Idempotency Exposes Consistency Problems in Sparse Autoencoders

Lennart Stöpler
Institute of Computer Science
Heidelberg University
lennart.stoepler@uni-heidelberg.de

Nikolai Bolik
Institute of Computer Science
Heidelberg University
bolik@uni-heidelberg.de

Artur Andrzejak
Institute of Computer Science
Heidelberg University
artur.andrzejak@uni-heidelberg.de

Abstract

The Linear Representation Hypothesis (LRH) posits that Transformer activations decompose into sparse linear combinations of interpretable directions, with Sparse Autoencoders (SAEs) serving as the predominant method for recovering such decompositions. We propose that the LRH motivates idempotency – the requirement that re-encoding an SAE's output reproduces the same latent representation – as a desirable property of any faithful dictionary learning method, and demonstrate empirically that most SAEs deviate very far from this ideal. We identify two underlying failure modes: exploding representation norms and unstable active feature sets. Further analysis reveals that even individual SAE latents cannot be reliably reconstructed under re-encoding, and that SAE outputs exhibit significant sensitivity to input scaling. In a proof-of-concept study we demonstrate that idempotency can be improved via an auxiliary loss term with mixed results on SAE interpretability.[1]

1 Introduction

The Linear Representation Hypothesis (LRH) (Elhage et al., 2022; Park et al., 2024) has become a cornerstone assumption in mechanistic interpretability, positing that the internal representations of Transformer models (Vaswani et al., 2017) can be expressed as sparse linear combinations of interpretable, nearly orthogonal feature directions (Bricken et al., 2023; Cunningham et al., 2024). Sparse Autoencoders (SAEs) have emerged as the predominant empirical tool for recovering such decompositions, with recent work scaling them to frontier models and treating their latents as meaningful units of analysis (Templeton et al., 2024; Lieberum et al., 2024).

While much work discusses what SAEs can reveal, less attention has been paid to what formal properties a dictionary learning method ought to satisfy to be considered a valid operationalization of the LRH. We argue that idempotency is one such desirable property. Formally, if an SAE were to faithfully decompose an activation vector into its constituent features, then re-encoding the reconstructed output should yield an identical decomposition – the representation is already expressed in the basis the SAE is designed to recover. Of course, real SAEs are never ideal – they are approximate, finite, and trained under competing objectives – and we therefore do not expect any SAE to be perfectly idempotent[2]. Nevertheless, idempotency defines a principled target, and the degree to which a trained SAE departs from it offers a concrete, internally grounded diagnostic of

[1] Code and trained SAEs: https://github.com/aip-hd-research/sae_idempotency

[2] Exact sparse recovery from an overcomplete dictionary is NP-hard, so SAEs are necessarily approximations.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

self-consistency. As we discuss in Appendix A, this criterion emerges naturally from the assumptions underlying the LRH itself: an SAE whose encoder and decoder faithfully recover the ground-truth latent structure must, by construction, satisfy it.

This is not merely a theoretical concern. A strongly non-idempotent SAE conflates properties of the input distribution with properties of the learned dictionary, undermining the interpretability claim that latents[3] correspond to meaningful, reusable features of model computation. Idempotency thus functions as a necessary – though not sufficient – internal consistency check, analogous to the role of calibration in probabilistic models (Guo et al., 2017; Ovadia et al., 2019) or equivariance in geometric deep learning (Cohen and Welling, 2016; Bronstein et al., 2017). Satisfying it is not a guarantee of faithful feature recovery, but systematically failing it is a meaningful warning sign.

We demonstrate that all tested SAEs empirically and systematically violate this criterion, and that many do so to a degree that warrants concern. Through an analysis of failure modes, we identify exploding representation norms and unstable active feature sets as central contributors, and show that these pathologies also manifest at the level of individual latents. We further find that SAE behavior can be highly sensitive to input scaling, indicating a fragility that should be accounted for when assessing claims of robust feature discovery. Taken together, these results motivate additional self-consistency-based evaluation criteria for SAEs, which we derive and propose as complementary diagnostics for future work.

To investigate whether idempotency can be actively improved, we train SAEs with an auxiliary loss term that directly incentivizes idempotency. In this proof-of-concept, the auxiliary loss considerably stabilizes the set and number of active latents and the direction of the reconstruction, showing that these aspects of idempotency can be targeted directly during training. Representation norms, however, continue to grow. We then evaluate whether more idempotent SAEs also perform better on standard interpretability metrics. The results are mixed: some metrics improve, while others degrade, indicating that enforcing idempotency can introduce trade-offs. We take this as an indication that idempotency is relevant to dictionary quality, while it remains one desideratum among several that a well-trained SAE should satisfy.

2 Terms and Definitions

2.1 Linear representation hypothesis

The Linear Representation Hypothesis (Elhage et al., 2022; Park et al., 2024) assumes that there exists a correspondence between features and latents. Moreover, it claims that model behavior can be explained as non-linear operations on these linearly embedded concepts. Specifically, it assumes that a representation vector x in embedding space can be decomposed as

$$x = b + \sum_i f_i d_i \tag{1}$$

for coefficients $f_i \in \mathbb{R}_{\geq 0}$, latents $d_i \in \mathbb{R}^d$, and a bias $b \in \mathbb{R}^d$ (Bricken et al., 2023)[4]. If true, the number of features embedded by the model almost certainly exceeds the dimensionality of the representation space. It follows that not all latents can be orthogonal and thus the presence of one latent will project onto other latents. The model must be robust to this interference.

Dictionary. A *dictionary* is a set of latents whose span is sufficient to explain the model's behavior. That is, the dictionary defines the conceptual coordinate system relative to which we describe representations and interventions.

2.2 Dictionary learning

Dictionary learning aims to construct a dictionary G together with an *encoder* $F : \mathbb{R}^d \to \mathbb{R}^m$ that maps a representation vector x in embedding space to coordinates $z = F(x)$ in the dictionary. These

[3]Interpretability literature (Elhage et al., 2022; Bricken et al., 2023; Cunningham et al., 2024; Park et al., 2024) does not clearly delineate between the human-interpretable concepts it aims to uncover and the mathematical objects we suspect a model might use to encode these concepts. In an attempt to avoid overloading terms too much, in this work we choose to say feature to denote a single interpretable concept and refer to a direction in the model's embedding space as a latent.

[4]There are more complex definitions of the LRH which we highlight in Section 5. However, this is the decomposition assumed in SAE literature.

elements of z indicate the presence and relevance of a dictionary latent within x and thus by extension the presence of its associated feature.
For orthogonal latents, F could be a linear projection on the dictionary. However, since we assume an overcomplete set of latents, the encoder needs to be able to disambiguate the non-unique choice of coordinates with respect to the dictionary. This problem has been studied extensively in compressed sensing literature (Donoho, 2006; Candes et al., 2006).

2.3 Sparse Autoencoders

Sparse Autoencoders are a popular unsupervised dictionary-learning method[5]. Concretely, an SAE is a one-hidden-layer MLP trained as an autoencoder. The dictionary is typically overcomplete ($m \gg d$), and the encoder uses a nonlinearity τ such as ReLU (Bricken et al., 2023; Templeton et al., 2024) or JumpReLU (Rajamanoharan et al., 2024b).
Let $x \in \mathbb{R}^d$ be a representation vector (e.g. a residual-stream activation). With encoder weights $W_e \in \mathbb{R}^{m \times d}$, decoder weights $W_d \in \mathbb{R}^{d \times m}$, and biases $b_e \in \mathbb{R}^m$, $b_d \in \mathbb{R}^d$, the SAE computes[6]

$$F(x) = \tau(W_e(x - b_d) + b_e)\,, \tag{2}$$

$$(G \circ F)(x) = W_d F(x) + b_d. \tag{3}$$

We treat the columns of W_d as the learned latents. They are typically constrained to have unit norm (Olshausen and Field, 1996; Bricken et al., 2023).

