE (Gao et al., 2024) selects only the top K activations, simplifying tuning and reducing shrinkage compared to L1; BatchTopK SAE (Bussmann et al., 2024) further improves it by applying the TopK constraint at the batch level for adaptive sparsity and improved reconstruction. The latter approach appears promising for our purposes, as it allows precise sparsity control along with excellent reconstruction capabilities. For a detailed overview of the SAE variations, we further refer the reader to the survey by Shu et al. (2025).

Despite ongoing advancements, SAEs continue to face challenges: Chanin et al. (2024) describes the phenomenon of feature *absorption*, when broad features absorb into more specific ones. Leask et al. (2025) highlights feature *composition*, when independent features merge into one larger feature, and introduces the MetaSAE, which emerges as a promising approach to identify these problems.

Recently, Bussmann et al. (2025) proposed Matryoshka SAE to address these issues by employing a hierarchical approach. It builds upon BatchTopK architecture and uses nested features with increasing latent space size so that SAE separately learns broad and specific features. However, this hierarchical design leads to feature hedging (Chanin et al., 2025), where narrow higher-level dictionaries merge correlated features, reducing interpretability. Additionally, this approach introduces substantial computational overhead ($+50\%$ compared to traditional SAEs) and a degradation in reconstruction performance. Furthermore, while its reliance on hierarchical representation seems intuitive, its interpretability remains poorly explored. In contrast, we explore an alternative direction of representations decorrelation (Cogswell et al., 2016; Wang et al., 2021; Rodríguez et al., 2017). OrtSAE proposes an efficient approach by directly enforcing the orthogonality between SAE latents. It avoids all of the mentioned problems while achieving performance similar to Matryoshka SAEs in mitigating the feature absorption and composition challenges.

3 ORTHOGONAL SPARSE AUTOENCODERS

3.1 TRADITIONAL SPARSE AUTOENCODERS

Sparse autoencoders (SAEs) aim to reconstruct model activations $\mathbf{x} \in \mathbb{R}^n$ as a sparse linear combination of $m \gg n$ feature vectors, or *latents*. Formally, a SAE consists of an encoder and a decoder:

$$\begin{aligned} \mathbf{h}(\mathbf{x}) &= \sigma(\mathbf{W}^{\text{enc}}\mathbf{x} + \mathbf{b}^{\text{enc}}), \\ \hat{\mathbf{x}}(\mathbf{h}) &= \mathbf{W}^{\text{dec}}\mathbf{h} + \mathbf{b}^{\text{dec}}. \end{aligned} \tag{1}$$

The encoder, followed by a non-linearity $\sigma(\cdot)$, learns a mapping from the activations to a sparse and overcomplete latent code $\mathbf{h}(\mathbf{x}) \in R^m$. Given $\mathbf{h}(\mathbf{x})$, the decoder reconstructs the original input as a sparse linear combination of latents, $\mathbf{W}^{\text{dec}}_{\mathbf{i}}$, $\mathbf{i} = 1, ..., m$, as $\hat{\mathbf{x}}$.

The standard loss function to train a SAE is defined as:

$$\mathcal{L}(\mathbf{x}) = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{h}(\mathbf{x}))\|_2^2}_{\mathcal{L}_{\text{reconstruct}}} + \underbrace{\lambda\, S(\mathbf{h}(\mathbf{x}))}_{\mathcal{L}_{\text{sparsity}}} + \alpha L_{\text{aux}}, \tag{2}$$

where S is a sparsity penalty and λ is a coefficient controlling the trade-off between sparsity and reconstruction quality. The optional L_{aux} term covers any auxiliary penalties (e.g. recycling dead units (Gao et al., 2024)).

Traditional SAEs focus on reducing reconstruction loss while increasing sparsity. The ReLU SAE (Bricken et al., 2023; Cunningham et al., 2023) uses the ReLU activation function and applies an L1 penalty to ensure sparsity in $\mathbf{h}(\mathbf{x})$. TopK SAE (Gao et al., 2024) achieves sparsity by zeroing all entries of $\mathbf{h}(\mathbf{x})$ except for the K largest ones. BatchTopK (Bussmann et al., 2024) SAE further improves the idea by selecting the top $B \times K$ entries across a batch of $\mathbf{h}(\mathbf{x})$, allowing some examples to have more or less active latents.

3.2 CHALLENGES IN TRAINING SAES

Traditional SAEs training objectives (Eq. 2) combine reconstruction loss and sparsity penalty. While sparsity is required to decompose activations into interpretable features, optimization of it results in multiple failure modes. *Feature absorption* (Fig. 2a) occurs when an interpretable feature becomes SAE latent which appears to represent that feature, but fails to fire on arbitrary tokens that it seemingly should activate on. Instead, token-aligned latents fire, "absorbing" part of the feature representation to satisfy the sparsity objective by activating fewer latents overall. *Feature composition* (Fig. 2b) occurs when features overlap. To optimize over sparsity, SAE learns a single latent that captures the specific combination of features (e.g. "red square") instead of representing the underlying features ("red" and "square") with separate latents.

Feature absorption and composition produce redundant representations where multiple latents capture overlapping concepts, leading to high cosine similarities between decoder vectors. Formally, consider two atomic features A ("red") and B ("square") (Fig. 2b). In traditional SAEs, these independent features can merge into a composite feature C ("red square"). Let $\mathbf{W}^{\text{dec}}_A$, $\mathbf{W}^{\text{dec}}_B$, and $\mathbf{W}^{\text{dec}}_C$ denote the decoder vectors for features A, B, and C, respectively. When feature composition occurs, C incorporates components of both features A and B. This creates higher correlations between C and each atomic feature: $\cos(\mathbf{W}^{\text{dec}}_C, \mathbf{W}^{\text{dec}}_A) > \cos(\mathbf{W}^{\text{dec}}_A, \mathbf{W}^{\text{dec}}_B)$ and $\cos(\mathbf{W}^{\text{dec}}_C, \mathbf{W}^{\text{dec}}_B) > \cos(\mathbf{W}^{\text{dec}}_A, \mathbf{W}^{\text{dec}}_B)$. Similarly, feature absorption creates overlapping latents, resulting in the decoder vectors that are more correlated than they should be for truly atomic features. To address these issues, OrtSAE extends the traditional SAE objective by enforcing orthogonality between SAE latents. At each training step, we penalize high cosine similarities between SAE latents, directly approaching both absorption and composition problems by encouraging the formation of more atomic features.

