

ATTENTION LAYERS ADD INTO LOW-DIMENSIONAL RESIDUAL SUBSPACES

Junxuan Wang Xuyang Ge Wentao Shu Zhengfu He Xipeng Qiu*

¹Shanghai Innovation Institute

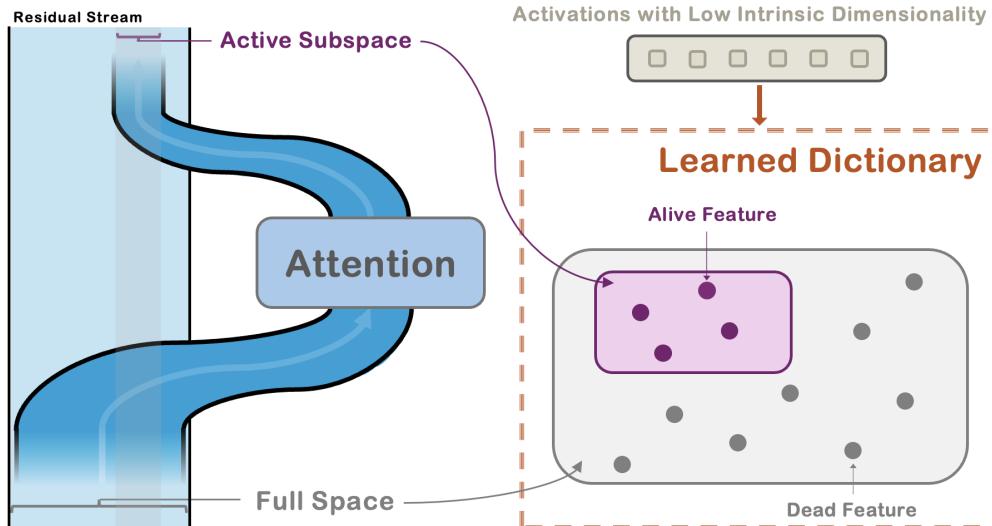
²OpenMOSS Team, School of Computer Science, Fudan University

jxwang25@m.fudan.edu.cn

{zfbe19, xpqiu}@fudan.edu.cn

Attention outputs exhibit pronounced low-rank structure compared to residual streams and MLP outputs, indicating that the attention layer writes into a subspace of the residual stream.

Low intrinsic dimensionality of activations is a root cause of dead features in sparse dictionary learning methods. Setting feature directions in the active subspace mitigates this issue.



ABSTRACT

Transformer architectures, and their attention mechanisms in particular, form the foundation of modern large language models. 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 consistently observed across diverse model families and datasets, and is induced by the attention output projection matrix. Critically, we find this low-rank structure as a key factor 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.

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 (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 (Templeton et al., 2024; Gao et al., 2024; Hazra et al., 2025). 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 dimensionality analyses (Guth et al., 2023; Staats et al., 2025), we demonstrate that this phenomenon holds universally across layers, datasets, and model families—Llama 3.1 (Dubey et al., 2024), Gemma 2 (Rivière et al., 2024), and Qwen 3 (Yang et al., 2025), 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 dimensionality strongly correlates with the number of dead features, suggesting a mismatch between random initialization and the low-dimensional geometry of the activations. Drawing inspiration from Phan et al. (2025)’s principal component initialization for the first network layer, 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 SparseAdam¹, 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 & Goldberg, 2020; Hu et al., 2022), and the inherent low-rank bottleneck in single-head outputs (Bhojanapalli et al., 2020).

Different from these prior lines of work, we demonstrate that the multi-head self-attention outputs exhibit a low-rank structure, revealing a distinct and under-explored phenomenon.

¹<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 suggests that high-level concepts align with linear directions in representation space (Arora et al., 2018; Olah et al., 2020; Elhage et al., 2022; Park et al., 2024). Motivated by this view, various sparse dictionary learning methods have been developed for interpretability, 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 methods decompose activations into sparse feature combinations, while differing in their strategies for predicting or approximating feature activations. The hypothesis has been validated across diverse 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*² (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.

2.4 PCA-INSPIRED NETWORK INITIALIZATION

A common practice applies PCA to input data for dimensionality reduction before network training (Hastie et al., 2009; Montavon et al., 2012; Jolliffe, 1986; Bishop & Nasrabadi, 2007). Recently, Phan et al. (2025) proposed *PCsInit*, which initializes the first layer weights of networks with top principal components of data—embedding the PCA transform directly into the network. This provides the model with a superior parameter set (Gu et al., 2025), boosting performance by construction.

3 PRELIMINARIES

3.1 MULTI-HEAD SELF-ATTENTION AND NOTATIONS

We consider a Transformer block with multi-head self-attention (MHSA) (Vaswani et al., 2017). Given input activations $X \in \mathbb{R}^{n \times d}$, where n is the token count and d is the model 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:

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

Let $Z = \text{Concat}[Z_1, \dots, Z_H] \in \mathbb{R}^{n \times d}$ denote the concatenated output of all attention heads (Nanda & 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 is 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) introduced by Gao et al. (2024). Unlike standard SAEs that impose an ℓ_1 penalty, TopK SAE enforces exact sparsity by keeping

²Following Bricken et al. (2023b), we define a feature as dead if it never activates over 10 million tokens in this paper.

