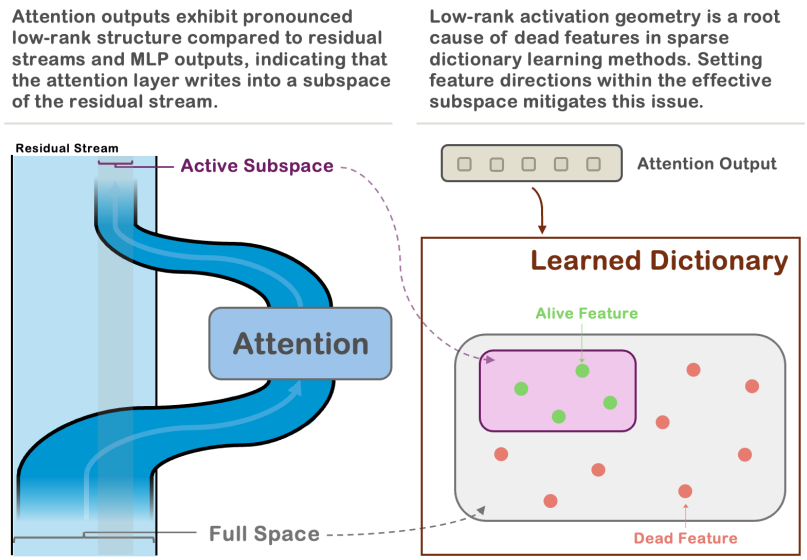
Attention Layers Add Into Low-Dimensional Residual Subspaces

Junxuan Wang **Xuyang Ge** **Wentao Shu** **Zhengfu He** **Xipeng Qiu***

[1]Shanghai Innovation Institute
[2]OpenMOSS Team, School of Computer Science, Fudan University
jxwang25@m.fudan.edu.cn
{zfhe19,xpqiu}@fudan.edu.cn



Abstract

While transformer models are widely believed to operate in high-dimensional hidden spaces, we show that attention outputs are confined to a surprisingly low-dimensional subspace, where about 60% of the directions account for 99% of the variance–a phenomenon that is induced by the attention output projection matrix and consistently observed across diverse model families and datasets. Critically, we find this low-rank structure as a fundamental cause of the prevalent dead feature problem in sparse dictionary learning, where it creates a mismatch between randomly initialized features and the intrinsic geometry of the activation space. Building on this insight, we propose a subspace-constrained training method for sparse autoencoders (SAEs), initializing feature directions into the active subspace of activations. Our approach reduces dead features from 87% to below 1% in Attention Output SAEs with 1M features, and can further extend to other sparse dictionary learning methods. Our findings provide both new insights into the geometry of attention and practical tools for improving sparse dictionary learning in large language models.

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability.

1 Introduction

Over the past years, mechanistic interpretability has shifted from a collection of proof-of-concept tools [Olsson et al., 2022, Wang et al., 2022, Meng et al., 2023, Gould et al., 2023] toward a fast-growing, scale-driven field [Templeton et al., 2024, Ameisen et al., 2025, Lindsey et al., 2025]. This transformation is driven by a wave of sparse dictionary learning methods–such as sparse autoencoders (SAEs) and their variants [Cunningham et al., 2023, Bricken et al., 2023b, Lindsey et al., 2024b], transcoders [Dunefsky et al., 2024, Ge et al., 2024], and low-rank sparse attention [He et al., 2025]–that once targeted small models but are now being pushed to larger architectures and wider model families. As these approaches scale in performance and model coverage, they provide increasingly complete and fine-grained explanations of neural network behavior [Lindsey et al., 2024a, Gao et al., 2024].

However, scaling these approaches presents practical difficulties [Templeton et al., 2024, Gao et al., 2024, Mudide et al., 2025]. As models and feature dictionaries grow, the number of parameters increases rapidly, driving up computational costs. At the same time, the prevalence of dead features leads to substantial waste in computation and memory [Templeton et al., 2024, Kissane et al., 2024], limiting the efficiency of interpretability methods. In this work, we identify **low-rank activation structure as a major driver of dead features** (Section 5.1).

In Section 4, we show that **attention outputs exhibit a remarkably strong low-rank structure** compared to multilayer perceptron (MLP) outputs and residual streams. Through singular value decomposition and intrinsic dimension analyses [Guth et al., 2023, Staats et al., 2025], we demonstrate that this phenomenon holds universally across layers, model families, and datasets, which is consistent with the universality hypothesis [Olah et al., 2020, Chughtai et al., 2023, Gurnee et al., 2024, Wang et al., 2025]. We further trace the origin of this low-rank structure to the anisotropy of the output projection matrix W^O, which compresses the multi-head outputs into a lower-dimensional subspace.

In Section 5, we investigate how the low-rank nature of attention outputs interacts with SAE training. By evaluating the full suite of open-source SAEs from *LlamaScope* [He et al., 2024], we show that low intrinsic dimension strongly correlates with the number of dead features, suggesting a mismatch between random initialization and the low-dimensional geometry of the activations. To address this, we propose *Active Subspace Initialization*, which aligns SAE features with the active subspace of activations, **substantially reducing dead features while improving reconstruction**. Following Lindsey et al. [2024a] and Gao et al. [2024], we conduct scaling experiments, which further reveal that ASI achieves superior reconstruction across feature counts, and when combined with SparseAdam[2], it achieves the best reconstruction in large scale and reduces dead features from 87% to below 1% in Attention Output SAEs with 1M features trained on Llama-3.1-8B [Dubey et al., 2024].

Furthermore, we show that **Active Subspace Init can generalize to sparse replacement models** [He et al., 2025, Dunefsky et al., 2024, Ameisen et al., 2025] (Section 5.4). When applied to other sparse dictionary learning methods, our initialization procedure systematically reduces the prevalence of dead parameters across architectures.

2 Related Work

2.1 Low-Rankness in Attention Mechanisms

Prior work has investigated various notions of "low-rankness" within attention mechanisms.: low-rank approximation of attention patterns [Wang et al., 2020, Tay et al., 2020, Raganato et al., 2020], low-rank parameterization for model compression [Noach and Goldberg, 2020, Hu et al., 2022], and the inherent low-rank bottleneck in single-head outputs [Bhojanapalli et al., 2020].

It is important to note that our perspective differs from these prior lines of work. We demonstrate that the multi-head self-attention outputs naturally exhibit a low-rank structure, revealing a distinct and under-explored phenomenon.