3.3 ORTSAE TRAINING PROCEDURE

The main contribution of OrtSAE is the introduction of a new orthogonalization penalty that penalizes high similarities between SAE latents. Formally, given a SAE with decoder matrix $\mathbf{W}^{\text{dec}} \in \mathbb{R}^{n \times m}$,

we first define the cosine similarity between two feature vectors as:

$$\cos(\mathbf{W}_\mathbf{i}^{\text{dec}}, \mathbf{W}_\mathbf{j}^{\text{dec}}) = \frac{\langle \mathbf{W}_\mathbf{i}^{\text{dec}}, \mathbf{W}_\mathbf{j}^{\text{dec}} \rangle}{\max(\|\mathbf{W}_\mathbf{i}^{\text{dec}}\|_2 \cdot \|\mathbf{W}_\mathbf{j}^{\text{dec}}\|_2, \delta)}, \tag{3}$$

where $\delta > 0$ is a small constant added to prevent division by zero. Using this definition, we formulate our orthogonality penalty as:

$$L_{\text{orthogonal}}(\mathbf{W}^{\text{dec}}) = \frac{1}{K(m)} \sum_{k=1}^{K(m)} \frac{1}{|C_k|} \sum_{i \in C_k} \left(\max_{\substack{j \in C_k \\ j \neq i}} \cos(\mathbf{W}_\mathbf{i}^{\text{dec}}, \mathbf{W}_\mathbf{j}^{\text{dec}}) \right)^2. \tag{4}$$

Instead of computing all pairwise similarities between feature vectors $\mathbf{W}_\mathbf{i}^{\text{dec}}$, which would require $\mathcal{O}(m^2)$ operations and is infeasible for large m, at each training step we *randomly* partition the latent space into $K := K(m)$ equal chunks, each containing a fixed number of latents, $|C_k|, k = 1, ..., K(m)$, proportional to m. Within each k-th chunk, we find the maximum pairwise cosine similarity between every $\mathbf{W}_\mathbf{i}^{\text{dec}}$, $\mathbf{i} \in C_k$ and all other latents from C_k, square this value to penalize highly correlated features more and compute the expectation. We compute the final value by averaging across all K chunks. This chunk-wise strategy reduces the computational complexity to $\mathcal{O}(m)$ and provides an efficient scaling strategy to a larger latent spaces.

With the orthogonal penalty defined, OrtSAE training objective is defined as:

$$\mathcal{L}_{\text{OrtSAE}}(x) = L_{\text{MSE}} + \lambda L_{\text{sparsity}} + \alpha L_{\text{aux}} + \gamma L_{\text{orthogonal}}, \tag{5}$$

where γ is an *orthogonality coefficient* that controls the strength of the applied penalty.

To further enhance computational efficiency, we explore computing the orthogonality loss every fifth training iteration, scaling the orthogonality coefficient γ by a factor of 5 to maintain regularization strength, which yields comparable performance with significantly reduced computational overhead, as detailed in Appendix C.

4 Experiments

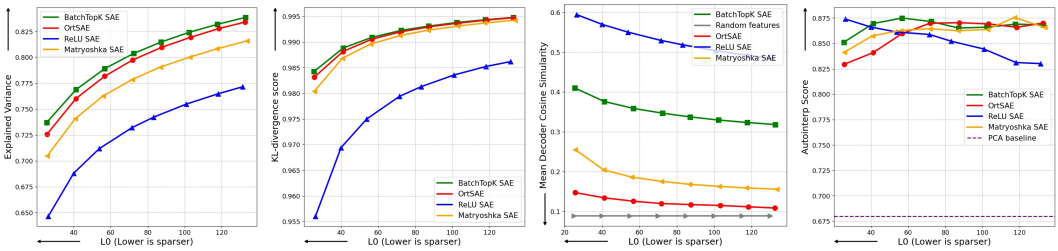
Here, we present our experimental evaluation of OrtSAE. Sec. 4.1 covers the experimental setup. Sec. 4.2 compares OrtSAE's core performance metrics to other methods. Sec. 4.3 presents quantitative and qualitative results on feature atomicity. Sec. 4.4 assesses OrtSAE's performance on downstream tasks using SAEBench.

4.1 Experimental Setup.

Baselines. We compare our model with: 1) traditional SAEs, such as ReLU SAE (Bricken et al., 2023; Cunningham et al., 2023) and state-of-the-art BatchTopK SAE (Bussmann et al., 2024); 2) recent Matryoshka SAE that enforce nested, hierarchical learning at multiple feature levels (Bussmann et al., 2025).

Models Configuration. Following the work of Bussmann et al. (2025) we train SAEs on the activations from layer 12 of the Gemma-2-2B (26 layers total). Each SAE has latent space of size $m = 65536$ and sparsity levels L0 in $\{25, 40, 55, 70, 85, 100, 115, 130\}$. The training uses 500 million tokens from the OpenWebText dataset (Gokaslan and Cohen, 2019) with a context length of 1024. As a basis of OrtSAE, we follow BatchTopK SAE repository, leveraging BatchTopK's precise L0 sparsity control. For OrtSAE, we set the number of chunks $K(m) = \lceil m/8192 \rceil$ (yielding $K = 8$ for $m = 65536$) with $\gamma = 0.25$. The full details and hyperparameters are available in the Appendix A. To assess the transferability of our approach, we also conduct experiments on layer 20 of Gemma-2-2B and layer 20 of Llama-3-8B and report the results in Appendix B.To ensure the reproducibility, we will publicly release all code and hyperparameters.