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

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

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

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

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}_{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

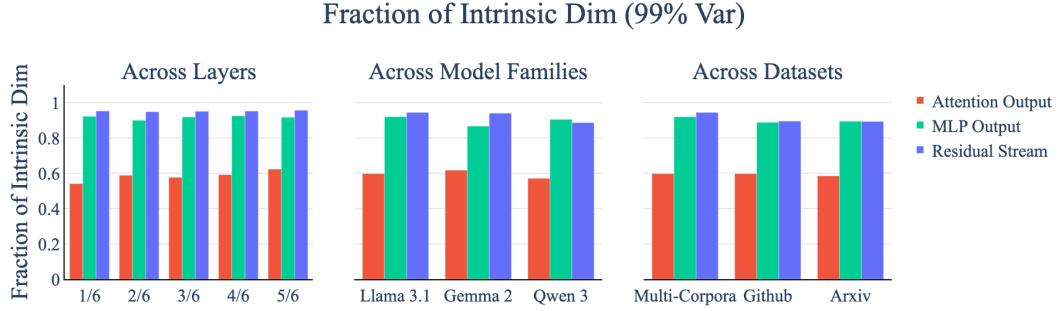


Figure 1: Across layers, model families and datasets, **attention outputs** exhibit dramatically lower intrinsic dimensions (details in Section 4.1) than **residual streams** and **MLP outputs**, showing that the attention layer writing into a low dimensional subspace of residual stream is a universal 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 $\tilde{A} \in \mathbb{R}^{n \times d}$, 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, \tilde{A} refers to the mean-centered activations. We refer readers to Appendix D for more details of activations sources.

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

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

where $U \in \mathbb{R}^{n \times r}$, $V \in \mathbb{R}^{d \times r}$, and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$ contains the singular values $\sigma_1 \geq \dots \geq \sigma_r \geq 0$, with $r = \text{rank}(\tilde{A})$. 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 subspace³ (Figure 1). We further compute the fraction of delta downstream loss recovered by different number of components (Section 4.2). These metrics complement our central findings, offering a numerical characterization of low-rankness.

For each type of activations, we collect 10 million activation vectors⁴. We empirically verify that this sample size is sufficient to produce stable and reproducible singular spectrum analysis.

4.2 EMPIRICAL EVIDENCE OF LOW-RANK STRUCTURE

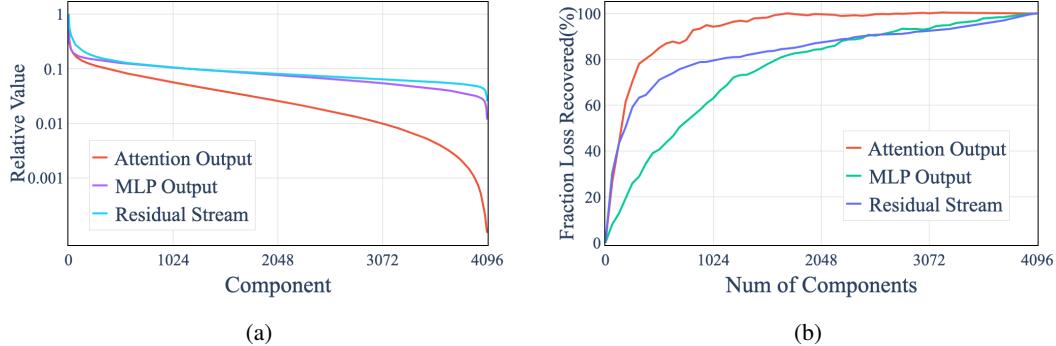


Figure 2: Results for 10M tokens from SlimPajama (Soboleva et al., 2023) fed into Llama-3.1-8B. **(a)** The attention outputs are the most low-rank, as indicated by the sharpest decay in singular values. **(b)** Fraction of loss recovered using varying numbers of top singular components.

We draw our findings from three lines of evidence:

Low Intrinsic Dimension of Attention Outputs Attention outputs has 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 Singular Spectral Decay in Attention Outputs This is quantitatively evidenced by the number of components retaining significant energy: only 74.7% singular values exceed 1% of the maximum in attention outputs, versus 100.0% for MLP outputs and residual streams (Figure 2a).

Efficient Downstream Loss Recovery Compared to zero ablation, attention outputs requires only 39.1% of dimensions to recover over 99% of the downstream loss, versus 95.3% and 96.9% for MLP outputs and residual streams (Figure 2b).

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

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 (singular value spectra of O) is shaped, we analyze the variance of O along an arbitrary

³We use 0.99 as the threshold in main text, results of some other thresholds are provided in Appendix F, with no influence to the conclusion.