[2]https://docs.pytorch.org/docs/stable/generated/torch.optim.SparseAdam.html

2.2 Linear Representation Hypothesis and Sparse Dictionary Learning Methods

The linear representation hypothesis posits that high-level concepts correspond to linear directions in representation space [Arora et al., 2018, Olah et al., 2020, Elhage et al., 2022, Park et al., 2024]. Building on this view, a series of sparse dictionary learning methods have been proposed as interpretability tools, including sparse autoencoders and their variants [Cunningham et al., 2023, Bricken et al., 2023b, Lindsey et al., 2024b], transcoders [Dunefsky et al., 2024, Ge et al., 2024], and low-rank sparse attention [He et al., 2025]. These approaches aim to decompose activations into combinations of sparsely activated features, while adopting different strategies, depending on their interpretability objectives, to predict or approximate feature activations. Importantly, this hypothesis has been shown to hold across a wide range of model scales [Templeton et al., 2024, Lieberum et al., 2024, He et al., 2024], architectures [Wang et al., 2025], and modalities [Abdulaal et al., 2024].

2.3 Dead Features in Sparse Dictionary Learning Methods

A persistent challenge in sparse dictionary learning methods is the emergence of *dead features*[3] [Templeton et al., 2024, Kissane et al., 2024], which are also referred to as *dead units* in sparse replacement models [Dunefsky et al., 2024, Ge et al., 2024, He et al., 2025]. These features contribute nothing to reconstruction quality, wasting parameters and computation. Existing approaches to mitigate this issue rely on auxiliary loss terms [Gao et al., 2024, Conerly et al., 2025] or resampling strategies [Bricken et al., 2023b] to encourage feature usage.

3 Preliminaries

3.1 Multi-Head Self-Attention and Notations

We consider a Transformer layer with multi-head self-attention (MHSA). Given input representations $X \in \mathbb{R}^{n\times d}$, where n is the number of tokens and d is the hidden size, each attention head i computes:

$$Q_i = XW_i^Q, \quad K_i = XW_i^K, \quad V_i = XW_i^V, \quad W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d\times d_h},$$

where $d_h = d/H$ is the dimensionality of each head, and H is the total number of heads. The attention weights and head outputs are then given by:

$$A_i = \mathrm{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d_h}}\right), \quad Z_i = A_i V_i.$$

Let $Z = \mathrm{Concat}[Z_1, \ldots, Z_H] \in \mathbb{R}^{n\times d}$ denote the concatenated output of all attention heads [Nanda and Bloom, 2022]. The final attention output is computed by applying the output projection:

$$O = ZW^O, \quad W^O \in \mathbb{R}^{d\times d}.$$

This formulation shows that O can be viewed as the sum of low-dimensional outputs from each head, projected into the residual stream space. O serves as the attention block's contribution to the residual stream.

3.2 TopK Sparse Autoencoders

In this work, we adopt the TopK sparse autoencoder (TopK SAE) variant introduced by Gao et al. [2024]. Unlike standard SAEs that impose an ℓ_1 penalty, TopK SAE enforces exact sparsity by keeping only the top-k activations in the latent representation for each input. Formally, given an input vector $x \in \mathbb{R}^d$, the encoder produces

$$z = \mathrm{TopK}(W_e x + b_e),$$

where $\mathrm{TopK}(v)$ sets to zero all but the largest k entries of v. The decoder then reconstructs

$$\hat{x} = W_d z + b_d.$$

[3]Following Bricken et al. [2023b], we define a feature as dead if it never activates over 10 million tokens in this paper.

The model is trained to minimize the reconstruction loss, optionally augmented with an auxiliary penalty to prevent dead latents:

$$\mathcal{L}_{\text{TopK-SAE}} = \|x - \hat{x}\|_2^2 + \alpha \cdot \mathcal{L}_{\text{aux}},$$

where $\mathcal{L}_{\text{aux}}$ is an optional term designed to penalize latents that never activate over a training period, and α balances reconstruction fidelity and latent utilization.

4 Low-Rank Structure of Attention Outputs

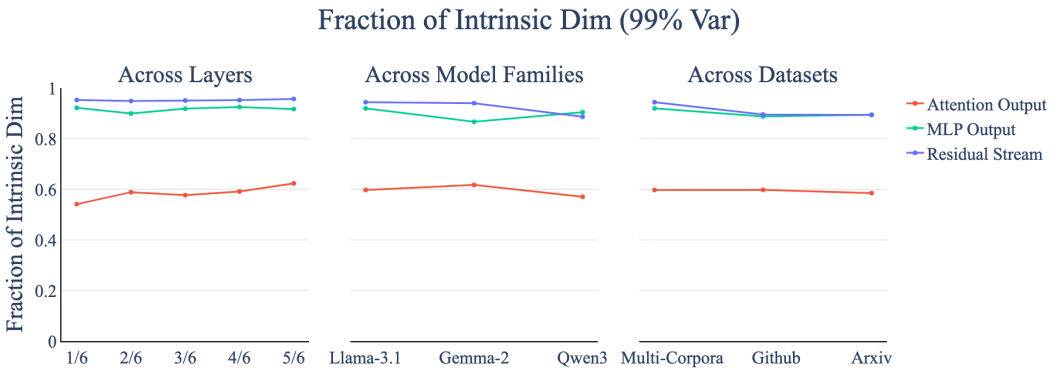


Figure 1: Across layers, model families and datasets, attention outputs exhibit dramatically lower intrinsic dimensions (details in Section 4.1) than residual streams and MLP outputs, showing that the attention layer writing into a low dimensional subspace of residual stream is a universal phenomenon.

We begin by presenting our central empirical finding: in Transformer models, attention outputs consistently display the strongest low-rank structure compared to MLP outputs and residual streams. As shown in Figure 1, attention outputs have a significantly lower intrinsic dimension. This phenomenon is remarkably robust, holding across different intermediate layers, model families and datasets. These observations highlight that the attention block modifies a subspace of the residual stream, while the MLP operates nearly on the full space.

4.1 Quantifying Low-Rankness with Relative Singular Values

We consider activation matrix $\widetilde{A} \in \mathbb{R}^{n \times d}$, where each row corresponds to one token's activation vector, and n is the number of data points, while d is the dimensionality of the activation space (e.g., the hidden size of the model). Unless otherwise specified, $\widetilde{A}$ refers to the mean-centered activations. We refer readers to Appendix B for more details of activations sources.

