Data Whitening Improves Sparse Autoencoder Learning

Anonymous submission

Abstract

Sparse autoencoders (SAEs) have emerged as a promising approach for learning interpretable features from neural network activations. However, the optimization landscape for SAE training can be challenging due to correlations in the input data. We demonstrate that applying PCA Whitening to input activations—a standard preprocessing technique in classical sparse coding—improves SAE performance across multiple metrics. Through theoretical analysis and simulation, we show that whitening transforms the optimization landscape, making it more convex and easier to navigate. We evaluate both ReLU and Top-K SAEs across diverse model architectures, widths, and sparsity regimes. Empirical evaluation on SAEBench, a comprehensive benchmark for sparse autoencoders, reveals that whitening consistently improves interpretability metrics, including sparse probing accuracy and feature disentanglement, despite minor drops in reconstruction quality. Our results challenge the assumption that interpretability aligns with an optimal sparsity–fidelity trade-off and suggest that whitening should be considered a standard preprocessing step for SAE training.

1 Introduction

Sparse autoencoders (SAEs) (Ng et al. 2011) have become a cornerstone of mechanistic interpretability, enabling researchers to extract human-understandable features from the internal activations of large language models (LLMs) (Bricken et al. 2023; Cunningham et al. 2023). Individual neurons are often polysemantic, encoding multiple unrelated concepts simultaneously (Elhage et al. 2022), making it difficult to isolate meaningful representations. By learning sparse, overcomplete dictionaries, SAEs decompose neural activations into latent dimensions that align with meaningful concepts or functions within a model's computation (Marks et al. 2024; Kharlapenko et al. 2025; Klindt et al. 2025).

Despite their promise, training SAEs remains challenging. The optimization landscape is complex (Evci et al. 2020), and finding features that are both interpretable and faithful to the original representations requires careful tuning of sparsity penalties and architecture choices (Gao et al. 2024; Templeton et al. 2024; Bussmann et al. 2025; O'Neill, Gumran, and Klindt 2024)—but these methods still operate on correlated data, leaving the structure of the activation space unchanged.

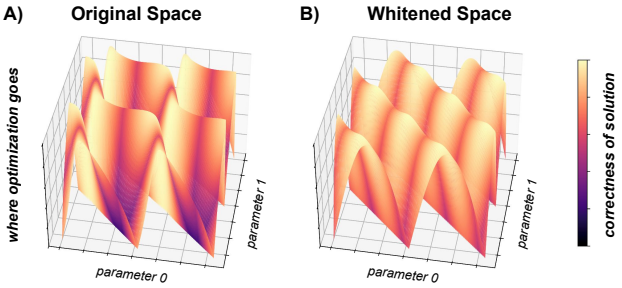

Figure 1: **Whitening transforms the optimization landscape.** 3D visualization of the sparse coding landscape over all dictionary angles $(\theta_0, \theta_1) \in [0, 2\pi]^2$. Surface height shows sparsity (higher = sparser); color indicates feature recovery quality (brighter = better). **A**: Without whitening, high sparsity regions (peaks) are misaligned with accurate feature recovery (bright). **B**: After whitening, the landscape becomes isotropic and sparsity aligns with feature quality.

We revisit a classical idea from sparse coding and neuroscience: *data whitening* via principal component analysis (PCA). While whitening is standard practice in classical sparse coding algorithms (Olshausen and Field 1996; Lee et al. 2006) and independent component analysis (ICA) (Hyvärinen, Hurri, and Hoyer 2001), it has been largely overlooked in modern SAE training. Whitening removes correlations and equalizes variance in the input space, simplifying optimization and encouraging more independent features. In the brain's visual system, whitening is believed to occur in the retina, where early sensory circuits decorrelate visual inputs to improve feature separability (Atick and Redlich 1990, 1992; Olshausen and Field 1997; Graham, Chandler, and Field 2006). Motivated by this, we apply PCA Whitening as a preprocessing step for SAE training. This simple modification yields substantial gains in interpretability without altering model architecture or loss design. Our key contributions are:

- We provide theoretical analysis showing how whitening improves the SAE optimization landscape, making it more convex and isotropic.
- We present simulation studies demonstrating the practical benefits of whitening for optimization.

- We conduct comprehensive experiments on SAE Bench showing that whitening improves interpretability metrics including sparse probing accuracy (+7.3%), spurious correlation removal (+54.03%), and targeted probe perturbation (+372%).

2 Related Work

Sparse Autoencoders. Recent work has explored improving the interpretability and faithfulness of SAEs. Top-K SAEs (Gao et al. 2024) fix the number of active latents per input, removing the need for an L_1 penalty. BatchTop-K (Bussmann, Leask, and Nanda 2024) extends this to the batch level to enhance reconstruction, while Gated and JumpReLU SAEs (Rajamanoharan et al. 2024a,b) address shrinkage and fidelity issues. Matryoshka SAEs (Bussmann et al. 2025) enable hierarchical feature discovery, and O'Neill, Gumran, and Klindt (2024) find that SAEs do not fully solve sparse coding, though deeper encoders may help. Our work complements these methods by showing that PCA whitening improves interpretability without altering model architecture or loss.