3 Idempotency as a Consistency Criterion for Sparse Autoencoders

Given an activation vector $x \in \mathbb{R}^d$, the SAE produces a sparse latent representation $F(x) \in \mathbb{R}^m$ and a reconstruction $(G \circ F)(x)$. This reconstruction can be decomposed as:

$$x = (G \circ F)(x) + \epsilon_x \tag{4}$$

where $\epsilon_x = x - (G \circ F)(x)$ is the residual error term – the component of x that could not be represented without violating sparsity[7]. Crucially, $(G \circ F)(x)$ is by construction exactly representable by the learned dictionary, as it is a sparse linear combination of decoder columns.

This decomposition motivates a self-consistency requirement for the dictionary. If the SAE faithfully[8] identifies the dictionary-representable component of x, then applying the encoder to the reconstruction $(G \circ F)(x)$ should yield the same latent representation as encoding x itself. Formally, we require:

$$(F \circ G \circ F)(x) = F(x) \quad \forall x \tag{5}$$

We refer to this property as *idempotency* of the SAE. Intuitively, the error term ϵ_x should be invisible to the encoder – it represents precisely the part of x that the dictionary cannot express, and re-encoding the reconstruction removes it by construction. An SAE satisfying idempotency therefore cleanly separates x into a dictionary-representable part $(G \circ F)(x)$ and a residual ϵ_x that the encoder consistently ignores.

Violation of idempotency implies that F is sensitive to the error term ϵ_x, meaning the encoder's behavior depends on components of the input it is not designed to represent. The latent $F(x)$ is then not a stable descriptor of the dictionary-representable content of x, but is conflated with properties of the full input distribution.

[5]The LRH does not single out SAEs. They are, however, the dictionary-learning method on which the field has consolidated, with public corpora spanning hundreds of SAEs across model families and architectures (Lieberum et al., 2024; McDougall et al., 2025; Karvonen et al., 2025). Whether other methods decompose activations better is an open question beyond the scope of this work. The idempotency criterion applies equally to closely related encoder–decoder dictionary-learning methods, but we make no claim for structurally different approaches.

[6]We choose a parameterization that includes a pre-encoder bias in line with work by Anthropic (Bricken et al., 2023). This conceptualizes the decoder bias as a constant shift in the data and is closest to the formulation in Equation 1. This is not a restriction on the type of SAE. SAEs not trained with a pre-encoder bias can readily be converted to this form post-training.

[7]As the dictionary is assumed to be overcomplete any input could be perfectly reconstructed were it not for the sparsity requirement.

[8]Throughout, we use *faithful* in the sense of consistency with the LRH decomposition (Equation (1)), not in the sense of reconstruction fidelity as measured, e.g., by Cross Entropy Loss Recovered. Reconstruction fidelity at a given sparsity is an unsupervised proxy of unclear practical relevance on its own (Karvonen et al., 2025).

We do not claim that a sparse decomposition carries semantic meaning a priori. Our argument instead rests on how SAE latents are used in practice. Natural-language interpretations are assigned to latents from their activating contexts, either manually (Bricken et al., 2023; Templeton et al., 2024) or via an LLM judge (Bills et al., 2023; Paulo et al., 2025). Circuit discovery attributes behavior to configurations of co-active latents and the edges between them (Marks et al., 2025), steering uses latents to manipulate model behavior (Templeton et al., 2024; Cho and Hockenmaier, 2025), and the supervised evaluations of SAEBench ablate selected subsets of latents (Karvonen et al., 2025). All of these methods treat the active support of $F(x)$ as a joint account of x, as the LRH suggests (Equation (1)). If this support is not a stable descriptor of the dictionary-representable content of x, the same content can receive different latent explanations depending on ϵ_x, which undermines the reliability of methods that interpret, connect, or ablate active latents. The converse does not hold: idempotency is necessary for support stability, but guarantees neither faithful feature recovery nor semantic interpretability. We discuss this further in Appendix A.

4 Experiments

4.1 Operationalizing Idempotency

The idempotency criterion defined in Equation (5) is binary by definition. However, in practice it is more informative to evaluate the degree to which an SAE violates it. To do so, we apply the SAE iteratively, defining the re-encoding sequence:

$$x_0 = x \qquad x_t = (G \circ F)(x_{t-1}), \tag{6}$$

for $t = 1, \ldots, T$. As stated in Equation (5), a perfectly idempotent SAE would recover the same latent code under repeated re-encoding and therefore produce $x_t = x_1$ for all $t \geq 1$. We propose the following metrics to quantify deviations from this ideal:

1. *IoU.* Idempotency requires not only that the reconstruction x_t remain stable under re-encoding, but crucially that the set of active latents does so as well. We hold that for interpretability, stability of the active support matters more than minimization of the reconstruction error, as it is the choice of active latents that reflects the conceptual decomposition assigned to an input (Marks et al., 2025; Cho and Hockenmaier, 2025). We therefore treat support stability as our primary metric, measured via intersection-over-union $|S_t \cap S_1|/|S_t \cup S_1|$, where S_t denotes the support of $F(x_{t-1})$.[9]
2. ℓ_0*-Norm.* Since the size of the support is a major confounder for *IoU*, we further record the number of active latents $\|F(x_{t-1})\|_0$. This allows us to distinguish shifts in the active feature set from cases in which the active feature set grows.[10]
3. *Cosine similarity.* Cosine similarity between two representation vectors is commonly used to quantify semantic closeness. We therefore track the directional deviation $\langle x_t, x_0 \rangle / (\|x_t\| \|x_0\|)$ from the initial input.
4. ℓ_2*-Norm.* We further track change in ℓ_2-norm $\|x_t\|_2$ between iterations.
5. *MSE.* Finally, we measure the mean-squared error $\|x_t - x_0\|_2^2$ between the original and re-encoded activations and use it as an indicator of total reconstruction drift.

These metrics serve as our selection criteria throughout the experiments that follow: a well-behaved SAE should exhibit IoU and cosine similarity close to 1, near constant ℓ_0, as well as stable MSE and ℓ_2-norm across all iterations.

4.2 Experiment 1: Are SAEs idempotent?

Setup. To evaluate the idempotency of current SAEs, we sample 10,000 tokens from the C4 dataset (Raffel et al., 2020) and extract residual stream activations from a representative set of models

[9] The value of IoU is undefined if the union is empty. For the purposes of our figures we define its value to be zero to indicate pathology.

[10] To ease comparison across SAE widths we normalize this metric by the width of the SAE if that improves readability.

[11] For each Gemma 3 SAE width we report results for both the low ℓ_0 (small) and high ℓ_0 (big) version.