To quantify the rank of data, we perform singular value decomposition (SVD) on $\widetilde{A}$:

$$\widetilde{A} = U \Sigma V^\top,$$

where $U \in \mathbb{R}^{n \times r}$, $V \in \mathbb{R}^{d \times r}$, and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r \times r}$ contains the singular values $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$, with $r = \text{rank}(\widetilde{A})$. The squared singular values σ_i^2 indicate the amount of variance captured along each principal direction.

To analyze the intrinsic dimension of these activations, we compute the smallest integer k such that:

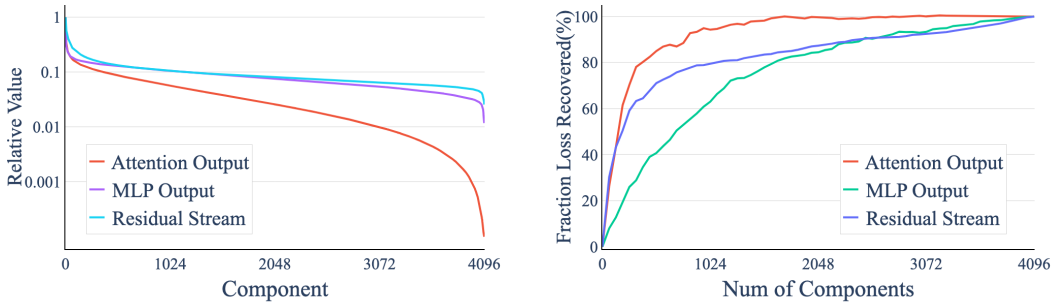
$$\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \geq \tau,$$

for a given threshold $\tau \in (0, 1)$. Since SVD yields the optimal low-rank approximation in terms of reconstruction error, this k provides a principled way to assess how concentrated the activations are in a low-dimensional subspace[4] (Figure 1). We further compute the fraction of delta downstream

[4]We use 0.99 as the threshold in main text, results of some other thresholds are provided in Appendix D, with no influence to the conclusion.

loss recovered by different number of components. These metrics complement our central findings, offering a numerical characterization of low-rankness.

4.2 Empirical Evidence of Low-Rank Structure



(a) (b)

Figure 2: Results for $1M$ samples from SlimPajama [Soboleva et al., 2023] fed into Llama-3.1-8B. **(A)** The attention outputs are the most low-rank, as indicated by the sharpest decay in singular values. **(B)** The fraction of loss recovered by different number of components, selected in descending order of singular values.

We draw our findings from three lines of evidence:

Low Intrinsic Dimension of Attention Outputs Attention outputs exhibit significantly lower intrinsic dimensionality compared to other activation types, with an intrinsic dimension of around 60% of the total dimensionality. In contrast, MLP outputs and the residual streams show much higher intrinsic dimensions above 90% (Figure 1).

Rapid Spectral Decay in Attention Outputs Attention outputs exhibit a sharply decaying singular value spectrum. This is quantitatively evidenced by the number of components retaining significant energy: only 1174 singular values exceed 5% of the maximum value in attention outputs, compared to 3276 for MLP outputs and 3943 for the residual streams (Figure 2a).

Efficient Downstream Loss Recovery Attention outputs demonstrates superior dimensional efficiency in downstream task performance. Compared to zero ablation, attention outputs requires only 25.0% of dimensions to recover 95% of the downstream loss. This contrasts sharply with MLP outputs and residual streams, which require 78.1% and 85.9% of dimensions respectively to achieve the same recovery level (Figure 2b).

More results of these metrics across different layers, models and datasets are shown in Appendix C.

4.3 Low-Rankness of Attention Outputs Results from the Output Projection Matrix

Among all activation types, attention outputs consistently exhibits the most rapid spectral decay. To investigate whether this low-rank structure originates from the attention heads (Z), the output projection matrix (W^O), or their interaction, we perform a decomposition-based analysis.

Recall that the attention output is computed as $O = ZW^O$, where $Z \in \mathbb{R}^{n \times d}$ is the concatenated output of all attention heads, and $W^O \in \mathbb{R}^{d \times d}$ is a learned linear projection. To understand how the variance in O is shaped, we analyze the variance of O along an arbitrary unit direction $\hat{e} \in \mathbb{R}^d$, given by:

$$\mathrm{Var}(O\hat{e}) = \mathrm{Var}(ZW^O\hat{e}).$$

This expression highlights that the variance along e is determined by two factors: the norm of $W^O e$ and the variance of Z projected onto the direction $W^O\hat{e}$. Specifically, we can rewrite the variance as:

$$\mathrm{Var}(O\hat{e}) = \mathrm{Var}(Z\hat{v}) \cdot \|v\|_2^2 \quad \text{where} \quad v = W^O\hat{e},\ \hat{v} = \frac{v}{\|v\|_2}.$$

We refer to $\mathrm{Var}(Zv)$ as the contribution of Z, capturing how much variance the head output Z provides in that direction, and $\|v\|_2^2$ as the contribution of W^O, measuring how much the output projection W^O scales or suppresses that direction.

We compute and visualize both quantities across a set of directions aligned with the right singular vectors of attention output, as shown in Figure 3. Our analysis reveals that the low-rank structure of attention outputs O arises primarily from the anisotropy of W^O, which heavily compresses the output space into a lower-dimensional subspace. From a mechanistic perspective, an intuitive way to see this is that although each attention head contributes a d_{head}-dimensional subspace, the superposition of heads [Jermyn et al., 2024, He et al., 2025] inherently leads to overlaps among these subspaces. We note the output of the i^{th} head as head_i. Consequently, the dimension of the MHSA output satisfies

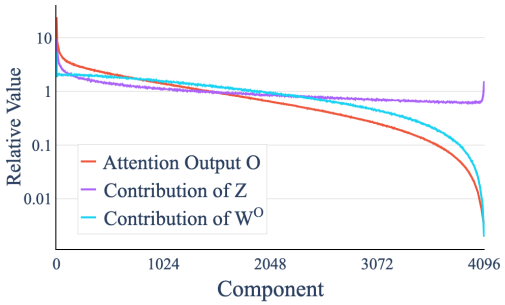


Figure 3: Decomposition of variance in attention output O. We analyze the contributions of the *concatenated head outputs* Z and the *projection matrix* W^O to the variance along each principal component of O ($=ZW^O$). All values are normalized to a common scale. The curve of O closely follow that of Z for the top components, whereas the downward trend of attention output at the tail is mainly due to W^O contribution.