Evaluation. Following Gao et al. (2024) and Bussmann et al. (2025), we evaluate SAEs core performance through explained variance, KL-divergence, orthogonality, and feature interpretability. Atomicity analysis includes absorption metrics, MetaSAE-based composition rate (Leask et al.,



(a) Explained variance. (b) KL divergence score. (c) Mean cosine similarity. (d) Autointerp Score.

Figure 3: **Core performance metrics.** (a) Explained variance: OrtSAE shows slightly lower reconstruction fidelity than BatchTopK SAE but outperforms Matryoshka SAE. (b) KL-divergence score: OrtSAE matches BatchTopK SAE and exceeds Matryoshka SAE. (c) Mean cosine similarity to closest decoder feature: OrtSAE achieves near-random initialization orthogonality, significantly lower than other SAE variants. (d) Autointerp Score: OrtSAE demonstrates interpretability comparable to both BatchTopK and Matryoshka SAEs.

2025), clustering, cross-model overlap, and qualitative investigation. Downstream assessment uses SAEBench (Karvonen et al., 2025) with spurious correlation removal, targeted probe perturbation, sparse probing, and RAVEL tasks.

4.2 Foundational Performance Analysis

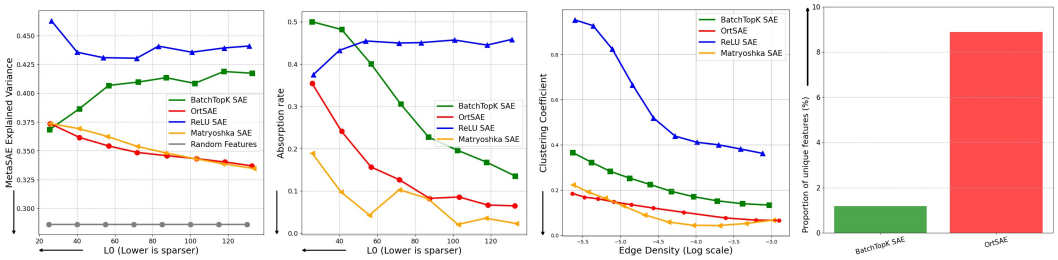
We evaluate OrtSAE using four metrics: (1) reconstruction fidelity, computed as the fraction of explained variance; (2) downstream predictive performance, measured by the KL-divergence score between the original LLM's output distributions and those generated using reconstructed activations; (3) feature orthogonality, calculated as the mean cosine similarity to each feature's nearest decoder neighbor; and (4) feature interpretability, assessed via the Autointerp Score. Fig. 3 shows these metrics across sparsity levels, demonstrating OrtSAE's balance between reconstruction quality and model functionality preservation. To further validate the generalizability of our findings across different model architectures and layers, we conducted additional experiments on layer 20 of Gemma-2-2B and layer 20 of Llama-3-8B, with results reported in App. B. We also analyze the impact of varying the number of chunks on OrtSAE performance to assess its robustness and scalability, with details provided in App. C.

Reconstruction and Predictive Performance. We first evaluate reconstruction quality through explained variance, measuring how accurately each SAE reconstructs input activations. To assess whether these reconstructions preserve the model's functionality, we additionally examine downstream predictive performance using KL-divergence scores (detailed descriptions provided in App.D). We also tested the effect of decoded activations on the base language model's perplexity using LogLoss, which shows the same patterns as the KL-divergence scores (see App. E). Fig. 3a shows OrtSAE achieves comparable performance to BatchTopK SAE while outperforming Matryoshka SAE by 2% in fraction of explained variance. The KL-divergence scores (Fig. 3b) reveal nearly identical predictive behavior between OrtSAE and BatchTopK SAE, with both slightly surpassing Matryoshka SAE. These results are particularly notable because OrtSAE maintains strong performance despite its additional orthogonality constraints, whose effects we analyze next through feature similarity.

Feature Orthogonality. To quantify the separation of decoder features, we compute the mean cosine similarity (MeanCosSim) for each feature vector i to its closest neighbor in the decoder matrix:

$$\text{MeanCosSim} = \frac{1}{m}\sum_{i=1}^{m}\max_{j\neq i}\cos(\mathbf{W}_i^{\text{dec}}, \mathbf{W}_j^{\text{dec}}). \tag{6}$$

As shown in Fig. 3c, OrtSAE achieves superior separation with MeanCosSim values 2.7 times lower than BatchTopK and 1.5 times lower than Matryoshka SAE, approaching random initialization levels. This enhanced orthogonality directly contributes to improved feature atomicity, as shown in Sec. 4.3.



(a) Composition rate. (b) Absorption rate. (c) Clustering rate. (d) Unique features.

Figure 4: **Atomicity metrics.** (a) MetaSAE-based Composition rate: OrtSAE shows reduced feature merging compared to traditional SAEs and matches Matryoshka SAE. (b) Absorption rate: OrtSAE shows significant improvement over BatchTopK with performance approaching Matryoshka SAE. (c) Feature clustering: OrtSAE matches Matryoshka SAE in low feature interconnection, both outperforming traditional SAEs. (d) Proportion of unique features: OrtSAE discovers substantially more distinct features than BatchTopK SAE.

Feature Interpretability. The Autointerp Score evaluates feature interpretability using GPT-4o-mini as an LLM judge. Following Paulo et al. (2024)'s methodology, we first generate interpretable descriptions of 1,000 latents using an LLM, then quantitatively assess these explanations by measuring how accurately the LLM can predict whether each latent activates (fires) on new input tokens (detailed descriptions provided in Appx. F). Fig. 3d demonstrates comparable interpretability between OrtSAE, BatchTopK SAE, and Matryoshka SAE latents across all tested sparsity levels.