Table 1: SAEs used in this work.[11]

Model Family	Model Size	SAE Size	SAE Type	Author
Gemma 2	2B	4K, 16K, 65K	JumpReLU, Matryoshka BatchTopK	Karvonen et al. (2025)
Gemma 3	270M, 1B, 4B, 12B, 27B	16K, 262K	Matryoshka JumpReLU	McDougall et al. (2025)
GPT2-small	124M	32K, 128K	TopK	Gao et al. (2025)
Llama 3.1	8B	32K, 131K	JumpReLU	He et al. (2024)
Mistral	7B	65K	ReLU	Engels et al. (2025a)

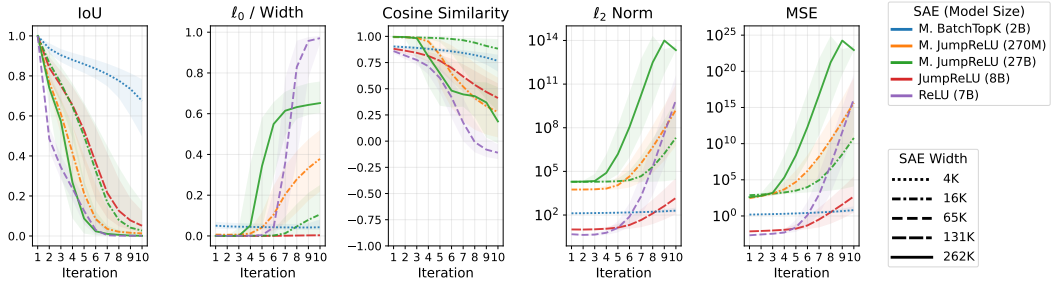


Figure 1: Idempotency metrics over 10 re-encoding iterations for Matryoshka BatchTopK 4K (Gemma 2 2B); Matryoshka JumpReLU 16K (big) (Gemma 3 270M); Matryoshka JumpReLU 16K (small) / 262K (big) (Gemma 3 27B), JumpReLU 131K (Llama 3.1 8B), and ReLU 65K (Mistral 7B). Lines show the median over tokens, shaded bands the interquartile range (25th to 75th percentile); both are averaged over layers.

spanning a range of parameter scales[12]: Gemma 2 2B (Gemma Team et al., 2024), Gemma 3 (270M, 1B, 4B, 12B, and 27B) (Gemma Team et al., 2025), GPT2-small (Radford et al., 2019), Llama 3.1 8B (Grattafiori et al., 2024), and Mistral 7B (Jiang et al., 2023). A detailed listing of the SAEs used in this work can be found in Table 1. For each activation x_0, we apply the SAE 10 times[13], defining the sequence $x_t = (G \circ F)(x_{t-1})$ for $t = 1, \ldots, 10$.[14]

Results. Figure 1 reports idempotency metrics across all 10 re-encoding iterations for a representative subset of models and SAE widths; results for all remaining model and SAE combinations, as well as an analysis of scaling behaviors, are provided in Appendix B. For each metric and layer we first calculate the median and the 25th and 75th percentiles across all tokens. We then average these percentiles over all layers of that model and SAE combination and plot the median together with the interquartile range [p25, p75].[15] The results are unambiguous: *all tested SAEs show degradation in idempotency metrics*. For all models, IoU degrades with each iteration, collapsing to near zero by iteration 10 in most cases. For many SAEs the ℓ_0-norm remains stable for one to five iterations before exploding, though some SAE types remain stable considerably longer. Crucially, IoU decreases even when ℓ_0 is stable, indicating a shift in *which* latents are active rather than merely how many. Cosine similarity likewise degrades rapidly, implying a semantic shift in the reconstructions. The ℓ_2-norm and MSE are strongly correlated and both increase by orders of magnitude – usually after an initial stable phase that broadly coincides with the stable phase in ℓ_0 – though some SAE types show growing ℓ_2 and MSE even while ℓ_0 remains constant.[16]

A notable exception in degree, though not in kind, is Gemma 2 2B with Matryoshka BatchTopK SAEs which exhibits significantly more stable behavior across all metrics than any other configuration we tested. This is particularly noteworthy given that this class of SAE also achieves the strongest performance on interpretability benchmarks in SAEBench (Karvonen et al., 2025), which may point to a link between idempotency and broader notions of dictionary quality.

[12]All results are for the pretrained versions of these models, before any instruction tuning.

[13]All experiments were performed using SAELens (Bloom et al., 2024) and TransformerLens (Nanda and Bloom, 2022).

[14]In preliminary experiments 10 iterations proved sufficient to identify the qualitative trends in all tested SAEs.

[15]The Matryoshka BatchTopK SAEs are only available for layer 12. In this case we aggregate across sparsity levels instead. We make this exception in all experiments.

[16]Since SAEs only approximate sparse recovery (Section 5), some deviation is expected. A small, bounded approximation error, however, does not explain deviations that compound over orders of magnitude.

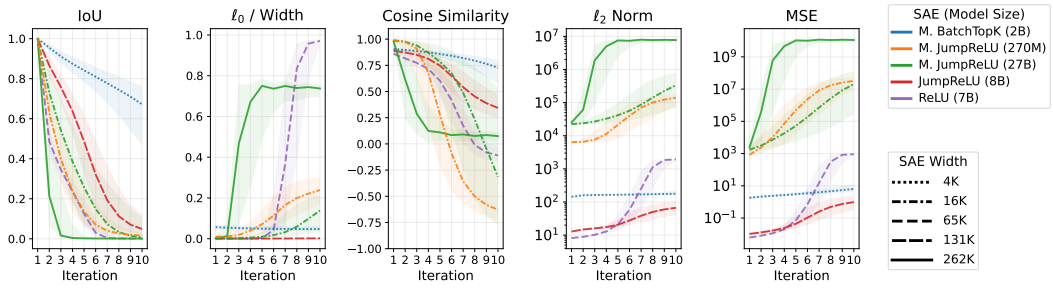


Figure 2: Idempotency metrics under norm-controlled re-encoding for the same selection of SAEs as in Fig. 1. Each input is rescaled to the mean layer activation norm μ_ℓ prior to each SAE application. Lines and bands as in Fig. 1.

4.3 Is scaling alone the reason?

By definition, affine SAE encoders are sensitive to input scale. Since the dictionary is overcomplete, scaling the input can increase projections onto further encoder directions beyond their respective bias thresholds, thereby changing the support of $F(x)$. It follows that SAEs can be expected to break down if the ℓ_2-norm of their input lies too far out-of-distribution. However, the results of Section 4.2 suggest that this sensitivity might be excessive.

4.3.1 Experiment 2: How sensitive are SAEs to input-scaling?

Table 2: Results of scaling the input to the SAE ($c = 1.2$). Entries are median [25th, 75th percentile] over tokens, averaged over layers.[17]

Model	SAE Type	SAE Size	$\|F(x)\|_0$	$\|F(\lambda(x))\|_0$	Identical Tokens	$\|\epsilon\|_2/\sigma$
Gemma 2 2B	M. BatchTopK	4K	*1.00 [0.74, 1.37]*	*1.37 [1.03, 1.84]*	87 [78, 95]%	0.20 [0.18, 0.22]
Gemma 3 270M	M. JumpReLU	16K (big)	106 [86, 130]	168 [142, 198]	97 [96, 98]%	0.17 [0.14, 0.20]
Gemma 3 27B	M. JumpReLU	16K (small)	20 [16, 25]	32 [26, 40]	100 [98, 100]%	0.05 [0.04, 0.06]
	M. JumpReLU	262K (big)	117 [102, 135]	1658 [380, 5513]	99 [97, 99]%	0.08 [0.07, 0.09]
Llama 3.1 8B	JumpReLU	131K	36 [29, 43]	50 [40, 60]	100 [99, 100]%	0.14 [0.12, 0.15]
Mistral 7B	ReLU	65K	86 [75, 96]	45014 [44340, 46793]	37 [23, 53]%	0.14 [0.14, 0.15]

Setup. We quantify the effect of input scale by slightly upscaling the activation x_0 with a scaling coefficient c. To remain consistent with the LRH in Section 2.1 and the definition of the SAE decoder in Equation 3 we first subtract the decoder bias to get the scaled activations $\lambda(x_0)$:

$$\lambda(x_0) := c \cdot (x_0 - b_d) + b_d. \tag{7}$$

We compare the ℓ_0-norm before and after scaling. An idealized SAE would preserve the active support under such transformations and should therefore exhibit little to no change in ℓ_0.[18] To quantify the semantic change induced by this scaling operation we apply the model's unembedding matrix[19] to the activations before and after scaling. We report the percentage of tokens that remained among the top-100 logits after scaling. Finally, we report the ℓ_2-norm of the change in activation $\epsilon = \lambda(x_0) - x_0$ as a fraction of the standard deviation σ of activation norms $\|x_0\|_2$ at this layer.