$$\dim\left(\bigcup_i \text{span}(\text{head}_i)\right) \leq \sum_i \dim\big(\text{span}(\text{head}_i)\big) = d_{\text{head}} \cdot n_{\text{head}} \ (= d_{\text{model}} \text{ in standard MHSA}).$$

5 Active Subspace Initialization for Sparse Autoencoders

5.1 Empirical Correlation Between Low-Rank Structure and Dead Features

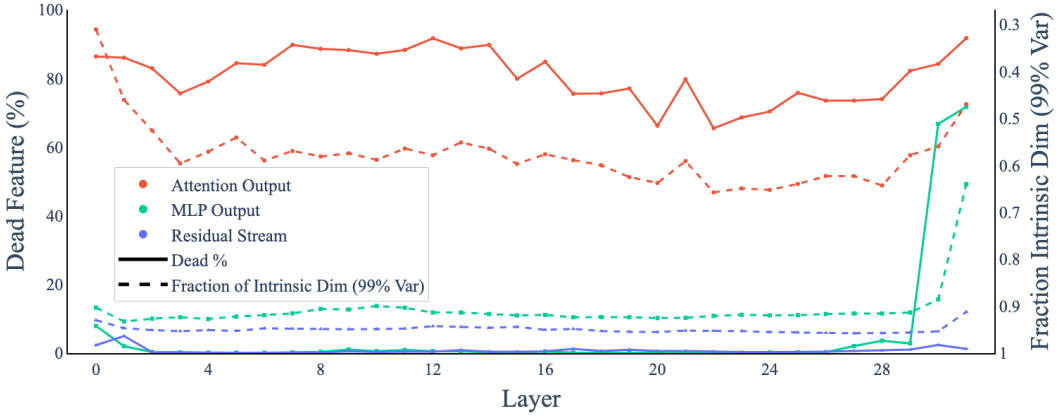


Figure 4: The number of dead features and the intrinsic dimension of each layer in Llama-3.1-8B, shows a surprising consistency: activations with lower intrinsic dimensions have more dead features.

To explore how low-rankness affects the interpretability of attention, we use the same framework and data as the original study to evaluate the LlamaScope SAEs [He et al., 2024][5], which provide a complete set of SAEs trained on attention output, MLP output, and residual stream. We find that the number of dead features is strongly related to intrinsic dimensions, as shown in Figure 4. This

[5]Another prominent open-source SAEs, GemmaScope [Lieberum et al., 2024], train their attention SAEs on Z rather than attention output.

observation suggests that dead features may stem from a fundamental mismatch between the SAE's randomly initialized weights and the geometry of the activation space.

5.2 Active Subspace Initialization for Sparse Autoencoders

To investigate this co-occurrence, we train SAEs on the attention output of Llama-3.1-8B at layer 15, initializing the SAE feature directions in the subspace spanned by the top d_{init} singular vectors of the activations. As shown in Figure 5, we find that constraining the initialization to a lower-dimensional active subspace substantially decreases the number of dead features.

Based on this observation, we propose *Active Subspace Initialization* (ASI), a lightweight and generalizable strategy for scaling SAEs to high capacities. Let d be the input dimension, h the SAE hidden dimension, and n the number of data points. Given activation matrices $\widetilde{A} \in \mathbb{R}^{n \times d}$, we compute the SVD:

$$\widetilde{A} = U \Sigma V^\top, \quad V \in \mathbb{R}^{d \times d}.$$

We select the top d_{init} right singular vectors to form the active subspace:

$$V_{\text{active}} = V_{:,:d_{\text{init}}} \in \mathbb{R}^{d \times d_{\text{init}}}.$$

We then initialize the SAE weights directly in this subspace:

$$W_d \in \text{span}(V_{\text{active}}), \quad W_e = W_d^\top.$$

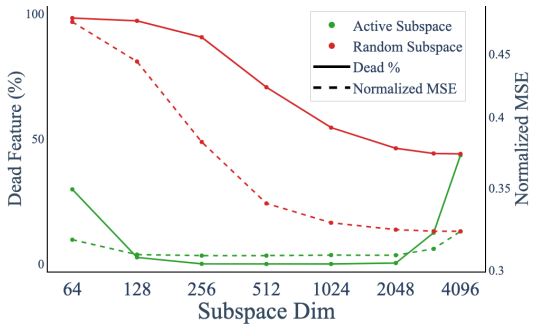
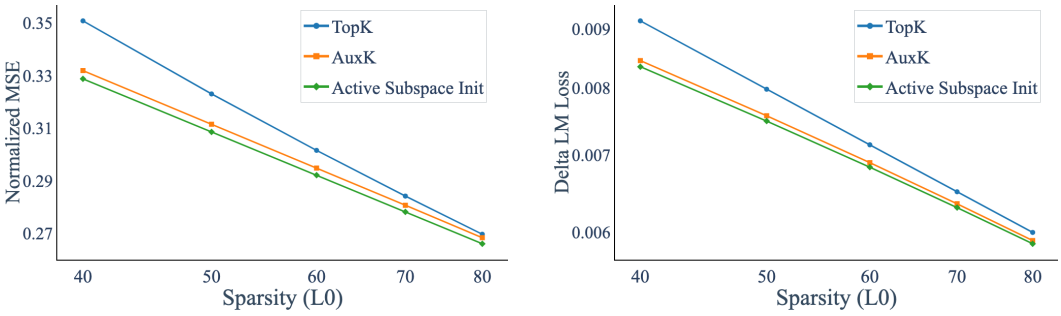


Figure 5: Proportion of dead features and normalized MSE across different subspace dimensions. Random subspaces are used as the baseline, whereas only initialization with the active subspace yields improvement.

Using **Active Subspace Initialization** offers several benefits:



(a) (b)

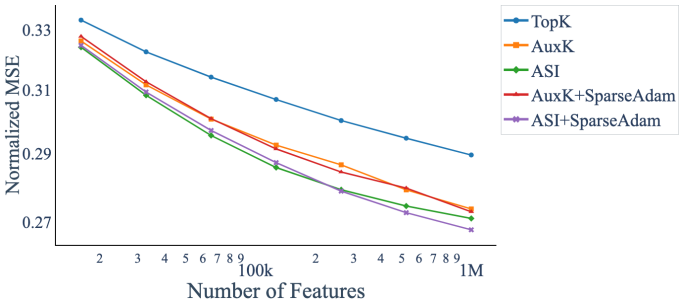
Figure 6: At a fixed number of features ($n = 32768$), *Active Subspace Init* achieves a better reconstruction-sparsity trade-off than Base TopK and AuxK. A similar trend is observed in its impact on Language Model Downstream Loss. **Note:** The improvement in downstream loss is less pronounced than that in reconstruction, likely because *Active Subspace Init* allocates more features to the active subspace, which implicitly enhances reconstruction quality.