⁴In rare cases, outlier activations inflate variance along some directions, making variance-based dimensionality estimates unreliable; to address this, we filter activations with a norm exceeding 5σ from the mean.

unit direction $\hat{e} \in \mathbb{R}^d$, given by:

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

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:

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

We refer to $\text{Var}(Zv)$ 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 MHSAs output satisfies

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

5 ACTIVE SUBSPACE INITIALIZATION FOR SPARSE AUTOENCODERS

5.1 EMPIRICAL CORRELATION BETWEEN LOW-RANK STRUCTURE AND DEAD FEATURES

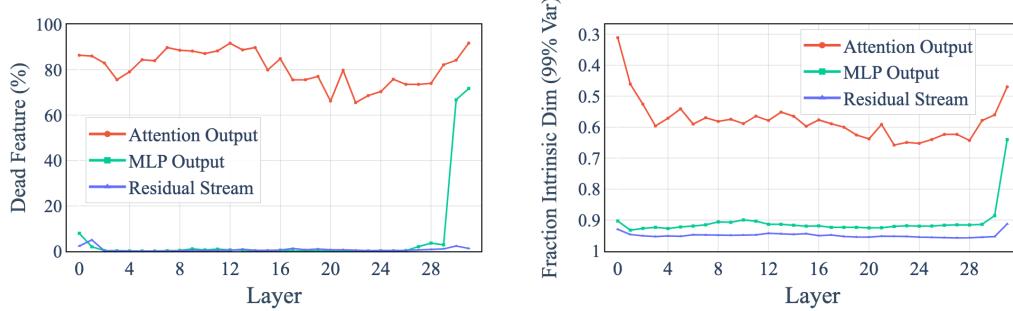


Figure 4: The number of dead features (left) and the intrinsic dimension (right) of each activation in Llama-3.1-8B, shows a surprising consistency: activations with lower intrinsic dimensions have more dead features, corresponding to all layers of attention output and last two layers of MLP output.

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), which provide a complete set of SAEs trained on attention output, MLP output, and residual stream⁵. We find that

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

the number of dead features is strongly related to intrinsic dimensions, as shown in Figure 4. This observation suggests that dead features may stem from the low-rank geometry of the activation space.

5.2 ACTIVE SUBSPACE INITIALIZATION FOR SPARSE AUTOENCODERS

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

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

where $U \in \mathbb{R}^{n \times d}$ is the left singular vector matrix, $\Sigma \in \mathbb{R}^{d \times d}$ is the diagonal matrix of singular values, and $V \in \mathbb{R}^{d \times d}$ contains the right singular vectors.

We select the top d_{init} right singular vectors to form the active subspace and then initialize the SAE weights directly in this subspace:

$$V_{\text{active}} = V_{:,d_{\text{init}}} \in \mathbb{R}^{d \times d_{\text{init}}}, \quad W_D \in \text{span}(V_{\text{active}}), \quad W_E = W_D^\top,$$

where W_E is the encoder weight matrix and W_D is the decoder weight matrix. Intuitively, ASI aligns the initial SAE parameters with the active directions of the data, ensuring that SAEs start in a meaningful low-dimensional subspace. As d_{init} decreases from the full space dimension⁶ within a certain range, the number of dead features in the SAE rapidly drops, with a corresponding improvement in Delta LM loss⁷ (Figure 5). Further ablation studies and implementation details are contained in Appendix B and Appendix I, respectively.

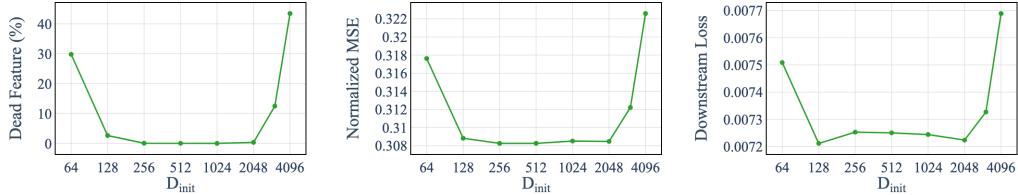


Figure 5: After using ASI, proportion of dead features (left), normalized MSE (mid) and Delta LM loss (right) across different subspace dimensions for activations with a full space dimension of 4096.

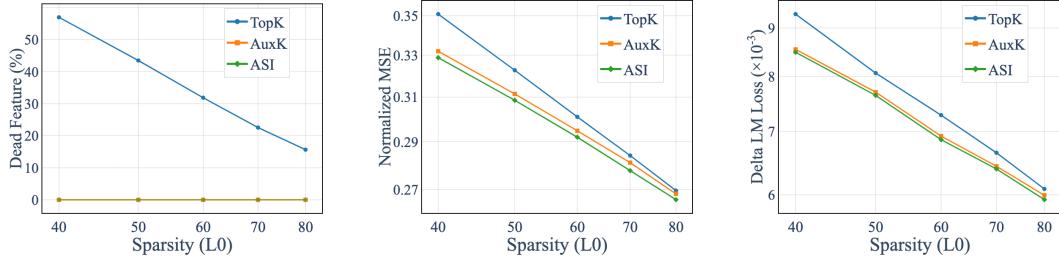


Figure 6: At a fixed number of features ($n = 32768$), **Active Subspace Init** achieves a better reconstruction-sparsity trade-off than **TopK** and **AuxK**. A similar trend is observed in its impact on Delta LM Loss. **Note:** The improvement in downstream loss is less pronounced than that in reconstruction,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

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

⁶Setting D_{init} equal to the full space dimension is equivalent to not using Active Subspace Initialization.

⁷This metric is defined as the difference between the original language model loss and the loss when the SAE is inserted at the corresponding position, evaluated over 1 million tokens.

Reduced dead features and Enhanced Sparsity-Reconstruction Frontier Without Additional Compute It achieves near-zero dead features and slightly superior results compared to the auxiliary loss approach (AuxK), at no additional computational cost of the same order. (Figure 6).