Results. Table 2 shows the results for scaling coefficient $c = 1.2$ and the same model choice as in Section 4.2; further results for all models, SAE types, and coefficients $c = 1.1$ and $c = 1.3$ are

[17] The Gemma 2 2B SAEs differ in their target sparsity, so their absolute ℓ_0 values are not directly comparable. We therefore report their ℓ_0 entries (in italics) relative to the median ℓ_0 of the unscaled input.

[18] An argument can be made that even idealized dictionary learning methods only tolerate scaling that is in-distribution and that scaling beyond that point causes positive interference indistinguishable from active latents. A further argument can be made the in-distribution scale can vary between latents and that by scaling up the activations uniformly we potentially lift individual latents out-of-distribution, even if the overall change in ℓ_0-norm is in-distribution. We believe this effect to be rather minor, but cannot conclusively say how big its impact is.

[19] Where applicable we fold the final Layernorm (Ba et al., 2016) into the unembedding matrix, as justified in (Elhage et al., 2021).

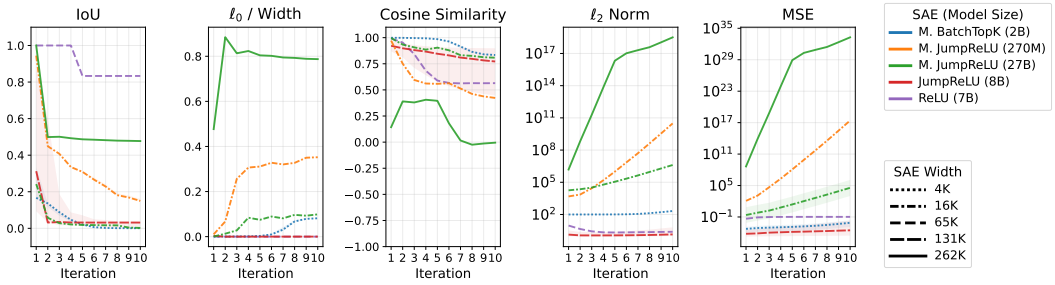


Figure 3: Idempotency metrics applied to synthetic single-latent inputs of the form $x_0 = \bar{f}_i d_i + b_d$ for the same selection of SAEs as Fig. 1. Lines show the median over latents, shaded bands the interquartile range; both are averaged over layers.

in Appendix C. Despite strong performance in the previous experiment, Matryoshka BatchTopK SAEs remain quite sensitive to input scale, activating over 30% more latents on the perturbed input. Matryoshka JumpReLU SAEs trained with a higher ℓ_0 target (big) are extremely susceptible, while their more sparse counterparts (small) are less so but still activate about 50% more latents, suggesting that sparsity target plays an important role in scaling stability.

The ReLU SAEs present a curious outlier. Like all others, the ϵ change lies within the normal input distribution, yet this is sufficient to massively change the unembedding and activate nearly all latents. We suspect this is related to the decoder bias for this class of SAE being extremely large – in some cases up to four times the mean activation norm.

4.3.2 Experiment 3: Does fixing the input-scale help?

Setup. To isolate the contribution of input scale to the idempotency failures observed in Section 4.2, we repeat the experiment with a single modification. Before each SAE application, we rescale the input activation to the mean ℓ_2-norm of activations in that layer. That is, we define

$$\tilde{x}_t = x_t \cdot \mu_\ell / \|x_t\|_2, \tag{8}$$

where μ_ℓ is the mean activation norm at layer ℓ, and apply the SAE to $\tilde{x}_t$ at each iteration. This ensures that all inputs to the SAE have identical norm, eliminating input scale as a confounding factor while preserving directional content.

Results. Figure 2 shows the same metrics as Section 4.2 under the norm-controlled protocol. Despite fixing the input norm at every iteration, the ℓ_2-norm and MSE still rise sharply before saturating, albeit at lower values than before. Critically, IoU and cosine similarity collapse at a rate comparable to the base case. Input scale is therefore likely a contributing factor, but not a sufficient explanation: the SAE induces systematic changes to the direction of the activation and the set of active latents regardless of input magnitude, pointing to a pathology intrinsic to the learned dictionary and encoder geometry.

4.4 Experiment 4: What happens to individual latents?

Setup. To isolate the behavior of individual dictionary elements under re-encoding, we construct synthetic inputs of the form $x_0 = \bar{f}_i d_i + b_d$, where $d_i = (W_d)_{:,i}$ is the i-th column of the decoder matrix and b_d is the decoder bias. $\bar{f}_i$ is the mean activation $\overline{F(x)_i}$ of the latent d_i over the same 10,000 tokens as before. We do this for all dictionary latents d_i. By construction, each such input is a single active dictionary latent shifted by the decoder bias, and an idempotent SAE should recover exactly this latent upon re-encoding. We apply the same iterative re-encoding protocol as in Sections 4.2 and 4.3.2, recording the same five metrics over 10 iterations.

Results. The results are shown in Figure 3. As before we first calculate the median and interquartile range over all latents of a layer before averaging these layer-wise percentiles. We observe three qualitative behaviors. First, some SAEs – such as Matryoshka JumpReLU (270M) – behave similarly

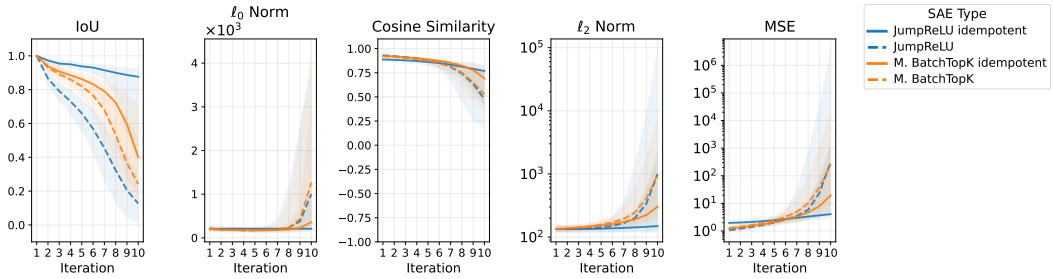


Figure 4: Idempotency metrics following the same setup as Experiment 1 (Sec. 4.2) for 16K JumpReLU and Matryoshka BatchTopK SAEs on Gemma 2 2B, trained with and without auxiliary intersection loss. Lines show the median over tokens, shaded bands the interquartile range; both are averaged over sparsity targets.

to Experiment 1, sometimes even more extremely[20]. Second, some SAEs show IoU well below one from the first iteration, as most inputs fail to activate any latent. These SAEs appear deceptively stable at first glance but this is really an artifact of the decoder bias dominating the results.[21] This includes the Matryoshka BatchTopK SAE that performed best in Experiment 1. Third, ReLU SAEs exhibit near-idempotent behavior: IoU is close to one and ℓ_0, ℓ_2, and MSE are stable, though a significant drift in cosine similarity persists, indicating that activation values shift even when the active set does not.

Exceptions. While no class of SAE is idempotent in aggregate, an analysis of individual SAEs reveals a significant number of outliers; full details and a breakdown by latent index for Matryoshka SAEs are in Appendix E. Notably, 10 SAEs (1.1% of the total) are—up to a small error—idempotent on the synthetic inputs, showing that while uncommon, such SAEs are possible.