Enhanced Sparsity-Reconstruction Frontier Without Additional Compute It empirically outperforms TopK on the sparsity-reconstruction frontier. It achieves slightly better results compared to the auxiliary loss approach while introducing no additional computational overhead (Figure 6).

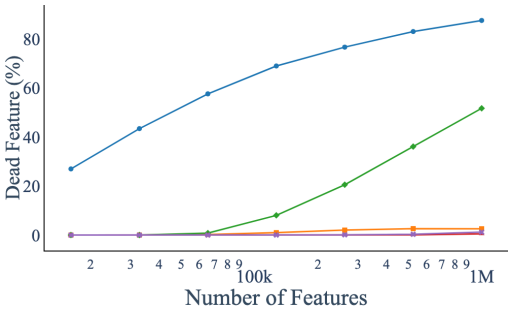
Optimal Scaling Characteristics Our approach demonstrates optimal scaling behavior across various model configurations. On SAEs with an expansion factor of 32 times or more, it can achieve the same or even better performance as the auxiliary loss approach with only half the number of parameters (Section 5.3).

Broad Architectural Applicability The technique maintains applicability to diverse activation functions and architectural variants, as it operates directly on the intrinsic properties of activations. This generalizability is further explored in Section 5.4.

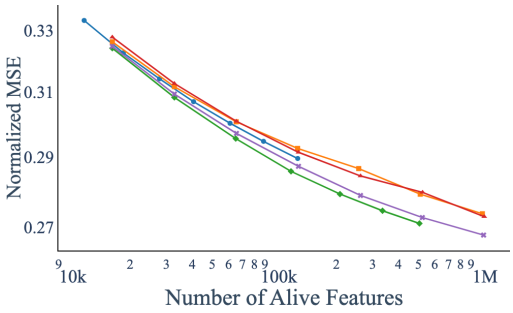
5.3 Scaling Laws



(a) Loss vs. Number of Parameters.



(b) Number of Dead vs. Number of Parameters.



(c) Loss vs. Number of Alive Parameters.

Figure 7: Scaling results of TopK SAEs and their variants enhanced with *AuxK*, *Active Subspace Init*, and *SparseAdam*–all trained on attention output from Llama-3.1-8B. (A) Loss at convergence across different feature counts: Active Subspace Init consistently achieves lower reconstruction error than TopK and AuxK. Active Subspace Init with SparseAdam achieves the best at large scale. (B) Dead features: Active Subspace Init reduces dead features compared to TopK, but still retains many at extremely large scales. Enhanced with SparseAdam, dead features can be reduced to less than 1%. (C) Loss across different number of alive features: Active Subspace Init achieves the most efficient utilization of alive features, while AuxK shows the lowest efficiency. Details in Section 5.3.

To understand how our method scales, we evaluate performance as the number of SAE features increases from 16K to 1M, keeping other hyperparameters fixed (details in Appendix E). Figure 7 summarizes the results.

Active Subspace Init improves reconstruction. As shown in Figure 7a, the normalized MSE follows a smooth power-law decay with increasing feature count. Across all scales, Active Subspace Init consistently outperforms baseline TopK and AuxK, achieving superior reconstruction at equivalent feature counts.

Caveat: some dead features remain at extremely large scales in Active Subspace Init. Figure 7b shows that, when scaling to extremely large feature counts, Active Subspace Init produces more dead features than AuxK. However, reconstruction performance remains better, indicating that the revived features from AuxK contribute little to actual reconstruction quality(Figure 7c).

Use Active Subspace Init with SparseAdam further improves performance. Prior work [Bricken et al., 2023a] identified *stale momentum* as a key factor in dead feature formation. Building on this insight, we propose using **SparseAdam**, an optimizer specifically designed for sparse activation settings. By updating only the moments and parameters corresponding to non-zero gradients, SparseAdam naturally avoids stale momentum and thus mitigates the dead feature issue. As shown

in Figures 7a, 7b, combining Active Subspace Init with SparseAdam substantially reduces dead features while reaching the lowest reconstruction error. While orthogonal to our initialization method, this choice provides a practical complement that further stabilizes training when scaling SAEs to very large capacities. We discuss more about *stale momentum* and **SparseAdam** in Appendix A.

5.4 Generalize to Sparse Replacement Models

Recent work by He et al. [2025] reports that Lorsa, a sparse replacement model for attention layers, exhibits a high proportion of dead parameters. We hypothesize that the low-rank structure of attention outputs contributes significantly to this phenomenon.

To test this, we apply **Active Subspace Initialization** to Lorsa[6]. This modification reduces the proportion of dead parameters from 68.4% to 40.5% under identical hyperparameter settings (Table 1), while also slightly improving reconstruction error.

Table 1: Effect of Active Subspace Initialization on reducing dead parameters in attention replacement model.

Method	**Vanilla**	**Active Subspace Init**
Dead Parameters (%)	68.4	**40.5**
Normalized MSE	0.130	**0.121**

These results indicate that **Active Subspace Initialization** provides an effective strategy for mitigating dead parameters when training sparse replacement models on low-rank activations. We posit that complete elimination of dead heads may require additional mechanisms, such as the **tied initialization** used in SAEs to ensure alignment between feature encoding and decoding methods[7]. This approach has been shown to be crucial for reducing dead features in SAEs [Gao et al., 2024], and its absence in Lorsa may limit further improvement.

6 Conclusion

We identified the low-rank structure of attention outputs as a fundamental property of Transformer models and a key cause of dead features in sparse dictionary learning. Our proposed *Active Subspace Initialization* method addresses this by aligning SAE features with the intrinsic geometry of activations, reducing dead features while improving reconstruction quality. The approach generalizes beyond SAEs to sparse replacement models.