Sparse Coding. Whitening has been standard in sparse coding since Olshausen and Field's seminal work (Olshausen and Field 1996). Efficient coding theory (Barlow et al. 1961) provides theoretical justification for whitening as optimal preprocessing (Atick and Redlich 1990, 1992). In ICA, whitening is standard preprocessing because it simplifies the problem of demixing latent variables, which are assumed to be white (Hyvärinen, Hurri, and Hoyer 2001).

Interpretability Evaluation. Recent work has established standardized frameworks for quantifying interpretability (Leavitt and Morcos 2020). Karvonen et al. (2025) introduced *SAEBench*, a large-scale benchmark that systematically compares SAE architectures across diverse interpretability metrics. Karvonen et al. (2024) proposed domain-grounded evaluation using structured tasks such as Chess and Othello, while Kantamneni et al. (2025) evaluated SAEs on real-world downstream probing tasks, highlighting limitations in current interpretability methods. Visual and non-linguistic interpretability metrics are proposed and discussed in (Zimmermann et al. 2021; Klindt et al. 2023; Zimmermann, Klindt, and Brendel 2024; Klindt et al. 2025; Paulo and Belrose 2025).

3 Background

3.1 Sparse Autoencoders

A sparse autoencoder learns to decompose neural network activations $\mathbf{x} \in \mathbb{R}^d$ into a sparse, overcomplete representation. The model consists of an encoder that maps inputs to features $\mathbf{f} = \sigma(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e) \in \mathbb{R}^m$ where $m > d$, and a decoder that reconstructs the input as $\hat{\mathbf{x}} = \mathbf{W}_d\mathbf{f} + \mathbf{b}_d$. The training objective minimizes reconstruction error while promoting sparsity:

$$\mathcal{L}(x) = \|x - \hat{x}\|_2^2 + \lambda S(f), \tag{1}$$

where $S(f)$ penalizes dense activations and λ controls the sparsity–reconstruction trade-off.

ReLU SAEs. ReLU-based SAEs impose *soft sparsity* by adding an L_1 penalty to the latent activations:

$$\mathcal{L}_{\text{ReLU}} = \|x - \hat{x}\|_2^2 + \lambda\|\mathbf{f}\|_1 \tag{2}$$

This encourages gradual sparsity, allowing some activations to vary smoothly while penalizing dense features.

Top-K SAEs. Top-K SAEs instead enforce *hard sparsity* by keeping only the k largest activations per input:

$$\mathbf{f} = \text{TopK}(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e, k) \tag{3}$$

3.2 Classical Sparse Coding and Whitening

In classical sparse coding (Olshausen and Field 1996), data whitening is a standard preprocessing step. Given data $\mathbf{x}$ with covariance $\mathbf{C} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top]$, whitening transforms the data to $\mathbf{z} = \mathbf{C}^{-1/2}\mathbf{x}$, ensuring the covariance of $\mathbf{z}$ is the identity matrix.

This preprocessing provides several important benefits for model training. First, whitening makes the optimization landscape more isotropic, preventing bias toward high-variance directions. Second, whitening removes correlations in the input data that interfere with feature learning, allowing the model to discover more independent and disentangled representations. Third, whitening improves conditioning of the reconstruction problem by equalizing variance across dimensions, which stabilizes gradient updates and accelerates optimization. Despite these benefits, whitening has been largely underutilized in modern SAE training.

3.3 SAEBench Metrics

SAEBench is a comprehensive evaluation suite designed to assess SAEs across multiple interpretability dimensions. We evaluate SAEs across five key SAEBench metrics:

CE Loss Score: Measures how well SAE-reconstructed activations preserve the model's next-token prediction accuracy. Higher scores indicate more faithful reconstruction.

Explained Variance: Fraction of variance in the original activations preserved by SAE reconstruction. Higher values indicate better reconstruction fidelity.

Sparse Probing (Top 1): Accuracy of linear probes trained on SAE activations across 35 binary diagnostic tasks. Higher values indicate more localized and interpretable features.

SCR (Top 20): Evaluates an SAE's ability to remove spurious correlations by ablating features associated with confounding variables(e.g., *gender* and *profession*). Higher values reflect stronger disentanglement and debiasing.

TPP (Top 20): Measures causal specificity by ablating latents linked to a target class and assessing selective performance drop across all classes. Higher scores indicate more disentangled and causally isolated features.

Metric Selection We selected these five metrics as they best capture reconstruction quality and latent interpretability, while remaining robust for models at the parameter scale ($<$ 2B) used in our experiments (see note: Karvonen 2025).