4.5 Experiment 5: Can idempotency be improved?

Setup. To test whether idempotency can be improved, we train idempotent variants of the SAEBench JumpReLU and Matryoshka BatchTopK SAEs using the exact training regime of the original authors, with six versions per architecture at varying sparsity targets. Our goal is to directly optimize the IoU of one re-encoding cycle. We hypothesize that the sparsity constraints already provide a gradient signal to minimize the union, and add the following loss term of the SAE to maximize the intersection:

$$-\zeta \left\| F(x) \odot (F \circ G \circ F)(x) \right\|_0 , \tag{9}$$

where $\odot$ denotes element-wise multiplication. We add this term, weighted by ζ, to the standard SAE training objective and optimize it jointly with the reconstruction and sparsity terms. This requires one additional encode–decode–encode pass per training step, while the trained SAE is applied exactly as before. Full training details are in Appendix F. We evaluate the new SAEs on the same four experiments and the full SAEBench suite (Karvonen et al., 2025).

Results. In Figure 4 we report the results on Experiment 1 for the 16K versions of the SAEs trained with intersection loss, alongside the results for the version of these SAEs trained without the extra loss. We average the per-token median and interquartile range across all sparsity targets. Figure 5 shows the results on SAEBench. The full results for all other experiments and for SAE sizes 4K and 65K are available in Appendix G.

The auxiliary loss improves the stability of the active support, measured by IoU and ℓ_0, as well as cosine similarity, while ℓ_2 and MSE do not improve. The improvement is significantly more pronounced for JumpReLU. Indeed, JumpReLU trained with the auxiliary loss becomes by far the best performing SAE type in terms of support stability. We speculate that this is because JumpReLU directly parameterizes the ℓ_0-norm, making it more responsive to a loss that penalizes changes in the active support. This trend holds consistently across Experiments 2, 3, and 4.

[20]The 262K (big) Matryoshka JumpReLU (27B) SAE activates almost 50% of its latents in the first iteration and more than 80% by iteration 2; the ground-truth ℓ_0 is 1.

[21]This is discussed in more detail in Appendix E.

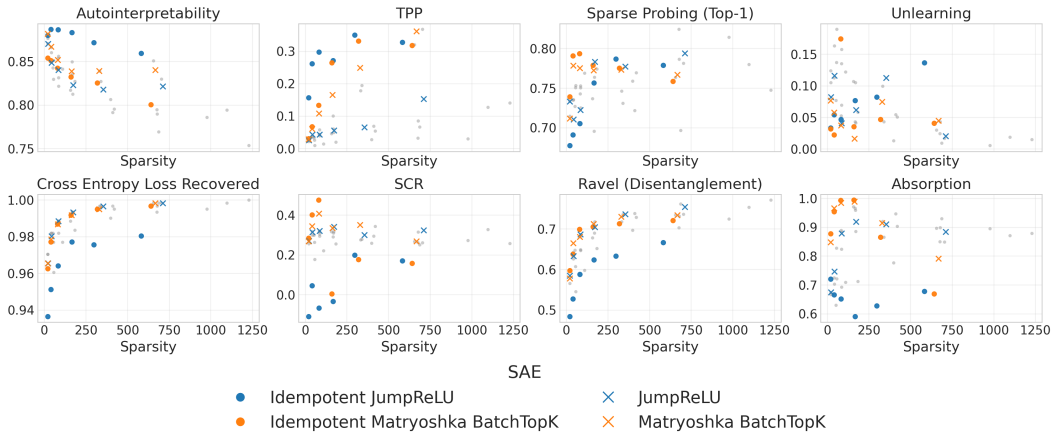


Figure 5: Results on SAEBench for the same SAEs as in Fig. 4. The gray dots indicate the full set of SAEs published in SAEBench. The x-axis ("Sparsity") shows the ℓ_0 of each SAE, so low values correspond to sparser SAEs.

The picture on SAEBench is more mixed. For JumpReLU, the auxiliary loss improves Autointerpretability and TPP scores, with the idempotent variant establishing a new Pareto frontier on these metrics. At the same time it leads to a decrease on the remaining metrics including Cross Entropy Loss Recovered. For Matryoshka BatchTopK the effect is less pronounced: Cross Entropy Loss Recovered remains almost identical to the baseline, Autointerpretability degrades slightly, and on all other metrics the idempotent variant outperforms at low ℓ_0 targets but underperforms at higher ℓ_0 targets. Taken together, these results suggest that enforcing idempotency improves some aspects of dictionary quality while introducing trade-offs on others, and that the nature of these trade-offs might depend on the underlying SAE architecture.

5 Related Work

Linear Representation Hypothesis The LRH was articulated for toy models by Elhage et al. (2022) and transferred to language models by Park et al. (2024). As a working hypothesis, it is supported by a broad body of empirical work: linear probes recover high-level concepts such as space and time (Gurnee and Tegmark, 2024) and truth (Marks and Tegmark, 2024), directional interventions can causally alter refusal behavior (Arditi et al., 2024), and sequence models develop linearly readable and causally manipulable world-state features (Nanda et al., 2023). SAE studies likewise report human-interpretable and causally effective latents (Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024). In Section 2.1 we presented a minimalistic version of the strong LRH. There is, however, mounting evidence that suggests this form of the LRH is incomplete (Smith, 2024). There is significant geometric structure in the features uncovered by SAEs that is unaccounted for by the LRH (Mendel, 2024; Park et al., 2025). Meanwhile, another line of work has demonstrated that circular (Engels et al., 2025a), as well as helical structures (Kantamneni and Tegmark, 2025), are used by the model to perform computation. This promotes the idea of multi-dimensional feature manifolds (Olah and Jermyn, 2024). Finally, Engels et al. (2025b) show that SAE reconstruction errors contain a component that is not linearly predictable from the input, which may reflect structure beyond the linear decomposition.

Sparse Autoencoders Improving SAE architectures is an active area of research. TopK (Gao et al., 2025) and BatchTopK (Bussmann et al., 2024) SAEs have been proposed as ways to encode sparsity constraints directly in the activation functions eschewing the need for a sparsity loss. Gated (Rajamanoharan et al., 2024a), ProLU (Taggart, 2024), and JumpReLU (Rajamanoharan et al., 2024b) SAEs separate the concerns of filtering interference from estimating the magnitude of active latents. Finally, Matryoshka SAEs (Bussmann et al., 2025) use grouped loss terms to deal with hierarchical features.

SAE Criticism The problem of compressed sensing (Donoho, 2006; Candes et al., 2006), that is extracting a sparse representation from an overcomplete dictionary, is combinatorially hard in general. In an abstract sense SAEs can thus be viewed as a strong approximation to a much more complex algorithm (Bricken et al., 2023). They are, therefore, by their very nature, imperfect. In fact, there is reason to believe that their reconstruction error might be pathological and no SAE can ever achieve perfect decomposition (Gurnee, 2024). At the same time it has been demonstrated that the features that SAEs do find are neither atomic (Leask et al., 2025; Bricken et al., 2023) nor complete (Leask et al., 2025; Olah and Jermyn, 2024) and that their interpretability often lags behind supervised baselines (Wu et al., 2025). More recently, individual failure modes of SAE latents have been studied, such as feature splitting (Templeton et al., 2024), feature absorption (Chanin et al., 2024b), and feature hedging (Chanin et al., 2025).