References

Ahmed Abdulaal, Hugo Fry, Nina Montaña Brown, Ayodeji Ijishakin, Jack Gao, Stephanie L. Hyland, Daniel C. Alexander, and Daniel C. Castro. An x-ray is worth 15 features: Sparse autoencoders for interpretable radiology report generation. *CoRR*, abs/2410.03334, 2024. doi: 10.48550/ARXIV.2410.03334. URL `https://doi.org/10.48550/arXiv.2410.03334`.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL `https://transformer-circuits.pub/2025/attribution-graphs/methods.html`.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Trans. Assoc. Comput. Linguistics*, 6:483–495, 2018. doi: 10.1162/TACL_A_00034. URL `https://doi.org/10.1162/tacl_a_00034`.

[6]Specifically, we initialize the feature directions (corresponding to W^O in Lorsa) within the active subspace. Other hyperparameters are the same as He et al. [2025].

[7]"Match" means encoder can be initialized to predict relatively accurate feature activation values for decoder.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 864–873. PMLR, 2020. URL http://proceedings.mlr.press/v119/bhojanapalli20a.html.

Trenton Bricken, Xander Davies, Deepak Singh, Dmitry Krotov, and Gabriel Kreiman. Sparse distributed memory is a continual learner. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023a. URL https://openreview.net/forum?id=JknGeelZJpHP.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023b. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 6243–6267. PMLR, 2023. URL https://proceedings.mlr.press/v202/chughtai23a.html.

Tom Conerly, Hoagy Cunningham, Adly Templeton, Jack Lindsey, Basil Hosmer, and Adam Jermyn. Circuits updates - january 2025. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/january-update/index.html#DL.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *CoRR*, abs/2309.08600, 2023. doi: 10.48550/ARXIV.2309.08600. URL https://doi.org/10.48550/arXiv.2309.08600.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. *CoRR*, abs/2406.11944, 2024. doi: 10.48550/ARXIV.2406.11944. URL https://doi.org/10.48550/arXiv.2406.11944.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

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. *CoRR*, abs/2406.04093, 2024. doi: 10.48550/ARXIV.2406.04093. URL https://doi.org/10.48550/arXiv.2406.04093.

Xuyang Ge, Fukang Zhu, Wentao Shu, Junxuan Wang, Zhengfu He, and Xipeng Qiu. Automatically identifying local and global circuits with linear computation graphs. *CoRR*, abs/2405.13868, 2024. doi: 10.48550/ARXIV.2405.13868. URL https://doi.org/10.48550/arXiv.2405.13868.

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor heads: Recurring, interpretable attention heads in the wild, 2023. URL https://arxiv.org/abs/2312.09230.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in GPT2 language models. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=ZeI104QZ8I.

Florentin Guth, Brice Ménard, Gaspar Rochette, and Stéphane Mallat. A rainbow in deep network black boxes. *CoRR*, abs/2305.18512, 2023. doi: 10.48550/ARXIV.2305.18512. URL https://doi.org/10.48550/arXiv.2305.18512.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *CoRR*, abs/2410.20526, 2024. doi: 10.48550/ARXIV.2410.20526. URL https://doi.org/10.48550/arXiv.2410.20526.

Zhengfu He, Junxuan Wang, Rui Lin, Xuyang Ge, Wentao Shu, Qiong Tang, Junping Zhang, and Xipeng Qiu. Towards understanding the nature of attention with low-rank sparse decomposition. *arXiv preprint arXiv:2504.20938*, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Adam Jermyn, Chris Olah, and Tom Conerly. Circuits updates - january 2024. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/jan-update/index.html#attn-superposition.

Connor Kissane, Robert Krzyzanowski, Arthur Conmy, and Neel Nanda. Sparse autoencoders work on attention layer outputs. Alignment Forum, 2024. URL https://www.alignmentforum.org/posts/DtdzGwFh9dCfsekZZ.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca D. Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *CoRR*, abs/2408.05147, 2024. doi: 10.48550/ARXIV.2408.05147. URL https://doi.org/10.48550/arXiv.2408.05147.

Jack Lindsey, Tom Conerly, Adly Templeton, Jonathan Marcus, and Tom Henighan. Circuits updates - april 2024. *Transformer Circuits Thread*, 2024a. URL https://transformer-circuits.pub/2024/april-update/index.html#scaling-laws.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. *Transformer Circuits Thread*, 2024b. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

Anish Mudide, Joshua Engels, Eric J. Michaud, Max Tegmark, and Christian Schroeder de Witt. Efficient dictionary learning with switch sparse autoencoders, 2025. URL https://arxiv.org/abs/2410.08201.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Matan Ben Noach and Yoav Goldberg. Compressing pre-trained language models by matrix decomposition. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 884–889. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.AACL-MAIN.88. URL https://doi.org/10.18653/v1/2020.aacl-main.88.

Christopher Olah, Lilian Pratt-Hartmann, et al. Zoom in: An introduction to circuits. *Distill*, 2020.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=UGpGkLzwpP.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. Fixed encoder self-attention patterns in transformer-based machine translation. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 556–568. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.49. URL https://doi.org/10.18653/v1/2020.findings-emnlp.49.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Max Staats, Matthias Thamm, and Bernd Rosenow. Small singular values matter: A random matrix analysis of transformer models, 2025. URL https://arxiv.org/abs/2410.17770.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention in transformer models. *CoRR*, abs/2005.00743, 2020. URL https://arxiv.org/abs/2005.00743.

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*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Junxuan Wang, Xuyang Ge, Wentao Shu, Qiong Tang, Yunhua Zhou, Zhengfu He, and Xipeng Qiu. Towards universality: Studying mechanistic similarity across language model architectures. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=2J18i8T0oI.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022. URL https://arxiv.org/abs/2211.00593.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768, 2020. URL https://arxiv.org/abs/2006.04768.

Author Contributions

Junxuan Wang and **Zhengfu He** co-discovered the low-rank structure in attention outputs and its correlation with dead features.

Junxuan Wang proposed the Active Subspace Initialization method and the use of SparseAdam for SAE training, conducted all experiments.

Xuyang Ge, **Zhengfu He**, and **Wentao Shu** made core contributions to the SAE training codebase infrastructure.

Zhengfu He helped to edit the manuscript.

Xipeng Qiu supervised the research and provided research guidance.

A Stale Momentum as Another Root Cause of Dead Features

Recent work by Bricken et al. [2023a] identifies *stale momentum* as a key cause to dead feature formation. Specifically, when a feature remains inactive over training steps, its associated optimizer momentum continues to accumulate. If the feature activates, the stale momentum results in disproportionately large updates, destabilizing training and potentially suppressing that feature permanently.