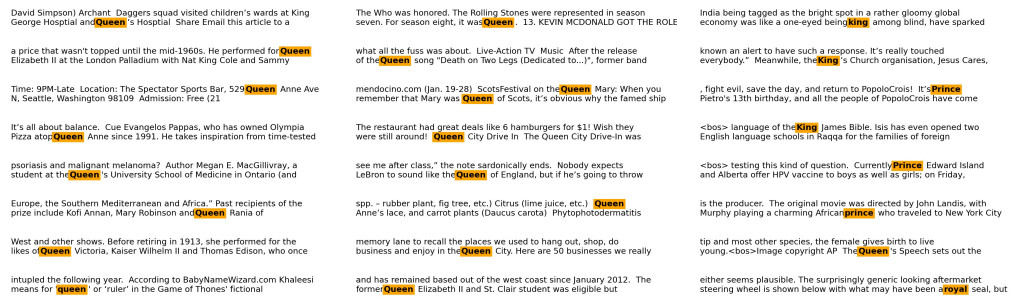
These results demonstrate that OrtSAE achieves an optimal balance - matching top reconstruction performance while significantly improving feature separation without compromising interpretability. This demonstrates that our new training procedure can enhance feature quality while preserving model functionality, as we explore further in our atomicity analysis.

4.3 Atomicity Analysis

We assess feature atomicity through four quantitative measures: (1) MetaSAE-based composition rates, (2) absorption metrics, (3) clustering coefficients, and (4) cross-model feature uniqueness, complemented by qualitative analysis of decomposed features. Fig. 4 presents the quantitative comparisons across sparsity levels, while Fig. 5 illustrates OrtSAE's ability to disentangle composite features through concrete examples.

MetaSAE-Based Feature Composition Analysis. We train a MetaSAE on the decoder features of SAEs, following the methodology of Leask et al. (2025). The MetaSAE follows the same training procedure as ordinary SAEs (Sec. 4.1) but operates on decoder features rather than LLM activations, attempting to decompose these higher-level representations into more atomic latent components (detailed descriptions provided in Appx. G). The MetaSAE employs a BatchTopK architecture with sparsity $k = 4$ and a dictionary size reduced to 25% of the original SAEs. We measure composition via explained variance, where lower values indicate higher atomicity. Fig. 4a shows that OrtSAE attains a composition rate much lower (by 0.06 at L0 of 70) than BatchTopK SAE and similar to Matryoshka SAE, reflecting pronounced feature atomicity and resistance to decomposition into simpler constituents.

Feature Absorption Analysis. Feature absorption is evaluated using SAEBench (Karvonen et al., 2025). Following established methodologies for studying absorption in sparse autoencoders (Chanin et al., 2024), we perform tests in the first-letter classification and hierarchical concept domains (detailed descriptions provided in Appx. B). Fig. 4b shows that OrtSAE achieves a significantly reduced absorption rate compared to BatchTopK SAE (by 0.17 at L0 of 70), but slightly higher than Matryoshka SAE, indicating effective minimization of conceptual overlap compared to traditional SAEs.



(a) BatchTopK SAE feature that activates only on "Queen" token

(b) OrtSAE feature that activates only on "Queen" token

(c) OrtSAE feature that activates only on titles and royal concepts

Figure 5: **Decomposition of a BatchTopK SAE Feature into OrtSAE Features.** (a) A BatchTopK SAE feature activating solely on the token "Queen", expressed as a linear combination of two OrtSAE features: (b) An OrtSAE feature specific to the token "Queen". (c) An OrtSAE feature capturing royal titles, revealing how OrtSAE disentangles the broader royalty concept absorbed by the "Queen" feature in BatchTopK SAE.

Clustering Properties of Decoder Features. The clustering coefficient measures how tightly connected features are in a graph of decoder interactions, where edges represent high cosine similarity between features (detailed descriptions provided in Appx. C). A lower coefficient indicates that features are more independent, forming fewer interconnected groups, which suggests greater feature atomicity. We evaluate this across 10 similarity thresholds at varying edge densities (the ratio of existing edges to the number of all possible edges). Fig. 4c shows the clustering coefficient of OrtSAE is substantially lower than that of BatchTopK SAE and similar to Matryoshka SAE, signifying enhanced feature independence compared to traditional SAEs.

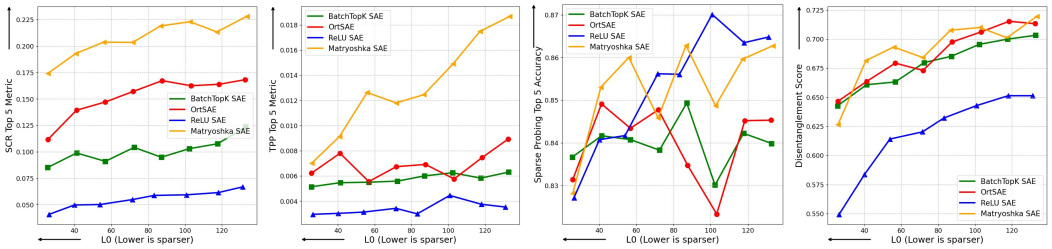
Cross-Model Feature Overlap Analysis. We measure feature uniqueness by computing maximum pairwise cosine similarity between OrtSAE and BatchTopK SAE features, at a L0 of 70. A feature is considered unique if all cross-model similarities are below 0.2. OrtSAE retains 9% unique features, compared to 1.5% for BatchTopK (Fig. 4d). This six-fold increase in unique features highlights OrtSAE's ability to discover novel features, enhancing scalability for larger dictionaries.

Qualitative Analysis of Feature Atomicity. We analyze feature atomicity by decomposing BatchTopK SAE features into sparse combinations of OrtSAE features, following a methodology adapted from MetaSAE (Leask et al., 2025). We select BatchTopK SAE (at L0 of 70) features if their OrtSAE (at L0 of 70) approximation has a cosine similarity above 0.95 and each coefficient is at least 0.1, ensuring meaningful contributions. Fig. 5 illustrates the decomposition of a BatchTopK SAE feature for "Queen terms" into two OrtSAE features: one for "Queen terms" and another for "titles and royal concepts". This demonstrates how OrtSAE disentangles broader concepts absorbed by specialized features. Additional examples of feature decompositions are provided in Appx. D.