Optimal Scaling Characteristics Our approach demonstrates optimal scaling behavior across various SAE training methods. It outperforms TopK and AuxK in any evaluated scale, from 16K to 1M features (Section 5.3).

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

5.3 SCALING LAWS

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 G.5).

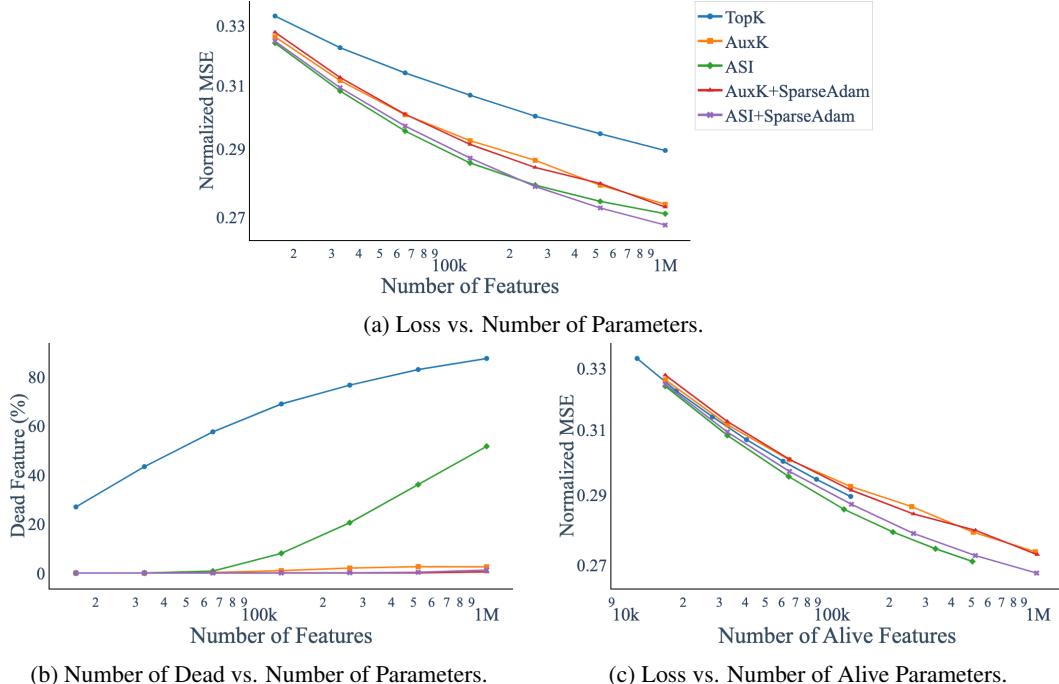


Figure 7: Scaling results of TopK SAEs and their variants enhanced with *AuxK*, *Active Subspace Init*, and *SparseAdam*—all trained on attention output from Llama-3.1-8B. (A) Loss at convergence across different feature counts: *Active Subspace Init* consistently achieves lower reconstruction error 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.

Active Subspace Init improves reconstruction. As shown in Figure 7a, *Active Subspace Init* consistently outperforms *TopK* and *AuxK* across all scales.

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 C.

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 evaluate this, we apply **Active Subspace Initialization** to Lorsa. This modification reduces the proportion of dead parameters significantly under identical hyperparameter settings (Figure 8), while also improving the reconstruction. Specifically, we initialize the decoder (corresponding to W^O in Lorsa) within the active subspace of each original MHSA head, while the encoder (corresponding to W^V in Lorsa) is aligned with the corresponding input-side directions of the decoder. Further implementation details are provided in Appendix I.

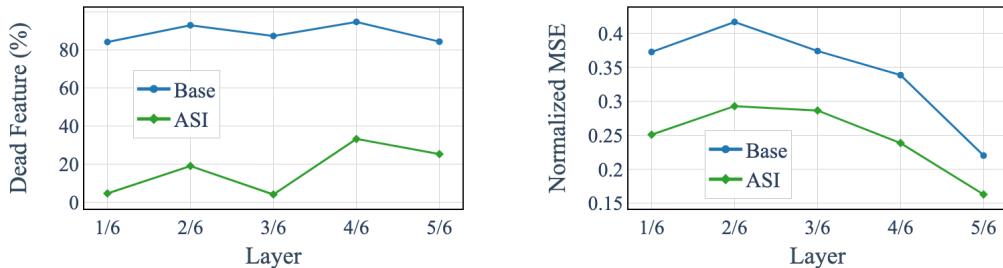


Figure 8: Effect of Active Subspace Initialization on reducing dead parameters in attention replacement model.

6 DISCUSSION

When to Use Active Subspace Initialization For activation sites that do not exhibit clear low-rank structures (e.g., residual stream), our method yields only limited gains (Appendix B.2). The singular value spectrum provides a more reliable indicator: when the spectrum decays rapidly in its tail, applying ASI is likely to be effective.

Causality between Low-Rank Structure and Dead Features We observed a strong correlation between low-rank structure and the emergence of dead features (Section 5), but the causal mechanism remains unclear. It may be tied to optimization dynamics or feature competition, yet a precise explanation is lacking. A deeper understanding of this relationship could also explain why ASI is less effective under high- ℓ_0 warm-up regimes (Appendix H). We leave this important line of inquiry to future work.