Idempotency in Learned Representations Idempotency has been studied as projection onto a target manifold in generative modeling (Shocher et al., 2024), under repeated encoding in neural audio codecs (O'Reilly et al., 2025), in learned and perceptual image compression (Li et al., 2023; Xu et al., 2024), and in codec forensics (Bestagini et al., 2013), and algebraically for convolution operations on fuzzy truth values (Zhang and Hu, 2022). Autoencoders can be idempotent by construction, for example when the encoder is a left inverse of the decoder (Otto et al., 2023), or can be trained explicitly towards idempotency (Shocher et al., 2024). Conversely, drift under repeated encoding is not unique to SAEs, as neural audio codecs degrade markedly after only a few rounds of re-encoding (O'Reilly et al., 2025).

These works primarily concern stability of generated or reconstructed signals under repeated application or processing, which should not degrade perceptual quality. In our setting the reconstruction is only an intermediate requirement. An SAE output is never consumed by a listener or by the model. The SAE latent activations are read as an explanation of the model's internal state. Our criterion is therefore defined on latent support and motivated by consistency toward the assumed LRH decomposition, rather than by a desirable engineering property. We are not aware of prior work evaluating idempotency as a criterion for dictionary learning on language-model activations.

6 Conclusion

We have argued that idempotency is a desirable property for any SAE that aims to faithfully operationalize the LRH, and demonstrated empirically that current state-of-the-art SAEs deviate from it substantially. Across a diverse set of models and architectures, repeated re-encoding causes IoU and cosine similarity to collapse and representation norms to grow by orders of magnitude. These failures persist even when input scale is controlled for, and extend to individual decoder latents, implicating systematic pathologies in the learned dictionary rather than distributional mismatch alone. Training with an auxiliary loss term improves the stability of the active support (IoU, ℓ_0) as well as cosine similarity, most strongly for JumpReLU SAEs, while the norm-based metrics do not improve. The downstream effect on interpretability benchmarks is mixed: some metrics improve while others degrade, and the pattern differs across architectures. We propose idempotency as a principled complementary diagnostic for SAE evaluation and encourage its adoption alongside existing metrics in future work.

7 Limitations

Several limitations should be noted. First, our analysis is restricted to residual stream activations; whether SAEs trained on other activation sites exhibit qualitatively different idempotency behavior remains open. Second, our coverage is not exhaustive – we do not evaluate Gated SAEs, and results may differ for architectures not represented in our study. Third, the iterative re-encoding protocol is a diagnostic tool rather than a realistic use case, as SAEs are typically applied once per activation. Furthermore, as discussed in Appendix A a lack of idempotency could be explained either by the LRH not holding or by SAEs not recovering it cleanly. In this work we are not able to distinguish the two. Finally, while we provide evidence for a connection between idempotency and interpretability, we could not conclusively demonstrate that this holds across all metrics and architectures.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their constructive feedback, which substantially improved this paper. This work was carried out as part of the authors' regular employment at Heidelberg University and received no external funding. All experiments were run on the compute resources of the AI for Programming group at Heidelberg University. The authors declare no competing interests.

References

Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., and Nanda, N. (2024). Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, pages 136037–136083.

Ba, L. J., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *CoRR*, abs/1607.06450.

Bestagini, P., Milani, S., Tagliasacchi, M., and Tubaro, S. (2013). Video codec identification extending the idempotency property. In *2013 4th European Workshop on Visual Information Processing (EUVIP)*, pages 220–225, Paris, France.

Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J., and Saunders, W. (2023). Language models can explain neurons in language models. OpenAI. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`.

Bloom, J., Tigges, C., Duong, A., and Chanin, D. (2024). SAELens. `https://github.com/decoderesearch/SAELens`.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N. L., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., 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`.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Bussmann, B., Leask, P., and Nanda, N. (2024). BatchTopK sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*.

Bussmann, B., Nabeshima, N., Karvonen, A., and Nanda, N. (2025). Learning multi-level features with matryoshka sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25.

Candes, E. J., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theor.*, 52(2):489–509.

Chanin, D., Bhatnagar, H., Dulka, T., and Bloom, J. (2024a). Toy models of feature absorption in SAEs. LessWrong. `https://www.lesswrong.com/posts/kcg58WhRxFA9hv9vN/toy-models-of-feature-absorption-in-saes`.

Chanin, D., Dulka, T., and Garriga-Alonso, A. (2025). Feature hedging: Correlated features break narrow sparse autoencoders. `https://arxiv.org/abs/2505.11756`.

Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., and Bloom, J. I. (2024b). A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In *Interpretable AI: Past, Present and Future*.

Cho, I. and Hockenmaier, J. (2025). Toward efficient sparse autoencoder-guided steering for improved in-context learning in large language models. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V., editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28961–28973, Suzhou, China. Association for Computational Linguistics.

Cohen, T. S. and Welling, M. (2016). Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, ICML'16, page 2990–2999. JMLR.org.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. (2024). Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Donoho, D. L. (2006). Compressed sensing. *IEEE Trans. Inf. Theor.*, 52(4):1289–1306.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. (2022). Toy models of superposition. *Transformer Circuits Thread*. `https://transformer-circuits.pub/2022/toy_model/index.html`.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*. `https://transformer-circuits.pub/2021/framework/index.html`.

Engels, J., Michaud, E. J., Liao, I., Gurnee, W., and Tegmark, M. (2025a). Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*.

Engels, J., Smith, L. R., and Tegmark, M. (2025b). Decomposing the dark matter of sparse autoencoders. *Transactions on Machine Learning Research*.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. (2020). The pile: An 800gb dataset of diverse text for language modeling. `https://arxiv.org/abs/2101.00027`.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. (2025). Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Gemma Team, Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S., Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X., Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr, I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.-T., Sinopalnikov, D., Bhupatiraju, S., Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A., Friesen, A., Sharma, A., Sharma, A., Gilady, A. M., Goedeckemeyer, A., Saade, A., Feng, A., Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A. S., Das, A., Bapna, A., Miech, A., Yang, A., Paterson, A., Shenoy, A., Chakrabarti, A., Piot, B., Wu, B., Shahriari, B., Petrini, B., Chen, C., Lan, C. L., Choquette-Choo, C. A., Carey, C., Brick, C., Deutsch, D., Eisenbud, D., Cattle, D., Cheng, D., Paparas, D., Sreepathihalli, D. S., Reid, D., Tran, D., Zelle, D., Noland, E., Huizenga, E., Kharitonov, E., Liu, F., Amirkhanyan, G., Cameron, G., Hashemi, H., Klimczak-Plucińska, H., Singh, H., Mehta, H., Lehri, H. T., Hazimeh, H., Ballantyne, I., Szpektor, I., Nardini, I., Pouget-Abadie, J., Chan, J., Stanton, J., Wieting, J., Lai, J., Orbay, J., Fernandez, J., Newlan, J., yeong Ji, J., Singh, J., Black, K., Yu, K., Hui, K., Vodrahalli, K., Greff, K., Qiu, L., Valentine, M., Coelho, M., Ritter, M., Hoffman, M., Watson, M., Chaturvedi, M., Moynihan, M., Ma, M., Babar, N., Noy, N., Byrd, N., Roy, N., Momchev, N., Chauhan, N., Sachdeva, N., Bunyan, O., Botarda, P., Caron, P., Rubenstein, P. K., Culliton, P., Schmid, P., Sessa, P. G., Xu, P., Stanczyk, P., Tafti, P., Shivanna, R., Wu, R., Pan, R., Rokni, R., Willoughby, R., Vallu, R., Mullins, R., Jerome, S., Smoot, S., Girgin, S., Iqbal, S., Reddy, S., Sheth, S., Põder, S., Bhatnagar, S., Panyam, S. R., Eiger, S., Zhang, S., Liu, T., Yacovone, T., Liechty, T., Kalra, U., Evci, U., Misra, V., Roseberry, V., Feinberg, V., Kolesnikov, V., Han, W., Kwon, W., Chen, X., Chow, Y., Zhu, Y., Wei, Z., Egyed, Z., Cotruta, V., Giang, M., Kirk, P., Rao, A., Black, K., Babar, N., Lo, J., Moreira, E., Martins, L. G., Sanseviero, O., Gonzalez, L., Gleicher, Z., Warkentin, T., Mirrokni, V., Senter, E., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Matias, Y., Sculley, D., Petrov, S., Fiedel, N., Shazeer, N., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya,

E., Alayrac, J.-B., Anil, R., Dmitry, Lepikhin, Borgeaud, S., Bachem, O., Joulin, A., Andreev, A., Hardin, C., Dadashi, R., and Hussenot, L. (2025). Gemma 3 technical report. Technical report, Google DeepMind. https://arxiv.org/abs/2503.19786.