4 Theoretical Analysis

4.1 How Whitening Reshapes the Optimization Landscape

We provide a theoretical analysis demonstrating why whitening improves SAE training. Consider a simple 2D sparse coding problem where observations $\mathbf{y} \in \mathbb{R}^2$ are generated by mixing sparse sources $\mathbf{z} \in \mathbb{R}^2$ through a dictionary $\mathbf{A} \in \mathbb{R}^{2\times 2}$:

$$\mathbf{y} = \mathbf{A}\mathbf{z} \tag{4}$$

The goal of sparse coding is to recover the true dictionary $\mathbf{A}$ by learning an inverse mapping $\mathbf{W}$ such that the reconstructed sources $\hat{\mathbf{z}} = \mathbf{W}\mathbf{y}$ are sparse. The optimization landscape depends on two competing objectives: i) **Sparsity**: The learned features should be sparse, measured by the inverse of the mean L1 norm: $\mathcal{S}(\mathbf{W}) = \frac{1}{\mathbb{E}[|\mathbf{W}\mathbf{y}|]}$ ii) **Feature Recovery**: The learned dictionary should align with the true dictionary, measured by the *mean correlation coefficient* (Hyvarinen and Morioka 2017, see Appendix B, Eq. (13)).

An ideal optimization landscape should exhibit high values for both metrics at the same location in parameter space, with a smooth, convex basin leading to this optimum.

4.2 Simulation: Optimization Landscape Analysis

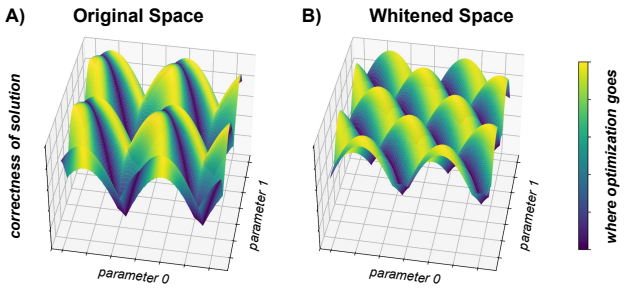

Figure 2: **Complementary view of optimization landscape.** Surface height shows feature recovery quality; color indicates sparsity level (brighter = sparser). **A**: Optimizing for sparsity (climbing to bright) may lead to poor feature recovery. **B**: After whitening, pursuing sparsity naturally yields interpretable features (bright colors at peaks).

To visualize how whitening affects these landscapes, we conducted a systematic simulation study. We generated synthetic 2D data with correlated features by mixing sparse sources through a random dictionary. We then computed both the sparsity metric $\mathcal{S}(\mathbf{W})$ and feature recovery metric $\mathcal{R}(\mathbf{W}, \mathbf{A})$ across a dense grid of possible dictionary angles $(\theta_0, \theta_1) \in [0, 2\pi]^2$, where:

$$\mathbf{W}(\theta_0, \theta_1) = \begin{bmatrix} \cos\theta_0 & \sin\theta_0 \\ \cos\theta_1 & \sin\theta_1 \end{bmatrix} \tag{5}$$

This exhaustive search over the parameter space reveals the complete structure of the optimization landscape.

Figure Interpretation. Figure 1 shows how whitening transforms the optimization landscape by comparing before (A) and after (B) whitening. The surface height represents sparsity level (higher = sparser) while colors indicate feature recovery quality (brighter = better recovery).

Without whitening (A), the landscape exhibits a narrow, elongated basin with steep gradients and poor conditioning. Critically, the highest sparsity regions (tall peaks) are misaligned with accurate feature recovery (which appears in duller colors), meaning that achieving high sparsity does not guarantee interpretable features. After whitening (B), the landscape becomes more isotropic with a wider basin. The bright colors now concentrate at the peaks, showing that sparsity and feature recovery are aligned—the sparsest solutions also yield the best feature reconstruction.

Figure 2 provides the complementary perspective, with surface height representing feature recovery quality and colors indicating sparsity (brighter = sparser). This view reveals that without whitening (A), pursuing bright regions (high sparsity) can lead to valleys rather than peaks (poor feature recovery). After whitening (B), the brightest colors align with the highest peaks, confirming that optimizing for sparsity naturally leads to better feature recovery.

4.3 Implications for SAE Training

These geometric transformations have direct consequences for SAE training. Whitening equalizes the eigenspectrum of the data covariance, transforming ill-conditioned problems into well-conditioned ones and leading to more stable gradient updates. In correlated data, sparsity and feature interpretability can be misaligned—pursuing sparsity may not yield semantically meaningful features. Whitening aligns these objectives, ensuring that sparse solutions correspond to interpretable features. While sparse coding is inherently non-convex, whitening makes the landscape more convex-like with a smoother basin around the global optimum, reducing sensitivity to initialization and hyperparameters. Finally, whitening enforces second-order independence by removing correlations, providing an inductive bias that encourages feature disentanglement. Detailed simulation methodology is provided in Appendix B.