7 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.
- 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*, pp. 864–873. PMLR, 2020. URL <http://proceedings.mlr.press/v119/bhojanapalli20a.html>.
- Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern Recognition and Machine Learning. J. Electronic Imaging*, 16(4):049901, 2007. doi: 10.1117/1.2819119. URL <https://doi.org/10.1117/1.2819119>.
- 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 (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 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, Bin 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 Nikoaidis, 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 Vasudevan 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>.

Naibin Gu, Yilong Chen, Zhenyu Zhang, Peng Fu, Zheng Lin, Shuo Wang, Yu Sun, Hua Wu, Weiping Wang, and Haifeng Wang. Advantageous parameter expansion training makes better large language models, 2025. URL <https://arxiv.org/abs/2505.24241>.

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>.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009. ISBN 9780387848570. doi: 10.1007/978-0-387-84858-7. URL <https://doi.org/10.1007/978-0-387-84858-7>.

Dron Hazra, Max Loeffler, Murat Cubuktepe, Levon Avagyan, Liv Gorton, Mark Bissell, Owen Lewis, Thomas McGrath, and Daniel Balsam. Under the hood of a reasoning model. <https://www.goodfire.ai/blog/under-the-hood-of-a-reasoning-model>, 2025. Accessed: 2025-09-15, Blog post from Goodfire Research.

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>.

Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 1986. ISBN 978-1-4757-1906-2. doi: 10.1007/978-1-4757-1904-8. URL <https://doi.org/10.1007/978-1-4757-1904-8>.

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>.

Grégoire Montavon, Genevieve B. Orr, and Klaus-Robert Müller (eds.). *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700 of *Lecture Notes in Computer Science*. Springer, 2012. ISBN 978-3-642-35288-1. doi: 10.1007/978-3-642-35289-8. URL <https://doi.org/10.1007/978-3-642-35289-8>.

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 (eds.), *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, ACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pp. 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>.

Nhan Phan, Thu Nguyen, Pål Halvorsen, and Michael A. Riegler. Principal components for neural network initialization. *CoRR*, abs/2501.19114, 2025. doi: 10.48550/ARXIV.2501.19114. URL <https://doi.org/10.48550/arXiv.2501.19114>.

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 (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 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>.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *CoRR*, abs/2407.14435, 2024. doi: 10.48550/ARXIV.2407.14435. URL <https://doi.org/10.48550/arXiv.2407.14435>.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charlène Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024. doi: 10.48550/ARXIV.2408.00118. URL <https://doi.org/10.48550/arXiv.2408.00118>.

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>.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

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>.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yingqi Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL <https://doi.org/10.48550/arXiv.2505.09388>.

A DECLARATION OF LLM USAGE

We acknowledge the use of large language models (LLMs) to assist in the preparation of this manuscript. Specifically, **ChatGPT** and **Claude** were utilized during the writing and coding process for the exclusive purpose of improving text clarity, grammar, overall fluency, and for generating boilerplate code structures. All ideas, theoretical developments, experimental designs, results, analyses, and scientific conclusions are entirely our own. The LLMs acted solely as assistive tools and were not involved in any aspect of the intellectual or scientific work.

B ABLATION STUDY

B.1 ACTIVE SUBSPACE INIT VS RANDOM SUBSPACE INIT

We employ random subspace initialization as a baseline and observe that it consistently degrades SAE training across all metrics, as shown in Figure 9.

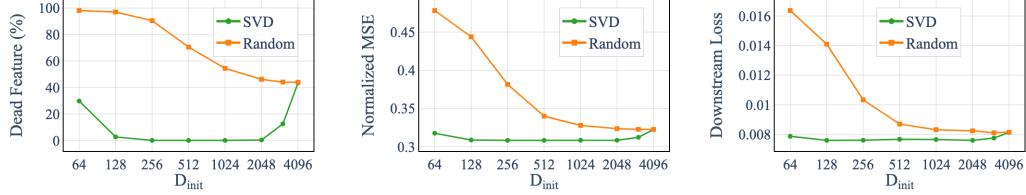


Figure 9: For activations with a full space dimension of 4096, proportion of dead features (left), normalized MSE (mid) and Delta LM loss (right) across different subspace dimensions. Random subspaces are used as the baseline, whereas only initialization with the active subspace yields improvement.

B.2 APPLY ON NEAR-FULL-RANK ACTIVATION

We also apply Active Subspace Initialization (ASI) to near-full-rank activations, such as those in the residual stream, to evaluate its generality. When training an SAE on the post-layer-15 residual stream of Llama-3.1-8B, we find ASI yields minimal gains (Figure 10). This is consistent with our expectation, as these activations inherently exhibit a lower rate of dead features even with standard initialization.



Figure 10: For activations with a full space dimension of 4096, proportion of dead features (left), normalized MSE (mid) and Delta LM loss (right) across different subspace dimensions. Random subspaces are used as the baseline, whereas only initialization with the active subspace yields improvement.

B.3 CHOICE OF THE INITIAL DICTIONARY SIZE D_{init}

As shown in Figure 5, D_{init} is a hyperparameter with a wide range of acceptable values (from 256 to 2048). We hypothesize that the performance degradation at very low values occurs because an excessively small dictionary fails to span the subspace containing critical information needed for effective reconstruction.

C 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.