Gemma Team, Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., Ferret, J., Liu, P., Tafti, P., Friesen, A., Casbon, M., Ramos, S., Kumar, R., Lan, C. L., Jerome, S., Tsitsulin, A., Vieillard, N., Stanczyk, P., Girgin, S., Momchev, N., Hoffman, M., Thakoor, S., Grill, J.-B., Neyshabur, B., Bachem, O., Walton, A., Severyn, A., Parrish, A., Ahmad, A., Hutchison, A., Abdagic, A., Carl, A., Shen, A., Brock, A., Coenen, A., Laforge, A., Paterson, A., Bastian, B., Piot, B., Wu, B., Royal, B., Chen, C., Kumar, C., Perry, C., Welty, C., Choquette-Choo, C. A., Sinopalnikov, D., Weinberger, D., Vijaykumar, D., Rogozińska, D., Herbison, D., Bandy, E., Wang, E., Noland, E., Moreira, E., Senter, E., Eltyshev, E., Visin, F., Rasskin, G., Wei, G., Cameron, G., Martins, G., Hashemi, H., Klimczak-Plucińska, H., Batra, H., Dhand, H., Nardini, I., Mein, J., Zhou, J., Svensson, J., Stanway, J., Chan, J., Zhou, J. P., Carrasqueira, J., Iljazi, J., Becker, J., Fernandez, J., van Amersfoort, J., Gordon, J., Lipschultz, J., Newlan, J., yeong Ji, J., Mohamed, K., Badola, K., Black, K., Millican, K., McDonell, K., Nguyen, K., Sodhia, K., Greene, K., Sjoesund, L. L., Usui, L., Sifre, L., Heuermann, L., Lago, L., McNealus, L., Soares, L. B., Kilpatrick, L., Dixon, L., Martins, L., Reid, M., Singh, M., Iverson, M., Görner, M., Velloso, M., Wirth, M., Davidow, M., Miller, M., Rahtz, M., Watson, M., Risdal, M., Kazemi, M., Moynihan, M., Zhang, M., Kahng, M., Park, M., Rahman, M., Khatwani, M., Dao, N., Bardoliwalla, N., Devanathan, N., Dumai, N., Chauhan, N., Wahltinez, O., Botarda, P., Barnes, P., Barham, P., Michel, P., Jin, P., Georgiev, P., Culliton, P., Kuppala, P., Comanescu, R., Merhej, R., Jana, R., Rokni, R. A., Agarwal, R., Mullins, R., Saadat, S., Carthy, S. M., Cogan, S., Perrin, S., Arnold, S. M. R., Krause, S., Dai, S., Garg, S., Sheth, S., Ronstrom, S., Chan, S., Jordan, T., Yu, T., Eccles, T., Hennigan, T., Kocisky, T., Doshi, T., Jain, V., Yadav, V., Meshram, V., Dharmadhikari, V., Barkley, W., Wei, W., Ye, W., Han, W., Kwon, W., Xu, X., Shen, Z., Gong, Z., Wei, Z., Cotruta, V., Kirk, P., Rao, A., Giang, M., Peran, L., Warkentin, T., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Sculley, D., Banks, J., Dragan, A., Petrov, S., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Borgeaud, S., Fiedel, N., Joulin, A., Kenealy, K., Dadashi, R., and Andreev, A. (2024). Gemma 2: Improving open language models at a practical size. Technical report, Google DeepMind. https://arxiv.org/abs/2408.00118.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur,

Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. (2024). The llama 3 herd of models. `https://arxiv.org/abs/2407.21783`.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org.

Gurnee, W. (2024). SAE reconstruction errors are (empirically) pathological. *LessWrong*. `https://www.lesswrong.com/posts/rZPiuFxESMxCDHe4B/sae-reconstruction-errors-are-empirically-pathological`.

Gurnee, W. and Tegmark, M. (2024). Language models represent space and time. In *The Twelfth International Conference on Learning Representations*.

He, Z., Shu, W., Ge, X., Chen, L., Wang, J., Zhou, Y., Liu, F., Guo, Q., Huang, X., Wu, Z., Jiang, Y.-G., and Qiu, X. (2024). Llama Scope: Extracting millions of features from Llama-3.1-8B with sparse autoencoders. `https://arxiv.org/abs/2410.20526`.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7b. `https://arxiv.org/abs/2310.06825`.

Kantamneni, S. and Tegmark, M. (2025). Language models use trigonometry to do addition. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J. I., Chanin, D., Lau, Y.-T., Farrell, E., McDougall, C. S., Ayonrinde, K., Till, D., Wearden, M., Conmy, A., Marks, S., and Nanda, N. (2025). SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Forty-second International Conference on Machine Learning*.

Leask, P., Bussmann, B., Pearce, M. T., Bloom, J. I., Tigges, C., Moubayed, N. A., Sharkey, L., and Nanda, N. (2025). Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations*.

Li, Y., Xu, T., Wang, Y., Liu, J., and Zhang, Y.-Q. (2023). Idempotent learned image compression with right-inverse. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 12878–12896. Curran Associates, Inc.

Lieberum, T., Rajamanoharan, S., Conmy, A., Smith, L., Sonnerat, N., Varma, V., Kramar, J., Dragan, A., Shah, R., and Nanda, N. (2024). Gemma scope: Open sparse autoencoders everywhere all at once on Gemma 2. In Belinkov, Y., Kim, N., Jumelet, J., Mohebbi, H., Mueller, A., and Chen, H., editors, *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, Miami, Florida, US. Association for Computational Linguistics.

Marks, S., Rager, C., Michaud, E. J., Belinkov, Y., Bau, D., and Mueller, A. (2025). Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*.

Marks, S. and Tegmark, M. (2024). The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*.

McDougall, C., Conmy, A., Kramár, J., Lieberum, T., Rajamanoharan, S., and Nanda, N. (2025). Gemma scope 2 - technical paper. Technical report, Google DeepMind. `https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/gemma-scope-2-helping-the-ai-safety-community-deepen-understanding-of-complex-language-model-behavior/Gemma_Scope_2_Technical_Paper.pdf`.

Mendel, J. (2024). SAE feature geometry is outside the superposition hypothesis. *AI Alignment Forum*. `https://www.alignmentforum.org/posts/MFBTjb2qf3ziWmzz6/sae-feature-geometry-is-outside-the-superposition-hypothesis`.

Nanda, N. (2023). Open source replication & commentary on Anthropic's dictionary learning paper. *AI Alignment Forum*. `https://www.alignmentforum.org/posts/fKuugaxt2XLTkASkk/open-source-replication-and-commentary-on-anthropic-s`.

Nanda, N. and Bloom, J. (2022). TransformerLens. `https://github.com/TransformerLensOrg/TransformerLens`.