5 Experiments

5.1 Experimental Setup

We evaluate the effect of PCA whitening on SAE training using SAEBench, a benchmark measuring multiple dimensions of SAE quality. Experiments were conducted on Pythia-160M (layer 8, d=768) and Gemma-2-2B (layer 12, d=2304). For each model, we trained both ReLU and Top-K SAEs with dictionary widths of 2^{14} and 2^{16}, applying three sparsity levels for ReLU (0.012, 0.02, 0.06) and two target L_0 values for Top-K (80, 160). Each configuration was trained under two conditions—standard training and training with PCA-whitened activations.

This yields 40 total configurations (24 ReLU, 16 Top-k), with each configuration trained both with and without whitening. Training followed the open-source ***dictionary_learning*** framework (Karvonen et al. 2025); full configuration details are provided in Appendix A.

Metric	ReLU	+Whitening	Δ	% Δ	p-value
CE Loss Score	**0.980** $\pm$ 0.005	0.954 $\pm$ 0.006	-0.026	-2.64%	$2.86 \cdot 10^{-5}$
Explained Variance	**0.813** $\pm$ 0.028	0.772 $\pm$ 0.027	-0.041	-5.02%	$2.84 \cdot 10^{-6}$
Sparse Probing (Top 1)	0.757 $\pm$ 0.008	**0.812** $\pm$ 0.008	+0.054	+7.15%	$1.05 \cdot 10^{-5}$
SCR (Top 20)	0.176 $\pm$ 0.015	**0.271** $\pm$ 0.019	+0.095	+54.03%	$3.25 \cdot 10^{-6}$
TPP (Top 20)	0.021 $\pm$ 0.004	**0.098** $\pm$ 0.010	+0.078	+372.00%	$5.66 \cdot 10^{-6}$

Table 1: ReLU architecture: SAE performance metrics averaged across all configurations (both models, all widths, all sparsity penalties). Values shown as mean $\pm$ SEM. Δ represents the difference between whitened and standard SAE. Bold indicates significantly better performance ($p < 0.05$).

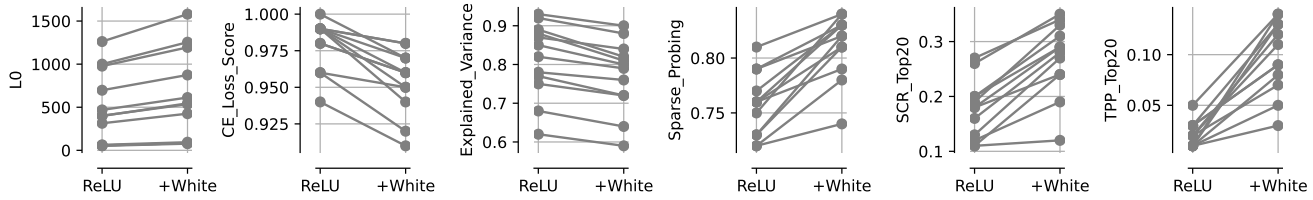

Figure 3: ReLU architecture: Each line connects paired runs before (left) and after whitening (right) averaged across all configurations (both models, all widths, all sparsity penalties). The figure illustrates significant increases in Sparse Probing, SCR, and TPP, accompanied by modest decreases in CE Loss and Explained Variance.

5.2 PCA Whitening

Before training begins, we fit a PCA Whitener to the model's activations collected from the target layer. Activations are sampled in batches of 2048. After sufficient activations are collected to estimate the covariance matrix, we compute the whitening parameters: the mean vector μ, the whitening matrix W, and the dewhitening matrix W^{-1}. These parameters are computed once prior to training and remain fixed throughout both training and evaluation. To compute the whitening transformation, the activation matrix X is first mean-centered:

$$\tilde{X} = X - \mu. \tag{6}$$

We then calculate the covariance matrix of the centered activations,

$$\Sigma = \frac{1}{n-1}\tilde{X}^\top \tilde{X}. \tag{7}$$

Next, we perform eigendecomposition on Σ,

$$\Sigma = EDE^\top, \tag{8}$$

where E contains the eigenvectors and D is a diagonal matrix of eigenvalues. The eigenvectors represent the principal axes of variation in the activation space, while the eigenvalues quantify the variance captured along each corresponding direction. The whitening transformation leverages this decomposition to decorrelate features by rotating the activations into the principal component basis and rescaling them to have unit variance. Formally, the whitening matrix is defined as

$$W = D^{-\frac{1}{2}}E^\top, \tag{9}$$

where each diagonal element of $D^{-\frac{1}{2}}$ is given by $1/\sqrt{\lambda_i + \varepsilon}$, and a small constant ε is added to ensure numerical stability.

A corresponding dewhitening matrix is constructed as

$$W^{-1} = ED^{\frac{1}{2}}, \tag{10}$$

which restores the original scale and covariance structure during reconstruction. Both W and W^{-1} are stored and reused across all forward passes once fitted.