D 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-3.1-8B⁸, Qwen3-8B⁹, and Gemma-2-9B¹⁰—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 model intermediate layers. For example, in LLaMA-3.1-8B (32 layers), we use activations from layer 15 (zero-indexed). We analyze three types of activations: (1) attention output, (2) MLP output, and (3) residual stream (post layer).

E MORE LOW-RANK RESULT ACROSS DIFFERENT MODELS AND DATASETS

We present relative singular values and fraction of loss recovered for some other model-dataset pairs in Figure 11. Models include pythia-2.8b¹¹. Datasets include RedPajamaGithub¹² and CCI3-Data¹³

F 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 12. Attention outputs show low-rank structure consistently.

G SAE TRAINING DETAILS

G.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.

⁸<https://huggingface.co/meta-llama/Llama-3.1-8B>

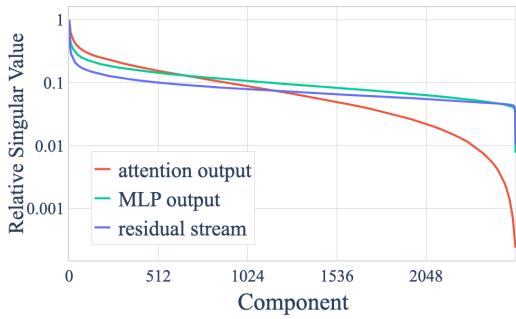
⁹<https://huggingface.co/Qwen/Qwen3-8B>

¹⁰<https://huggingface.co/google/gemma-2-9b>

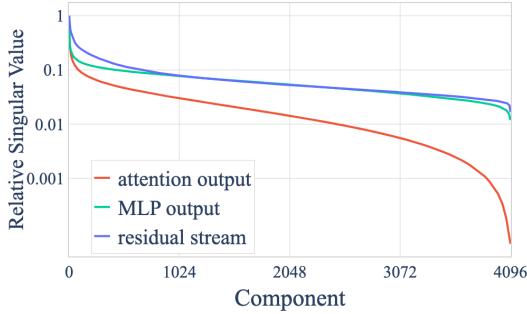
¹¹<https://huggingface.co/EleutherAI/pythia-2.8b>

¹²<https://huggingface.co/datasets/cerebras/SlimPajama-627B>

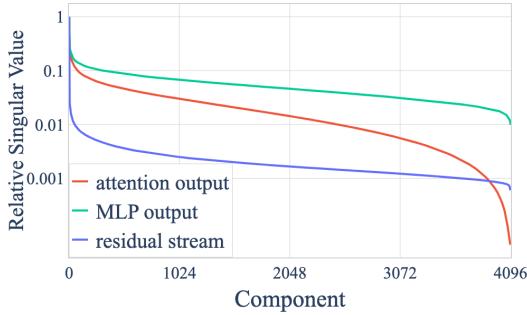
¹³<https://huggingface.co/datasets/BAAI/CCI3-Data>