To directly address this, we adopt *SparseAdam*, an optimizer tailored for sparse activation settings, designed for more efficient use of compute and memory. SparseAdam updates both parameters and moments only when the corresponding feature is active. This could effectively prevent the harmful accumulation of stale momentum. Empirically, we observe that this change substantially reduces the rate of dead feature formation in large-scale SAE training. We believe that this is a core technique for scaling sparse dictionary methods, as stale momentum is a common problem for them.

B Activation Sources

The spectral characteristics of activations vary substantially across model architectures, datasets, and positional contexts. Below, we describe the experimental configurations used to support a broad and representative analysis.

Models We study three large language models of different families–`Llama-3.1-8B`[8], `Qwen3-8B`[9], and `Gemma-2-9B`[10]–all based on the Transformer architecture. This allows us to assess the robustness of spectral properties under varying model training configurations.

Datasets To investigate how dataset diversity affects activation spectra, we select three datasets with varying linguistic and domain characteristics: (1) `SlimPajama`, an English corpus comprising web text, books, and other sources; (2) `RedPajamaGithub`, a large-scale code corpus; and (3) `CCI3-Data`, a Chinese dataset with broad domain coverage.

Activation Positions Unless otherwise specified, activations are extracted from intermediate layers. For example, in `LLaMA-3.1-8B` (32 layers), we use activations from layer 15 (zero-indexed). We analyze four types of activations: (1) attention output, (2) MLP output, and (3) residual stream (post

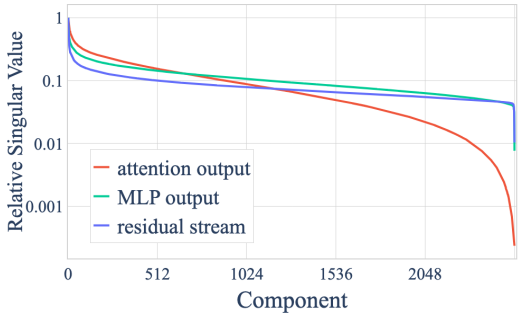
[8]https://huggingface.co/meta-llama/Llama-3.1-8B

[9]https://huggingface.co/Qwen/Qwen3-8B

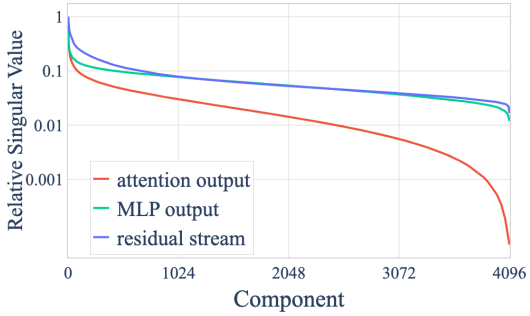
[10]https://huggingface.co/google/gemma-2-9b

layer). For each type, we collect 10 million activation vectors. We empirically verify that this sample size is sufficient to produce stable and reproducible spectral analyses.

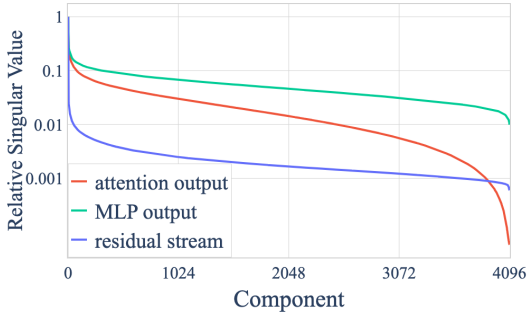
C More Low-Rank Result Across Different Models and Datasets



(a) Activation spectra for many samples from SlimPajama fed into pythia-2.8b.



(b) Activation spectra for many samples from CCI3-Data fed into Qwen3-8B.



(c) Activation spectra for many samples from RedPajamaGithub fed into Qwen3-8B.

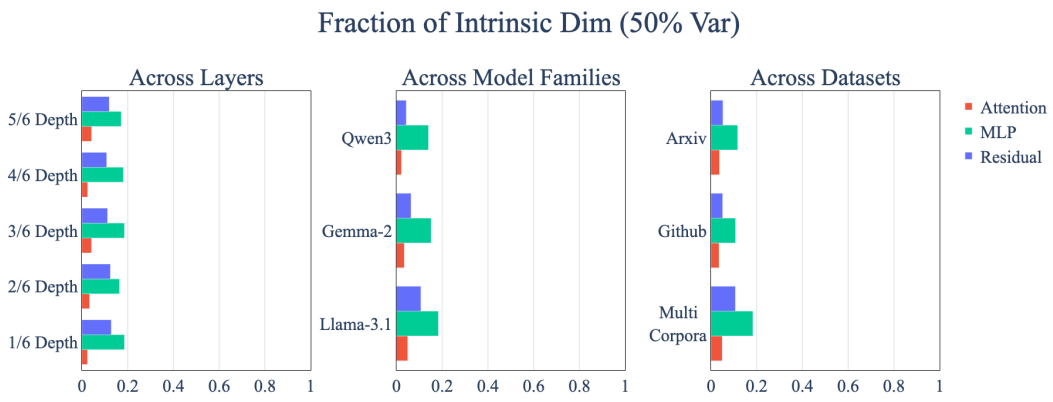
Figure 8

We present relative singular values and fraction of loss recovered for some other model-dataset pairs in Figure 8. Models include pythia-2.8b[11]. Datasets include RedPajamaGithub[12] and CCI3-Data[13]

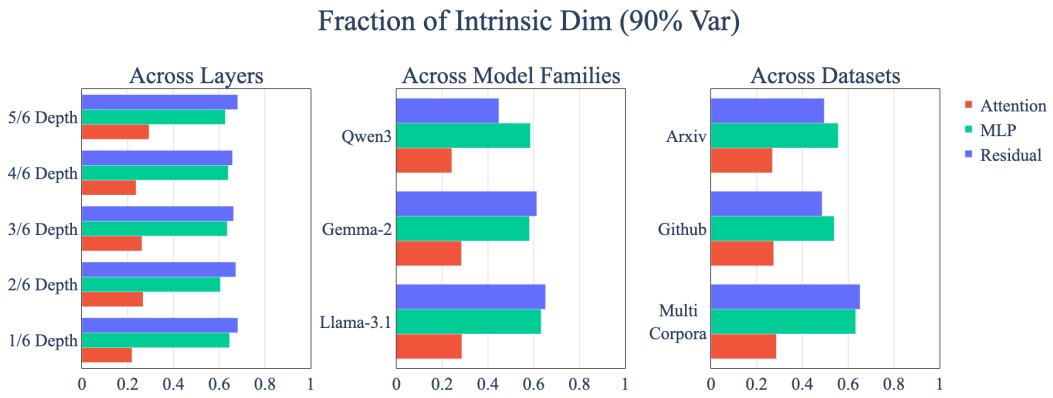
D Different Choose of Variance Threshold for Intrinsic Dimension

We use 0.99 as the variance threshold in the main text. We show other threshold chose make no influence to the conclusion in Figure 9. Attention outputs show low-rank structure consistently.