These results demonstrate OrtSAE improves feature atomicity over traditional SAEs, achieving lower composition and absorption rates, reduced clustering, and a higher proportion of unique features. Qualitative evidence supports these findings, demonstrating the effective decomposition of complex BatchTopK features by OrtSAE features. These results highlight the efficacy of orthogonality constraint in yielding disentangled representations, paving the way for downstream tasks.

4.4 Downstream Benchmarks

We evaluate OrtSAE using the SAEBench (Karvonen et al., 2025), which measures SAEs quality across a diverse set of tasks related to practical downstream applications. The key metrics we report on are Spurious Correlation Removal (SCR), Targeted Probe Perturbation (TPP), Sparse Probing, and RAVEL.



(a) SCR score. (b) TPP score. (c) Probing accuracy. (d) RAVEL score.

Figure 6: **Results on SAEBench.** (a) Spurious Correlation Removal: OrtSAE outperforms traditional SAEs but achieves lower scores than Matryoshka. (b) Targeted Probe Perturbation: OrtSAE shows modest improvements over traditional SAEs, while Matryoshka demonstrates the strongest performance. (c) Sparse Probing: OrtSAE maintains accuracy comparable to baseline methods. (d) RAVEL: OrtSAE achieves scores similar to both BatchTopK and Matryoshka SAEs.

Spurious Correlation Removal and Targeted Probe Perturbation. SCR evaluates the removal of spurious correlations (e.g., gender in profession classification) by zero-ablating SAE latents, while TPP tests class-specific concept isolation in multi-class settings via targeted ablation. Fig. 6 demonstrates that OrtSAE achieves stronger performance than traditional SAEs in both tasks, with a significant improvement in SCR scores and modest gains in TPP compared to BatchTopK SAE. While Matryoshka SAE achieves the highest absolute scores, OrtSAE delivers similarly strong results with lower computational overhead and traditional architecture.

Sparse Probing. Sparse Probing evaluates the ability of SAEs to isolate specific concepts, such as sentiment, within individual latents without explicit supervision. For each concept, we select the top-k latents by comparing their mean activations on positive versus negative examples. A linear probe is then trained on these latents to predict the concept. High probe accuracy indicates that the latents effectively capture the target concept in a disentangled manner. As shown in Fig. 6c, OrtSAE achieves probing accuracy comparable to BatchTopK and Matryoshka SAEs across various sparsity levels, demonstrating its capability to produce interpretable, concept-aligned features.

RAVEL. RAVEL (Resolving Attribute–Value Entanglements in Language Models) (Huang et al., 2024) evaluates disentanglement by manipulating attributes in LLM activations (e.g., transferring city-related features from "Tokyo" to "Paris") while minimizing interference with unrelated attributes (e.g., France-related context). Fig. 6d demonstrates that OrtSAE, BatchTopK SAE, and Matryoshka SAE exhibit equivalent performance, highlighting OrtSAE's ability to enable precise interventions without compromising efficacy.

Evaluated against traditional SAEs, OrtSAE demonstrates superior SCR performance, competitive TPP results, and consistently comparable outcomes in Sparse Probing and RAVEL, highlighting its ability to generate atomic, disentangled features without sacrificing downstream utility.

5 Conclusion

In this work, we introduce OrtSAE, a novel sparse autoencoder training approach that enhances latent atomicity through orthogonal constraints on decoder features. This method effectively addresses feature absorption and composition, key obstacles to interpretable representations, while preserving reconstruction fidelity comparable to traditional SAEs. Notably, OrtSAE achieves this with minimal computational overhead through an efficient chunk-wise orthogonalization penalty that scales linearly with feature count. Experiments across different language model families demonstrate OrtSAE's significant reductions in absorption and composition rates, yielding 9% more distinct features, superior spurious correlation removal (+6%), and on-par performance across other SAEBench tasks. These insights underscore geometric constraints' role in disentangling superposed representations, offering fresh perspectives on the superposition hypothesis. Future work should investigate using orthogonal features as more interpretable building blocks for neural circuit discovery, potentially leading to clearer mechanistic models of model computations.

6 REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we have taken several measures throughout the paper and supplementary materials. All experimental details, including model architectures, hyperparameters, and training procedures, are comprehensively documented in Section 4.1 and Appendix A. We provide complete specifications for our OrtSAE implementation, including the orthogonal penalty formulation and chunking strategy in Section 3.3. The evaluation metrics and benchmarks are described in detail in Sections 4.2-4.4, with additional methodological explanations in Appendices B-G. Code for OrtSAE training, evaluation scripts, feature analysis tools, and SAEBench integration will be released anonymously as supplementary material and made fully public upon acceptance. The datasets used in our experiments (OpenWebText) are publicly available, and we specify exact data processing steps in Appendix A. For the SAEBench evaluations, we follow established protocols from prior work with detailed descriptions of each task. Additional experiments on different model architectures and layers (Appendix B) and ablation studies on key hyperparameters (Appendix C) further validate the robustness of our approach.

REFERENCES

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv preprint arXiv:2409.14507*, 2024.

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

Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations, 2016. URL https://arxiv.org/abs/1511.06068.

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

Abhinav Dubey et al. Llama 3: Advancements in open-source language models. *arXiv preprint arXiv:2407.XXXX*, 2024. URL https://arxiv.org/abs/2407.XXXX.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

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

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

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

Robert Huben, Logan Riggs, Aidan Ewart, Hoagy Cunningham, and Lee Sharkey. Sparse autoencoders can interpret randomly initialized transformers, 2025. URL https://arxiv.org/abs/2501.17727.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, et al. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*, 2025.

Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. *arXiv preprint arXiv:2502.04878*, 2025.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.19. URL https://aclanthology.org/2024.blackboxnlp-1.19/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024.

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

Pau Rodríguez, Jordi Gonzàlez, Guillem Cucurull, Josep M. Gonfaus, and Xavier Roca. Regularizing cnns with locally constrained decorrelations, 2017. URL https://arxiv.org/abs/1611.01967.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. *arXiv preprint arXiv:2503.05613*, 2025.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

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

Zhennan Wang, Canqun Xiang, Wenbin Zou, and Chen Xu. Mma regularization: Decorrelating weights of neural networks by maximizing the minimal angles, 2021. URL https://arxiv.org/abs/2006.06527.