(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 11

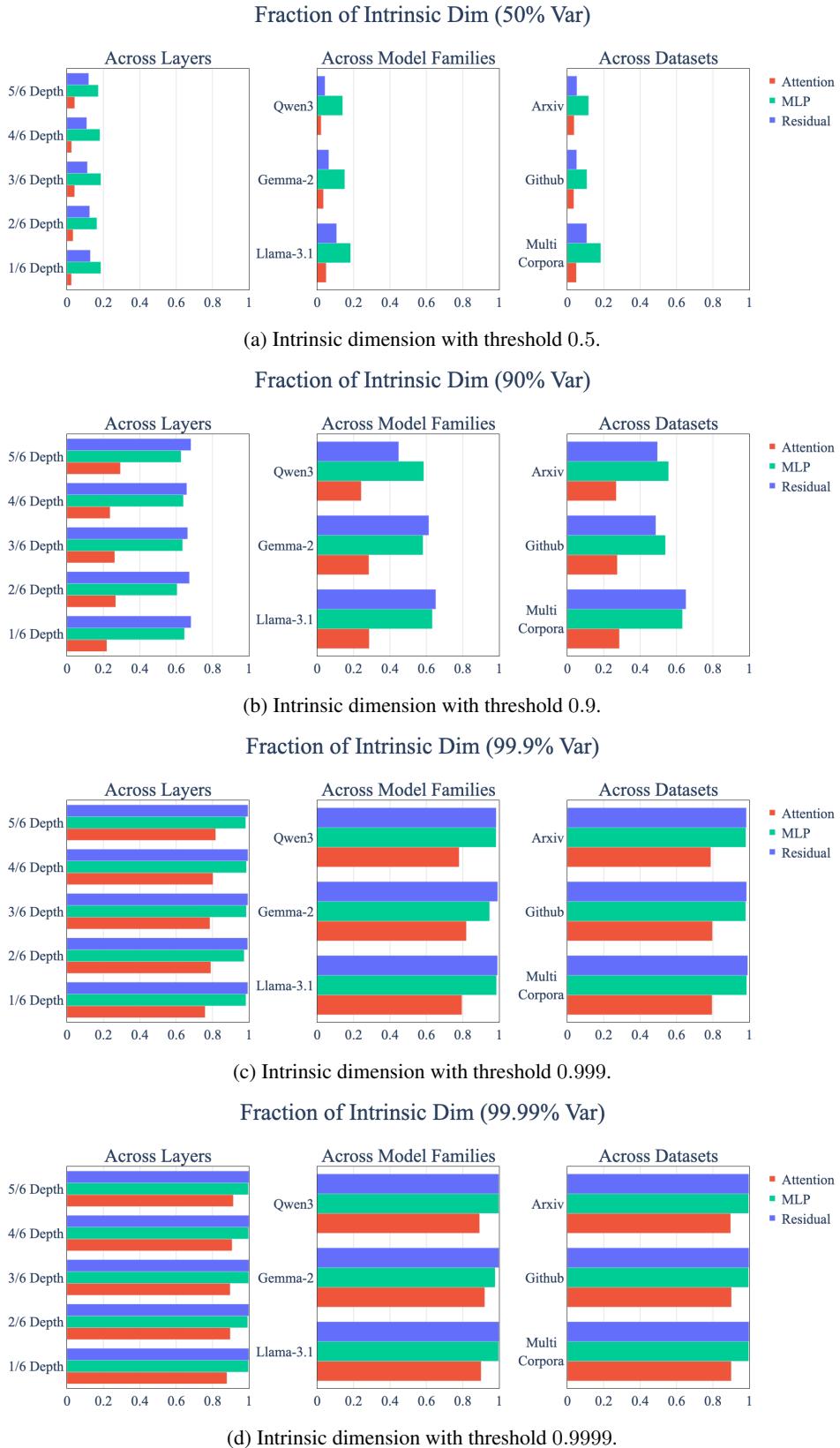


Figure 12: Comparison of intrinsic dimensions across different variance thresholds.

G.2 INITIALIZATION

The decoder columns $W_{:,i}^{dec}$ are initialized uniformly, and the optimal norm for them is found through a grid search to minimize the initial reconstruction loss. We find that the specific initialization norm has little impact, as long as in a reasonable scope. For example, initializing $W_{:,i}^{dec}$ uniformly with a fixed bound, as in Conerly et al. (2025), yields similar results. The encoder weights $W_{:,i}^{enc}$ are initialized as the transpose of $W_{:,i}^{dec}$, while both the encoder bias b^{enc} and decoder bias b^{dec} are set to zero.

G.3 OPTIMIZATION

We train SAEs using the Adam optimizers, both with $lr = 2 * 10^{-4}$, $batchsize = 32768$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The differences in scaling law experiments are described in the Appendix G.5.

G.4 JUMPRELU SAEs

We trained JumpReLU SAEs (Rajamanoharan et al., 2024) under two distinct hyperparameter configurations: one maintaining a consistently low ℓ_0 value throughout training, and another where ℓ_0 is gradually decreased from a higher initial value. Unless otherwise specified, all JumpReLU SAEs were trained using the same settings as Conerly et al. (2025), which corresponds to the latter configuration. The key modifications for the former setting are as follows: (1) we initialized the encoder bias to zero instead of applying the heuristic that equalizes feature activation counts at initialization, and (2) we kept the sparsity coefficient fixed rather than employing a global warm-up schedule. As a result, the ℓ_0 sparsity level started at a relatively low value early in training. This design is critical to our approach: we observed that if the model remains in a high- ℓ_0 regime (e.g., on the order of $d_{model}/2$) for an extended period before sparsity increases, the feature directions tend to drift away from the active subspace during this phase, thereby diminishing the effectiveness of our method (Appendix H.1).

G.5 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 the batch size to 4096—below the critical batch size—which differs from the setting of 32768 used in other experiments in this paper. This reduction is necessary because training SAEs with 1 million features requires substantial memory, posing significant computational challenges.

Learning Rate The learning rate for **Adam** and **SparseAdam** is swept 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**. We employ a three-phase learning rate schedule consisting of a linear warm-up, a stable phase, and a linear decay. The learning rate increases linearly from zero to its maximum value over the first 500 steps, remains constant during the intermediate phase, and then decays linearly to 1% of the maximum value over the final 20% of the total training steps.

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 We use 768 for all experiments.

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

H USE ASI ON OTHER ACTIVATION FUNCTIONS

H.1 JUMPRELU

Another wildly used activation function is JumpReLU (Rajamanoharan et al., 2024). We trained the JumpReLU SAEs under two different hyperparameter settings: one with a consistently low ℓ_0 value and another where ℓ_0 gradually decreases from a higher initial value, as described in Appendix G.4. We observed that our method is effective in the former case (Figure 13) but shows little improvement in the latter (Figure 14).

For cases where one follows a schedule that gradually reduces ℓ_0 from a high initial value, we recommend first applying PCA to reduce the dimensionality of the data. The SAE can then be trained on the reduced representation until the ℓ_0 level reaches the target range. Afterwards, the PCA projection matrix can be folded into the model parameters, and training can continue in the original space. This achieves a similar effect without the drawbacks of prolonged training in the high- ℓ_0 regime.