During SAE training, activations are first mean-centered and projected into the whitened space via W. The encoder learns sparse representations from these decorrelated features, and the sparsity penalty is computed in the whitened space. The decoder reconstructs activations in the same space, after which the output is dewhitened using W^{-1} prior to computing the reconstruction loss. This ensures that reconstruction quality is evaluated relative to the model's original activation distribution.

Evaluation To ensure consistent preprocessing, the trained SAE is wrapped with a whitening interface during evaluation. This wrapper automatically whitens input activations before encoding and dewhitens reconstructed activations after decoding, maintaining the same transformations applied during training(W and W^{-1}).

5.3 Results

Our results reveal a consistent pattern: *whitening significantly improves interpretability metrics while slightly reducing reconstruction quality*.

Interpretability gains. As shown in Table 1 and Table 2, whitening yields substantial interpretability improvements across both ReLU and Top-K-based SAEs. For ReLU SAEs, whitening improves sparse probing accuracy by **+7.15%**, SCR by **+54.03%**, and TPP by **+372.00%**, all statistically significant ($p < 0.001$). Top-K SAEs show similar gains in sparse probing (**+7.30%**) with no significant changes in

Metric	Top-k	+Whitening	Δ	$\%\ \Delta$	p-value
CE Loss Score	**0.990** $\pm$ 0.002	0.968 $\pm$ 0.004	-0.022	-2.27%	$4.68 \cdot 10^{-4}$
Explained Variance	**0.837** $\pm$ 0.025	0.794 $\pm$ 0.025	-0.044	-5.22%	$1.12 \cdot 10^{-4}$
Sparse Probing (Top 1)	0.754 $\pm$ 0.005	**0.809** $\pm$ 0.008	+0.055	+7.30%	$2.62 \cdot 10^{-5}$
SCR (Top 20)	0.311 $\pm$ 0.008	0.304 $\pm$ 0.010	-0.008	-2.41%	0.23
TPP (Top 20)	0.141 $\pm$ 0.037	0.152 $\pm$ 0.031	+0.011	+7.96%	0.24

Table 2: Top-k architecture: SAE performance metrics averaged across all configurations (both models, all widths, all target L0s). Values shown as mean $\pm$ SEM. Δ represents the difference between whitened and standard SAE. Bold indicates significantly better performance ($p < 0.05$).

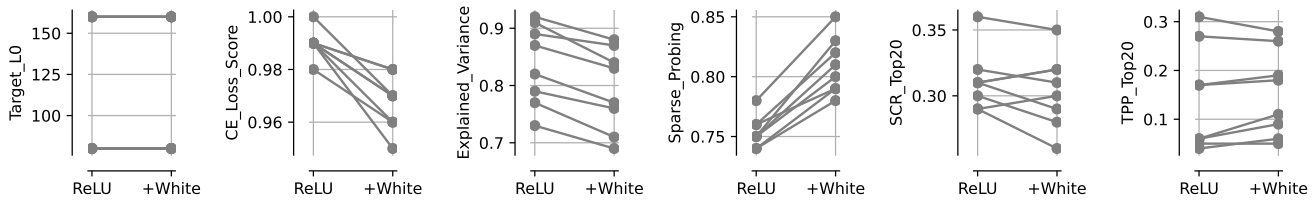

Figure 4: Top-K architecture. Each line connects paired runs before (left) and after whitening (right), averaged across all configurations (both models, widths, and target L0s). The figure shows a strong increase in Sparse Probing with no significant changes in SCR or TPP, alongside small decreases in CE Loss and Explained Variance.

SCR or TPP. These findings indicate that whitening consistently enhances latent interpretability across architectures.

Reconstruction trade-offs. Both architecture types exhibit small yet statistically significant decreases in reconstruction metrics. For ReLU SAEs, CE Loss and explained variance decrease by **2.64%** and **5.02%**, respectively, while Top-K SAEs show reductions of **2.27%** and **5.22%**. However, these decreases are modest compared to the substantial gains in interpretability.

Architecture-specific effects. We observe that the benefits of whitening are more pronounced for ReLU-based architectures across all metrics. This suggests that differences in interpretability improvements across SAE variants may partly arise from implicit whitening effects or improved optimization conditioning inherent to certain architectures.

Visual summary. Figure 3 and Figure 4 provide a paired comparison of metric changes before and after whitening for both ReLU and Top-K architectures.

5.4 Interpretation

Our results challenge the conventional paradigm that optimizing for the sparsity–fidelity trade-off alone yields interpretable features. Despite mild drops in reconstruction quality, PCA-whitened SAEs produced significantly more interpretable representations. This finding is consistent with Karvonen et al. (2025), who report that the Matryoshka SAE achieves the best scores on several SAEBench interpretability metrics while performing worse on the sparsity–fidelity frontier. These results support our theoretical prediction: feature formation based on data structure rather than variance leads to more interpretable features.