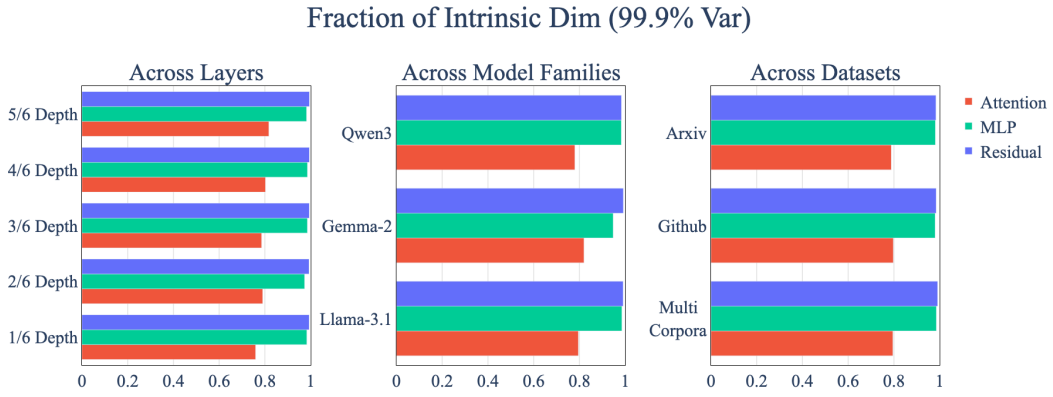
[11] EleutherAI/pythia-2.8b
[12] https://huggingface.co/datasets/cerebras/SlimPajama-627B
[13] https://huggingface.co/datasets/BAAI/CCI3-Data



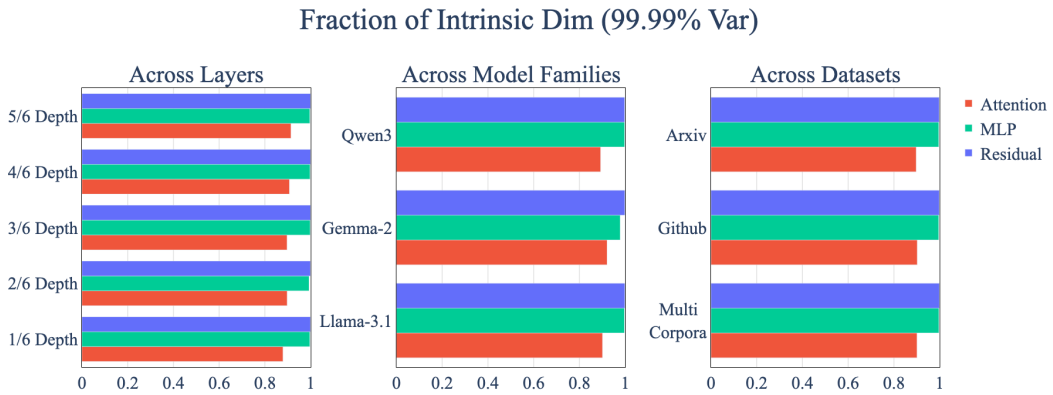
(a) Intrinsic dimension with threshold 0.5.



(b) Intrinsic dimension with threshold 0.9.



(c) Intrinsic dimension with threshold 0.999.



(d) Intrinsic dimension with threshold 0.9999.

Figure 9: Comparison of intrinsic dimensions across different variance thresholds.

E SAE Training Details

E.1 Collecting Activations

We truncate each document to 1024 tokens and prepend a <bos> token to the beginning of each document. During training, we exclude the activations corresponding to the <bos> and <eos> tokens.

It has been observed that activations from different sequence positions within the same document are often highly correlated and may lack diversity. To mitigate this issue, it is common to introduce randomness into the training data. Our shuffling strategy maintains a buffer that is reshuffled whenever the buffer is refilled.

E.2 Initialization and Optimization

The decoder columns $W^{dec}_{:,i}$ are initialized uniformly and normalized to achieve the lowest initial reconstruction loss. We find that the specific initialization norm has little impact, as long as in a reasonable scope. For example, initializing $W^{dec}_{:,i}$ uniformly with a fixed bound, as in Conerly et al. [2025], yields similar results. The encoder weights W^{enc} are initialized as the transpose of W^{dec}, while both the encoder bias b^{enc} and decoder bias b^{dec} are set to zero.

This initialization scheme ensures that the SAE begins training with an almost zero reconstruction loss. Such initialization has been widely observed to benefit SAE training.

We train SAEs using the Adam and SparseAdam optimizers, both with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.

E.3 Fixed Hyperparameters in Scaling Law

Model, Dataset, Layer, Pos Llama-3.1-8B, SlimPajama, 15(index start at 0), attention output.

Sparsity We empirically set $k = 50$ for a reasonable sparsity in scaling laws.

Batch Size We empirically set $batchsize = 4096$, which belows the critical batch size.

Learning Rate The learning rate for **Adam** and **SparseAdam** is sweeped separately in [1e−5, 2e−5, 4e−5, 6e−5, 8e−5, 1e−4, 2e−4, 4e−4], and we ultimately use 4e−5 for **Adam** and 6e−5 for **SparseAdam**.

AuxK We follow Gao et al. [2024] to set auxiliary loss coefficient α as $\frac{1}{32}$. We sweep the k_{aux} in [256, 512, 1024, 2048] and finally choose 512.

Dimension of Subspace for SAE Initialization As shown in Figure 5, d_{init} is a hyperparameter with a wide range of sub-optimal value space (from 256 to 2048). We use 768 for all experiments.

Total Tokens We use $2.5B$ tokens for each SAE training.