A Additional details of SAE training setup

Following the approach of (Bussmann et al., 2025), we train Sparse Autoencoders (SAEs) on activations from layer 12 of the Gemma-2-2B model (Team et al., 2024) (26 layers total) to align with prior work. To assess the generalizability of our approach, we also conduct experiments on layer 20 of Gemma-2-2B and layer 20 of Llama-3-8B (Dubey et al., 2024) (32 layers total). Each SAE has latent space of size $m = 65536$ and sparsity levels L0 in $\{25, 40, 55, 70, 85, 100, 115, 130\}$. The training uses 500 million tokens from the OpenWebText dataset (Gokaslan and Cohen, 2019) with a context length of 1024. We employ the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 2×10^{-4} and a batch size of 2048. All SAE variants except ReLU SAE employ an auxiliary loss coefficient $\alpha = 1/32$ to mitigate dead features during training. As a basis of OrtSAE, we follow BatchTopK SAE repository, leveraging BatchTopK's precise L0 sparsity control. For OrtSAE, we set the number of chunks $K(m) = \lceil m/8192 \rceil$ (yielding $K = 8$ for $m = 65536$) with $\gamma = 0.25$. All SAEs are trained comparably with identical data ordering and hyperparameters on 1 H100 GPU, 80GB. For one SAE, training requires approximately 10 hours. To ensure reproducibility, we will publicly release all code, hyperparameters, and instructions for accessing the datasets. The results of additional experiments on layer 20 of Gemma-2-2B and Llama-3-8B focusing on a sparsity level of L0 = 70 are reported in the Appendix B.

B Additional Experiments on Gemma-2-2B and Llama-3-8B

To address the generalizability of our findings, we conducted additional experiments on layer 20 of Gemma-2-2B (26 layers total) and layer 20 of Llama-3-8B (32 layers total) (Dubey et al., 2024), following the experimental setup described in Sec. 4.1. We trained SAEs with a sparsity level of L0 = 70 and measured key metrics: explained variance, mean cosine similarity, composition rate, absorption rate, and Spurious Correlation Removal (SCR) score. The results, presented in Tables 1 and 2, confirm the findings from layer 12 of Gemma-2-2B, showing consistent reductions in feature absorption and composition, as well as improved performance in tasks such as Spurious Correlation Removal. Notably, the Explained Variance gap between Matryoshka SAE and OrtSAE widens to 0.04 in Llama-3-8B (0.722 vs. 0.762), reinforcing OrtSAE's advantage in reconstruction performance.

Table 1: **Performance of SAEs trained on layer 20 of Gemma-2-2B with L0=70.**

SAE model	Expl. var.	Mean Cos. sim.	Comp. rate	Abs. rate	SCR score
ReLU SAE	0.784	0.549	0.527	0.371	0.144
BatchTopK SAE	**0.843**	0.354	0.490	0.220	0.308
Matryoshka SAE	0.811	0.148	0.349	**0.015**	**0.385**
OrtSAE	0.836	**0.112**	**0.340**	0.095	0.322

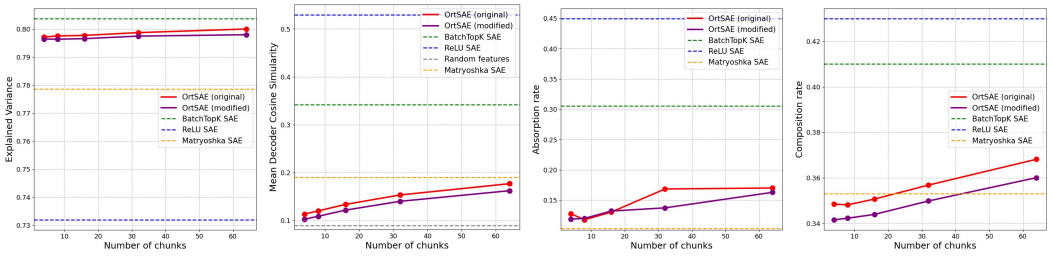
Table 2: **Performance of SAEs trained on layer 20 of Llama-3-8B with L0=70.**

SAE model	Expl. var.	Mean Cos. sim.	Comp. rate	Abs. rate	SCR score
ReLU SAE	0.704	0.517	0.413	0.490	0.060
BatchTopK SAE	**0.769**	0.327	0.461	0.148	0.103
Matryoshka SAE	0.722	0.149	0.323	**0.022**	**0.191**
OrtSAE	0.762	**0.107**	**0.316**	0.070	0.151

C Effect of Number of Chunks on OrtSAE Performance

We evaluated the impact of the number of chunks and the frequency of orthogonality loss computation on OrtSAE performance, following the experimental setup in Sec. 4.1. OrtSAE was tested with chunk counts $K \in \{4, 8, 16, 32, 64\}$, using a fixed sparsity at L0 of 70. Additionally, to enhance computational efficiency, we explored a modified OrtSAE variant where the orthogonality loss is computed every fifth training iteration, with the orthogonality coefficient γ scaled by a factor of 5 to maintain regularization strength.

As shown in Figure 7, OrtSAE demonstrates robust performance in core reconstruction and atomicity metrics across different numbers of chunks. The original OrtSAE and the modified OrtSAE show similar trends, with the modified version maintaining performance even better then original. Explained variance (Fig.7a) remains stable around 0.77–0.78 across chunk counts for both variants, slightly outperforming ReLU SAE and Matryoshka SAE. Mean cosine similarity (Fig.7b) increases modestly with more chunks (e.g., from 0.10 at $K = 4$ to 0.20 at $K = 64$), but remains lower than BatchTopK and ReLU SAEs. Absorption rate (Fig.7c) and composition rate (Fig.7d) also increase slightly with more chunks (e.g., absorption from 0.15 to 0.20), yet both variants outperform traditional SAEs. The modified OrtSAE achieves performance within 1–2% of the original OrtSAE across these metrics, reducing the computational overhead of the orthogonality loss by approximately five times, as it is calculated in only 20% of training iterations. Overall, OrtSAE demonstrates both robustness and scalability by maintaining core SAE performance across varying chunk counts while reducing training overhead to within 4% of the BatchTopK baseline (see Table 3).