Among SAE variants, ReLU models benefit most from whitening. ReLU SAEs impose soft sparsity, enabling distributed representations where features of varying strength encode a concept. Whitening stabilizes these co-activation patterns, helping ReLU SAEs form more disentangled and semantically aligned features. In contrast, Top-K SAEs enforce hard sparsity, activating a fixed number of features per input regardless of concept complexity. This constraint discards weaker yet informative activations, suppressing distributed representations needed for higher-order disentanglement. Whitening still improves the semantic alignment of active features—reflected in higher Sparse Probing—but is less effective for SCR or TPP under rigid sparsity.

6 Conclusion

We have demonstrated that PCA whitening—a classical technique from sparse coding—substantially improves modern sparse autoencoder training. Through theoretical analysis, simulation, and benchmarking, whitening is found to reshape the optimization landscape, yielding significantly more interpretable features. Our results indicate that whitening should be a standard preprocessing step in SAE training.

Moreover, we find that optimizing along the sparsity–fidelity frontier alone does not necessarily yield interpretable representations. This highlights the need for a deeper understanding of how activation geometry shapes learned features. As interpretability becomes increasingly essential for advancing scientific understanding and discovery, whitening offers a simple yet effective means to reveal the structure underlying neural representations.

Limitations and Future Work Our experiments focus on middle layers of language models. Future work should explore whitening's effects across different layers, architectures, and modalities. Additionally, investigating the interaction between whitening and other training innovations (e.g., gated SAEs, transcoders) could yield further improvements.

Acknowledgments

We thank the SAE Bench team for providing comprehensive evaluation tools and the broader interpretability community for valuable discussions.

References

Atick, J. J.; and Redlich, A. N. 1990. Towards a theory of early visual processing. *Neural computation*, 2(3): 308–320.

Atick, J. J.; and Redlich, A. N. 1992. What does the retina know about natural scenes? *Neural computation*, 4(2): 196–210.

Barlow, H. B.; et al. 1961. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01): 217–233.

Bricken, T.; Templeton, A.; Batson, J.; Chen, B.; Jermyn, A.; Conerly, T.; Turner, N.; Anil, C.; Denison, C.; Askell, A.; Lasenby, R.; Wu, Y.; Kravec, S.; Schiefer, N.; Maxwell, T.; Joseph, N.; Hatfield-Dodds, Z.; Tamkin, A.; Nguyen, K.; McLean, B.; Burke, J. E.; Hume, T.; Carter, S.; Henighan, T.; and Olah, C. 2023. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bussmann, B.; Leask, P.; and Nanda, N. 2024. BatchTopK Sparse Autoencoders. arXiv:2412.06410.

Bussmann, B.; Nabeshima, N.; Karvonen, A.; and Nanda, N. 2025. Learning Multi-Level Features with Matryoshka Sparse Autoencoders. arXiv:2503.17547.

Cunningham, H.; Ewart, A.; Riggs, L.; Huben, R.; and Sharkey, L. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Elhage, N.; Hume, T.; Olsson, C.; Schiefer, N.; Henighan, T.; Kravec, S.; Hatfield-Dodds, Z.; Lasenby, R.; Drain, D.; Chen, C.; et al. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Evci, U.; Pedregosa, F.; Gomez, A.; and Elsen, E. 2020. The Difficulty of Training Sparse Neural Networks. arXiv:1906.10732.

Gao, L.; la Tour, T. D.; Tillman, H.; Goh, G.; Troll, R.; Radford, A.; Sutskever, I.; Leike, J.; and Wu, J. 2024. Scaling and evaluating sparse autoencoders. arXiv:2406.04093.

Graham, D. J.; Chandler, D. M.; and Field, D. J. 2006. Can the theory of "whitening" explain the center-surround properties of retinal ganglion cell receptive fields? *Vision research*, 46(18): 2901–2913.

Hyvärinen, A.; Hurri, J.; and Hoyer, P. O. 2001. Independent component analysis. In *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, 151–175. Springer.

Hyvarinen, A.; and Morioka, H. 2017. Nonlinear ICA of temporally dependent stationary sources. In *Artificial intelligence and statistics*, 460–469. PMLR.

Kantamneni, S.; Engels, J.; Rajamanoharan, S.; Tegmark, M.; and Nanda, N. 2025. Are Sparse Autoencoders Useful? A Case Study in Sparse Probing. arXiv:2502.16681.

Karvonen, A. 2025. SAEBench: A Comprehensive Benchmark for Sparse Autoencoders. GitHub repository. Accessed: 2025-10-21.

Karvonen, A.; Rager, C.; Lin, J.; Tigges, C.; Bloom, J.; Chanin, D.; Lau, Y.-T.; Farrell, E.; McDougall, C.; Ayonrinde, K.; et al. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*.

Karvonen, A.; Wright, B.; Rager, C.; Angell, R.; Brinkmann, J.; Smith, L. R.; Verdun, C. M.; Bau, D.; and Marks, S. 2024. Measuring Progress in Dictionary Learning for Language Model Interpretability with Board Game Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Kharlapenko, D.; Shabalin, S.; Barez, F.; Conmy, A.; and Nanda, N. 2025. Scaling sparse feature circuit finding for in-context learning. arXiv:2504.13756.