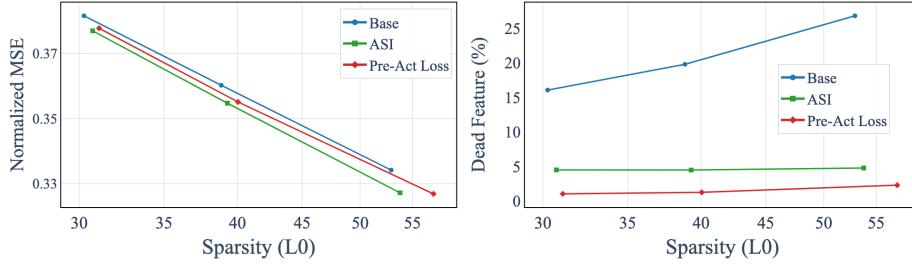


Figure 13

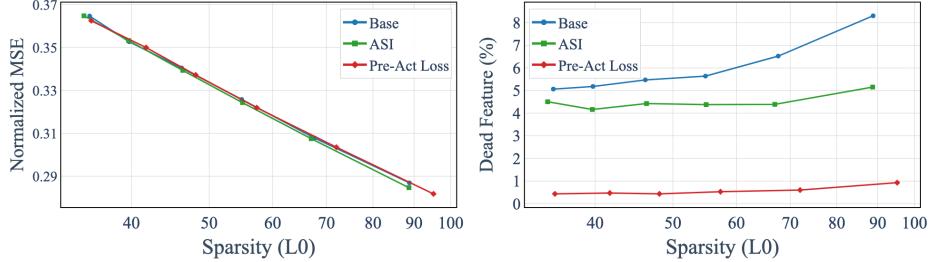


Figure 14

H.2 TOPK WITH K ANNEAL

To enhance the finding in Section H.1, we conduct experiments on a variant of TopK, which sets K to a high value and then lets it decrease during training (He et al., 2024). We find ASI also fails in this case (Figure 15).

I PSEUDO-CODE FOR IMPLEMENTING ACTIVE SUBSPACE INIT

Below is a PyTorch-style pseudo-code for Active Subspace Initialization.

Use on SAE

```
# X: activation batch [batch_size, d_model]
# W_E: decoder weight [d_model, d_sae]
# W_D: decoder weight [d_sae, d_model], initialized uniformly
```

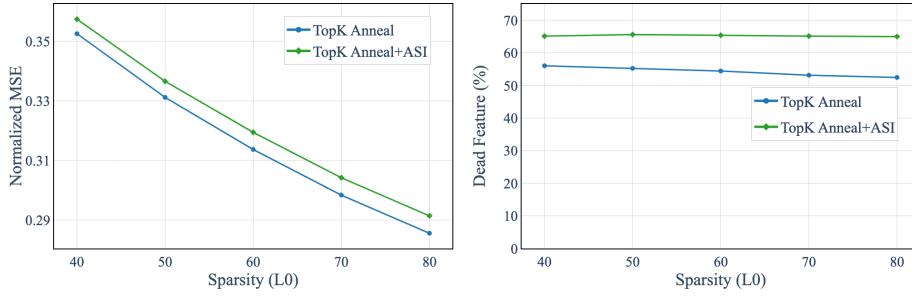


Figure 15

```
# d_active_subspace: target subspace dimension

# 1. Demean the activations
demeaned_X = X - X.mean(dim=0) # [batch_size, d_model]

# 2. Compute SVD
U, S, V = torch.svd(demeaned_label) # V: [d_model, d_model]

# 3. Take top-d_init singular vectors
proj_weight = V[:, :d_init] # [d_model, d_init]

# 4. Fold projection into decoder weights
W_D.copy_(W_D[:, :d_init] @ proj_weight.T)

# 5. Init W_E with W_D.T
W_E.copy_(W_D.T)
```

Use on Lorsa

```
# X: activation batch [batch_size, seq_len, d_model]
# mhsa: pretrained MHSA module with W_O and W_V
# W_O, W_V: Lorsa decoder/encoder weights
# d_head: dimension of original MHSA heads
# n_heads: number of original MHSA heads
# n_lorsa_heads: number of Lorsa heads

# 1. Run MHSA to capture per-head outputs
Z = mhsa.compute_z(X) # [batch_size, seq_len, n_heads, d_head]
output_per_head = torch.einsum('bsnh, nhd->bsnd', Z, mhsa.W_O)

# 2. For each original head
# initialized some larsa heads into it's active subspace
rate = n_lorsa_heads // n_heads
for mhsa_index in range(n_heads):
    head_slice = [rate*mhsa_index:rate*(mhsa_index+1)]
    # [B, S, d_model]
    output=output_per_head[:, :, mhsa_index, :]
    output_flat=output.flatten(0, 1) # [B*S,d_model]

    # 3.1 Demean
    demeaned_output=output_flat - output_flat.mean(dim=0)

    # 3.2 Compute SVD
    U, S, V = torch.svd(demeaned_output)
```

```

# 3.3 Take top-d_head singular vectors
proj_weight = V[:, :d_head]

# 3.4 Update part of decoder weights W_O
W_O[head_slice] = W_O[head_slice, :d_head] @ proj_weight.T

# 3.5 Update part of encoder weights W_V
head_trans = mhsa.W_V[mhsa_index] @ mhsa.W_O[mhsa_index]
W_V[head_slice] = W_O[head_slice] @ head_trans.T

```

The strategy of initialize W_V in Lorsa is a method like the **tied initialization** used in SAEs to ensure alignment between feature encoding and decoding¹⁴. This approach has been shown to be crucial for reducing dead features in SAEs (Gao et al., 2024). We think the same thought could also be used to improve the replacement model for MLP (trancoder and cross layer transcoder), which we leave a deeper investigation to future work.

¹⁴"Match" means encoder can be initialized to predict relatively accurate feature activation values for decoder.