(a) Explained variance (b) Mean cosine similarity (c) Absorption rate (d) Composition rate

Figure 7: **Performance of OrtSAE across different number of chunks at L0 of 70**. OrtSAE shows robust performance in core reconstruction and atomicity metrics across different number of chunks.

Table 3: **Training times for SAE variants.** Ratios are relative to BatchTopK SAE.

Method	Training Time (minutes)	Time Ratio*
BatchTopK SAE	325	1.0×
Matryoshka SAE	373	1.15×
OrtSAE modified ($K = 8$)	361	1.11×
OrtSAE modified ($K = 64$)	**340**	**1.04**×

*Relative to BatchTopK SAE

D KL-divergence score definition

The KL-divergence score assesses how effectively a sparse autoencoder (SAE) preserves the predictive behavior of a language model by comparing next-token probability distributions. We define P_{orig} as the distribution from the original model, P_{SAE} as the distribution when activations are replaced by SAE reconstructions, and P_{ablated} as the distribution when activations are set to zero, serving as a baseline.

The score is given by:

$$\text{KL-Divergence Score} = \frac{D_{KL}(P_{\text{ablated}} \parallel P_{\text{orig}}) - D_{KL}(P_{\text{SAE}} \parallel P_{\text{orig}})}{D_{KL}(P_{\text{ablated}} \parallel P_{\text{orig}})}$$

where D_{KL} denotes the Kullback-Leibler divergence. This metric ranges from 0, indicating no improvement over the zero-ablated baseline, to 1, indicating perfect reconstruction ($P_{\text{SAE}} = P_{\text{orig}}$). In SAEBench, the score is averaged over a dataset to evaluate SAE reconstruction quality.

E LogLoss Results

To further evaluate the impact of decoded activations on the base language model's predictive performance, we compute LogLoss scores for SAEs trained on layer 12 of Gemma-2-2B across various sparsity levels (L0 $\in \{40, 70, 100, 130\}$). LogLoss measures the negative log-likelihood of the model's next-token predictions, with lower values indicating better preservation of the base model's predictive behavior. The results, shown in Table 4, align closely with the KL-divergence findings (Appendix D), confirming that OrtSAE maintains predictive performance comparable to BatchTopK SAE, with both outperforming ReLU SAE and Matryoshka SAE.

Table 4: **LogLoss of SAEs trained on layer 12 of Gemma-2-2B.** Lower LogLoss indicates better preservation of the base language model's predictive performance.

SAE model	L0=40	L0=70	L0=100	L0=130
No SAE (LLM's LogLoss)	2.4533	2.4533	2.4533	2.4533
ReLU SAE	2.7657	2.6562	2.6094	2.5935
BatchTopK SAE	**2.5623**	**2.5312**	**2.5152**	**2.5020**
Matryoshka SAE	2.5786	2.5321	2.5161	2.5025
OrtSAE	2.5646	2.5319	2.5159	2.5024

F Feature Interpretability Metrics Details

To evaluate the interpretability of Sparse Autoencoder (SAE) features, we follow the automated interpretability methodology outlined in (Paulo et al., 2024; Karvonen et al., 2025), leveraging large language models (LLMs) to generate and validate human-readable feature descriptions. We select 1,000 random SAE features, excluding "dead" features. For each feature, we collect up to 10 top-activating sequences from the OpenWebText dataset (Gokaslan and Cohen, 2019) and prompt GPT-4o-mini to generate a concise description, such as "sentiment terms" or "math expressions," capturing the feature's core concept.

To assess these descriptions, we create a test set for each feature with 100 sequences: 50 activating the feature at varying strengths and 50 random non-activating sequences, all sourced from OpenWebText. A separate GPT-4o-mini model predicts whether each sequence activates the feature based on the description, treating it as a binary (yes/no) classification task. The *Autointerp Score*, shown in Fig. 3d, is the prediction accuracy, measuring how well the description generalizes to new data. A high score indicates a monosemantic, interpretable feature, while a lower score may suggest polysemanticity or an inaccurate description. OrtSAE demonstrates interpretability comparable to BatchTopK and Matryoshka SAEs across sparsity levels (Sec. 4.2), confirming its ability to produce clear, disentangled feature representations.

G MetaSAE-Based Feature Composition Metrics Details

The MetaSAE-based feature composition analysis, originally proposed by Leask et al. (2025) and extended by Bussmann et al. (2025), provides a quantitative method for assessing the atomicity of features learned by sparse autoencoders. This approach measures the degree of feature composition by training a secondary sparse autoencoder (MetaSAE) on the feature vectors of the primary SAE to decompose them into more atomic components.

In our implementation, we train MetaSAEs on the decoder weight matrices of both OrtSAE and BatchTopK SAE. The decoder weights are treated as input data points, where each feature vector from the primary SAE's dictionary serves as an input to the MetaSAE. The MetaSAE follows the same BatchTopK architecture as our primary SAEs but operates on this different input space.

The MetaSAE is configured with a dictionary size equal to one-quarter of the primary SAE's dictionary size (16,384 meta-latents for our primary SAEs with 65,536 features). We apply a sparsity constraint ensuring an average of 4 active meta-latents per decoder vector reconstruction. The MetaSAE learns to reconstruct each primary SAE feature vector using a sparse combination of meta-latents, where the meta-features represent atomic sub-components of the primary SAE's features.

The key metric for assessing composition is the explained variance, which measures the proportion of variance in the primary SAE's decoder weights that the MetaSAE can reconstruct. A higher explained variance indicates that the MetaSAE can effectively reconstruct the primary SAE's features using shared, atomic sub-features, suggesting that the original features were composed of these simpler components. Conversely, a lower explained variance implies that the primary features are already atomic and resist decomposition into simpler constituents. In our experiments, we interpret lower MetaSAE explained variance as indicating better feature atomicity in the primary SAE.