Klindt, D.; O'Neill, C.; Reizinger, P.; Maurer, H.; and Miolane, N. 2025. From superposition to sparse codes: interpretable representations in neural networks. *arXiv preprint arXiv:2503.01824*.

Klindt, D.; Sanborn, S.; Acosta, F.; Poitevin, F.; and Miolane, N. 2023. Identifying interpretable visual features in artificial and biological neural systems. *arXiv preprint arXiv:2310.11431*.

Leavitt, M. L.; and Morcos, A. 2020. Towards falsifiable interpretability research. *arXiv preprint arXiv:2010.12016*.

Lee, H.; Battle, A.; Raina, R.; and Ng, A. 2006. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19.

Marks, S.; Rager, C.; Michaud, E. J.; Belinkov, Y.; Bau, D.; and Mueller, A. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*.

Ng, A.; et al. 2011. Sparse autoencoder. *CS294A Lecture notes*, 72(2011): 1–19.

Olshausen, B. A.; and Field, D. J. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583): 607–609.

Olshausen, B. A.; and Field, D. J. 1997. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23): 3311–3325.

O'Neill, C.; Gumran, A.; and Klindt, D. 2024. Compute optimal inference and provable amortisation gap in sparse autoencoders. *arXiv preprint arXiv:2411.13117*.

Paulo, G.; and Belrose, N. 2025. Evaluating SAE interpretability without explanations. *arXiv preprint arXiv:2507.08473*.

Rajamanoharan, S.; Conmy, A.; Smith, L.; Lieberum, T.; Varma, V.; Kramár, J.; Shah, R.; and Nanda, N. 2024a. Improving Dictionary Learning with Gated Sparse Autoencoders. arXiv:2404.16014.

Rajamanoharan, S.; Lieberum, T.; Sonnerat, N.; Conmy, A.; Varma, V.; Kramár, J.; and Nanda, N. 2024b. Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders. arXiv:2407.14435.

Templeton, A.; Conerly, T.; Marcus, J.; Lindsey, J.; Bricken, T.; Chen, B.; Pearce, A.; Citro, C.; Ameisen, E.; Jones, A.; Cunningham, H.; Turner, N. L.; McDougall, C.; MacDiarmid, M.; Freeman, C. D.; Sumers, T. R.; Rees, E.; Batson, J.; Jermyn, A.; Carter, S.; Olah, C.; and Henighan, T. 2024. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*.

Zimmermann, R. S.; Borowski, J.; Geirhos, R.; Bethge, M.; Wallis, T.; and Brendel, W. 2021. How well do feature visualizations support causal understanding of CNN activations? *Advances in Neural Information Processing Systems*, 34: 11730–11744.

Zimmermann, R. S.; Klindt, D. A.; and Brendel, W. 2024. Measuring mechanistic interpretability at scale without humans. In *ICLR 2024 Workshop on Representational Alignment*.

A Hyperparameter Configuration

Hyperparameter	Value
Tokens processed	500M
Learning rate	5×10^{-5}
Learning rate warmup (from 0)	1,000 steps
Sparsity penalty warmup (from 0)	5,000 steps
Learning rate decay (to 0)	Last 20% of training
Dataset	The Pile
Batch size	2,048
LLM context length	1,024

Table 3: SAE training hyperparameters.

All Sparse Autoencoders (SAEs) were trained using the *dictionary_learning* repository, following the hyperparameter configurations specified in *SAEBench* (Karvonen et al. 2025). Each SAE was trained on 500M tokens following this configuration. For each [layer, width, architecture] combination, we trained SAEs in a directly comparable manner, maintaining identical data and data ordering across runs. Top-k SAEs were trained with target L_0 values of 80 and 160, as Karvonen et al. (2025) found that although optimal sparsity levels vary substantially across tasks, moderate L_0 values in the range of 50–150 offer a reasonable compromise across metrics.

To fit the PCA whitener, we collected activation batches prior to training. For *Pythia-160M*, we collected 10 batches, yielding an activation matrix of size $20{,}480 \times 768$. For *Gemma-2-2B*, we collected 16 batches, resulting in an activation matrix of size $32{,}768 \times 2{,}304$. These matrices were used to compute the whitening transformation applied before SAE training.

B Simulation Details

B.1 Data Generation

We generated synthetic 2D data to create a controlled environment for analyzing optimization landscapes:

Sparse Sources. We sampled 10,000 points from a uniform grid over $[-1, 1]^2$, then applied a rotation and nonlinear transformation to create sparse, super-Gaussian sources $\mathbf{z} \in \mathbb{R}^{10000\times 2}$. The sources were designed to have heavy tails and sparsity structure similar to natural latent variables in neural networks.

True Dictionary. We generated a random mixing matrix $\mathbf{A} \in \mathbb{R}^{2\times 2}$ from a Gaussian distribution, shifted to ensure well-separated dictionary columns. This represents the "ground truth" feature directions we aim to recover.

Observed Data. Neural activations were generated as $\mathbf{y} = \mathbf{zA}$, creating correlated observations with anisotropic variance—mimicking the structure found in real neural network activations.

Whitened Data. We applied PCA whitening to obtain $\mathbf{y}_{\text{white}} = \mathbf{y}\mathbf{W}_{\text{whiten}}$, where $\mathbf{W}_{\text{whiten}} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^\top$ is computed from the eigendecomposition of the covariance matrix $\mathbf{C} = \mathbf{y}^\top\mathbf{y}$. The whitening matrix was normalized to preserve overall scale.

B.2 Landscape Computation

To visualize the complete optimization landscape, we exhaustively evaluated two metrics across all possible 2D dictionary configurations:

Parameter Grid. We discretized the dictionary space using 1,024 angles for each of the two dictionary vectors: $\theta_0, \theta_1 \in \{0, \frac{2\pi}{1024}, \frac{4\pi}{1024}, \ldots, 2\pi\}$. Each point (θ_0, θ_1) defines a candidate dictionary:

$$\mathbf{W}(\theta_0, \theta_1) = \begin{bmatrix} \cos\theta_0 & \sin\theta_0 \\ \cos\theta_1 & \sin\theta_1 \end{bmatrix} \tag{11}$$

Sparsity Metric. For each candidate dictionary $\mathbf{W}$, we computed the inverse mean L1 norm of the reconstructed sources:

$$\mathcal{S}(\mathbf{W}) = \frac{1}{\mathbb{E}_{\mathbf{y}}[|\mathbf{W}^{-1}\mathbf{y}|_1]} \tag{12}$$

Higher values indicate sparser reconstructions, which is desirable in sparse coding.

Feature Recovery Metric. We measured how well $\mathbf{W}$ recovers the true dictionary $\mathbf{A}$ using cosine similarity:

$$\mathcal{R}(\mathbf{W}, \mathbf{A}) = \max(a, b) \tag{13}$$

where

$$a = \frac{|\mathbf{A}[0] \cdot \mathbf{W}[0]| + |\mathbf{A}[1] \cdot \mathbf{W}[1]|}{2} \tag{14}$$

and

$$b = \frac{|\mathbf{A}[0] \cdot \mathbf{W}[1]| + |\mathbf{A}[1] \cdot \mathbf{W}[0]|}{2} \tag{15}$$

This metric is invariant to permutations and sign flips of dictionary columns. Values near 1 indicate accurate recovery.

Landscape Construction. We computed both metrics for all $1024 \times 1024 = 1{,}048{,}576$ grid points, yielding four landscape matrices:

- $\mathcal{S}(\mathbf{W})$ and $\mathcal{R}(\mathbf{W}, \mathbf{A})$ for non-whitened data
- $\mathcal{S}_{\text{white}}(\mathbf{W})$ and $\mathcal{R}_{\text{white}}(\mathbf{W}, \mathbf{A}_{\text{white}})$ for whitened data

Note that for whitened data, the ground-truth dictionary is transformed to $\mathbf{A}_{\text{white}} = \mathbf{A}\mathbf{W}_{\text{whiten}}$.

B.3 Visualization

The landscapes were visualized as 3D surface plots with the following design choices:

Figure Layout. Each figure shows two side-by-side 3D surfaces comparing non-whitened (left) versus whitened (right) landscapes.

Surface Height and Color. Two complementary views are provided:

- **View A**: Surface height represents feature recovery $\mathcal{R}$, colored by sparsity $\mathcal{S}$ (viridis colormap)
- **View B**: Surface height represents sparsity $\mathcal{S}$, colored by feature recovery $\mathcal{R}$ (magma colormap)

Camera Angle. All 3D plots use elevation = 40° and azimuth = 10° for consistent viewing angles.

B.4 Key Observations

The exhaustive landscape analysis revealed several critical differences:

1. **Isotropy**: Whitened landscapes are approximately rotationally symmetric, while non-whitened landscapes show strong directional biases aligned with the principal components of the data.
2. **Convexity**: The whitened landscape exhibits a smooth, bowl-like basin around the optimum, whereas the non-whitened landscape has multiple local optima and saddle points.
3. **Objective Alignment**: In whitened space, regions of high sparsity strongly overlap with regions of accurate feature recovery. In non-whitened space, these objectives can be misaligned—pursuing sparsity may lead away from the true features.
4. **Gradient Quality**: The Hessian conditioning at the optimum is significantly better after whitening, as evidenced by the smoother, less elongated basin shape.

These geometric properties directly translate to improved optimization: gradient descent on whitened data converges faster, is less sensitive to learning rate, and finds better minima corresponding to more interpretable features. The simulation validates our theoretical analysis and motivates the use of whitening as a standard preprocessing step for SAE training.