This methodology provides an objective, quantitative measure of feature composition that complements our qualitative analyses and other atomicity metrics, offering insights into the hierarchical structure of the representations learned by different SAE variants.

H Clustering Coefficient Definition

The global clustering coefficient quantifies the tendency of nodes in a graph to form clusters. For Sparse Autoencoders (SAEs), nodes represent decoder features, and edges connect feature pairs with cosine similarity above a threshold.

Edge density, the proportion of possible edges present, is defined as:

$$\text{density} = \frac{2E}{n(n-1)},$$

where E is the number of edges and n is the number of nodes. Varying the similarity threshold adjusts the density, enabling analysis across connectivity levels.

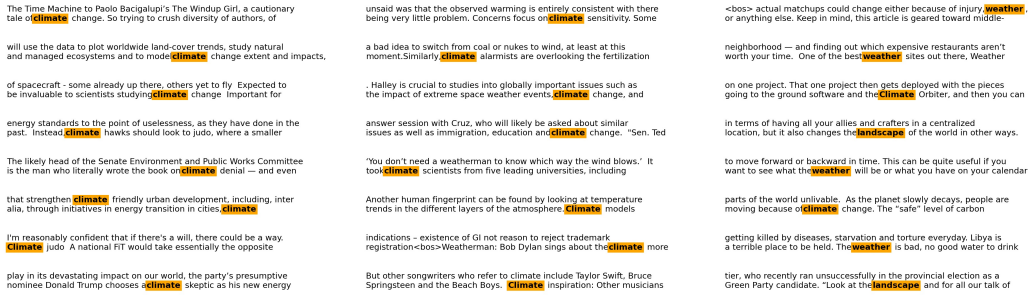
The clustering coefficient C is computed as:

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triples}},$$

where a triangle is three nodes fully connected by edges, and a connected triple is three nodes linked by at least two edges. C ranges from 0 (no clustering) to 1 (maximal clustering).

I Additional Qualitative Examples

We provide three additional examples of BatchTopK SAE features decomposed into orthogonal, atomic OrtSAE components. These cases further illustrate how OrtSAE disentangles composite concepts through orthogonality constraint.

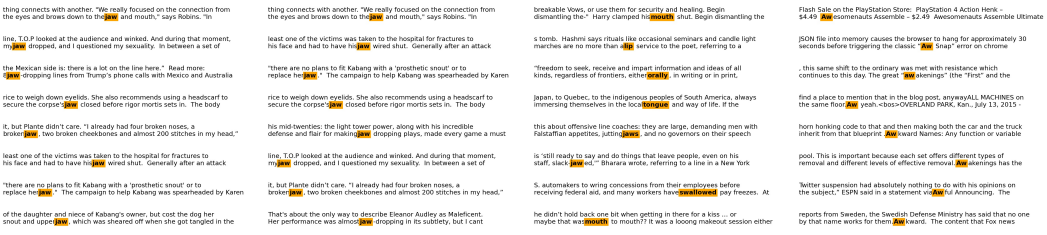


(a) BatchTopK SAE feature activating on climate-related terms.

(b) OrtSAE feature activating on on climate-related terms.

(c) OrtSAE feature activating on broader environmental and weather contexts.

Figure 8: **Decomposition of Climate-Related BatchTopK SAE Feature into OrtSAE Features.** (a) A BatchTopK SAE (L0=70) feature that activates on climate-related terms, which can be represented as a linear combination of two OrtSAE (L0=70): (b) An OrtSAE feature that activates specifically on climate-related terms. (c) An OrtSAE feature that activates on broader environmental and weather contexts.



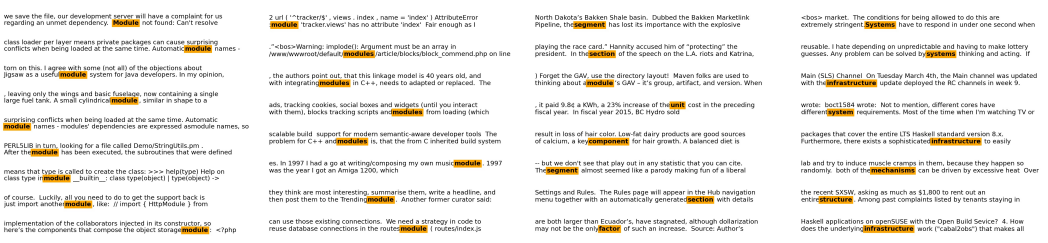
(a) BatchTopK SAE feature activating on “jaw” token.

(b) OrtSAE feature activating on “jaw” token.

(c) OrtSAE feature activating on mouth and oral concepts.

(d) OrtSAE feature activating on “aw” token.

Figure 9: **Decomposition of Jaw-Related BatchTopK SAE Feature into OrtSAE Features.** (a) A BatchTopK SAE (L0=70) feature that activates on the token “jaw”, which can be represented as a linear combination of three OrtSAE (L0=70) features: (b) An OrtSAE feature that activates specifically on the token “jaw”. (c) An OrtSAE feature that activates on mouth and oral concepts. (d) An OrtSAE feature that activates on the token “aw”.



(a) BatchTopK SAE feature activating on “module” token.

(b) OrtSAE feature activating on “module” token.

(c) OrtSAE feature activating on concepts of parts, segments.

(d) OrtSAE feature activating on concepts of complex system.

Figure 10: **Decomposition of Module-Related BatchTopK SAE Feature into OrtSAE Features.** (a) A BatchTopK SAE (L0=70) feature that activates on the token “module”, which can be represented as a linear combination of three OrtSAE (L0=70) features: (b) An OrtSAE feature that activates specifically on the token “module”. (c) An OrtSAE feature that activates on concepts of parts and segments. (d) An OrtSAE feature that activates on concepts of complex systems.