Nanda, N., Lee, A., and Wattenberg, M. (2023). Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Olah, C. and Jermyn, A. (2024). Circuits updates - july 2024. *Transformer Circuits Thread*. `https://transformer-circuits.pub/2024/july-update/index.html`.

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.

O'Reilly, P., Seetharaman, P., Su, J., Jin, Z., and Pardo, B. (2025). Code drift: Towards idempotent neural audio codecs. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Hyderabad, India. IEEE.

Otto, S. E., Macchio, G. R., and Rowley, C. W. (2023). Learning nonlinear projections for reduced-order modeling of dynamical systems using constrained autoencoders. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(11):113130.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Park, K., Choe, Y. J., Jiang, Y., and Veitch, V. (2025). The geometry of categorical and hierarchical concepts in large language models. In *The Thirteenth International Conference on Learning Representations*.

Park, K., Choe, Y. J., and Veitch, V. (2024). The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Paulo, G. and Belrose, N. (2026). Sparse autoencoders trained on the same data learn different features. In *The Fourteenth International Conference on Learning Representations*.

Paulo, G., Mallen, A., Juang, C., and Belrose, N. (2025). Automatically interpreting millions of features in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 48393–48421. PMLR.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. Technical Report 8, OpenAI. `https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1).

Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramar, J., Shah, R., and Nanda, N. (2024a). Improving sparse decomposition of language model activations with gated sparse autoencoders. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. `https://arxiv.org/abs/2407.14435`.

Shocher, A., Dravid, A., Gandelsman, Y., Mosseri, I., Rubinstein, M., and Efros, A. A. (2024). Idempotent generative network. In *The Twelfth International Conference on Learning Representations*.

Smith, L. (2024). The 'strong' feature hypothesis could be wrong. *LessWrong*. `https://www.lesswrong.com/posts/tojtPCCRpKLSHBdpn/the-strong-feature-hypothesis-could-be-wrong`.

Taggart, G. (2024). ProLU: A nonlinearity for sparse autoencoders. *LessWrong*. `https://www.lesswrong.com/posts/HEpufTdakGTTKgoYF/prolu-a-pareto-improvement-for-sparse-autoencoders`.

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., Tamkin, A., Durmus, E., Hume, T., Mosconi, F., 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*. `https://transformer-circuits.pub/2024/scaling-monosemanticity/#related-work`.

Till, D. and Chanin, D. (2024). Broken latents: Studying SAEs and feature co-occurrence in toy models. LessWrong. `https://www.lesswrong.com/posts/XHpta8X85TzugNNn2/broken-latents-studying-saes-and-feature-co-occurrence-in`.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wu, Z., Arora, A., Geiger, A., Wang, Z., Huang, J., Jurafsky, D., Manning, C. D., and Potts, C. (2025). Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Forty-second International Conference on Machine Learning*.

Xu, T., Zhu, Z., He, D., Li, Y., Guo, L., Wang, Y., Wang, Z., Qin, H., Wang, Y., Liu, J., and Zhang, Y.-Q. (2024). Idempotence and perceptual image compression. In *The Twelfth International Conference on Learning Representations*.

Zhang, F. and Nanda, N. (2024). Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

Zhang, W. and Hu, B. Q. (2022). The idempotency of convolution operations on fuzzy truth values. *IEEE Transactions on Fuzzy Systems*, 30(4):990–998.

A Motivation: Idempotency as a Consistency Criterion for Sparse Autoencoders

Consider a ground-truth reference SAE, $\tilde{G} \circ \tilde{F}$, that has learned a faithful latent decomposition in the sense of the LRH for an input distribution $\tilde{X}$, i.e. there exists a set of approximately orthogonal latent directions $\{\tilde{l}_i\}$, stored in the decoder $\tilde{G}$, which span the relevant subspace as an overcomplete base and correspond to the ground-truth of the input. Consider an arbitrary in-distribution input direction $\tilde{x} \in \tilde{X}$, then its representation with respect to these ground-truth latents is:

$$\tilde{x} = \tilde{b}_d + \sum_i \alpha_i \tilde{l}_i \ , \tag{10}$$

where $\tilde{b}_d$ is the decoder bias of this ideal SAE and α_i being linear coefficients under this sparse representation, also recoverable by the ground-truth encoder $\tilde{F}$. Assuming this SAE with latents corresponding to the ground truth exists, it is almost certain that we will not be able to recover it perfectly. Instead, training gives us an approximation to the true SAE: $\hat{G} \circ \hat{F}$ with latents $\{\hat{l}_i\}$. Given the input $\tilde{x} \in \tilde{X}$, the reconstruction under this practically obtainable SAE can be written as

$$\hat{x} = (\hat{G} \circ \hat{F})(\tilde{x}) = \hat{b}_d + \sum_i \gamma_i \hat{l}_i \ , \tag{11}$$

with $\hat{b}_d$ representing the learned decoder bias, and γ_i being the coordinates uncovered by the encoder $\hat{F}$. Thus we know $\hat{F}(\tilde{x}) = [\gamma_1 \dots \gamma_m]$, with m being the dimensionality of the SAE. Given the case that the ideal SAE is faithfully recovered, i.e., $\hat{F} \approx \tilde{F}$ and $\hat{G} \approx \tilde{G}$, one can arguably expect $\alpha_i \approx \gamma_i$. Consequently, re-encoding the reconstruction should approximately recover the same latent code, $\hat{F}(\tilde{x}) \approx \hat{F}(\hat{x})$. Equivalently, this requires $(\hat{F} \circ \hat{G} \circ \hat{F})(\tilde{x}) \approx \hat{F}(\tilde{x})$, which corresponds to the idempotency requirement defined in Equation (5). In this view, LRH-aligned SAEs that recover true latents and an encoder that projects the input onto the coordinates defined by these latents admit to idempotency and thus it defines a criterion for self consistency of the dictionary.
When this criterion is violated there are two potential causes, both corresponding to different practical implications:

- $\hat{\boldsymbol{F}} \neq \tilde{\boldsymbol{F}}$ Even under the strong assumption that the trained decoder exactly recovers the ideal decoder, $\hat{G} = \tilde{G}$, overcompleteness of the dictionary can make the latent representation not uniquely identifiable. The encoder may select under a typical input $\tilde{x}$ an alternative set of coordinates instead of the ground-truth coordinates. This is problematic for interpretability experiments that directly assign semantic meaning to active latent features (Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024) as it upper bounds the achievable recall.

- $\hat{G} \neq \tilde{G}$ Conversely, even under the assumption that the latent code always corresponds to the ground truth, thus $\hat{F} = \tilde{F}$, the high dimensionality of the dictionary makes it in principle possible for the SAE to learn a different set of decoder directions whose linear combinations nevertheless achieve low reconstruction loss. This is problematic for work that relies on the causal validity of individual latents, for example for causal tracing, or in steering literature (Zhang and Nanda, 2024; Marks et al., 2025; Templeton et al., 2024; Cho and Hockenmaier, 2025).

In particular, both error types[22] violate the idempotency requirement in Equation (5). This makes idempotency a self-consistency criterion which, when included in the evaluation of trained SAEs, may help constrain potential failure modes.

Note, however, that the criterion is necessary rather than sufficient. It is possible for a SAE to be idempotent, but not recover the ground-truth dictionary $\tilde{G}$. Consider for example the case of feature splitting (Templeton et al., 2024), where a SAE learns to split one underlying feature into two in order to minimize sparsity. The SAE could be idempotent in that it always recovers this split representation under re-encoding and yet still fail to represent the ground truth features. Relatedly, we make no claim that the learned dictionary is unique: SAEs trained on the same data with different random seeds can recover different sets of latents (Paulo and Belrose, 2026). Idempotency is defined with respect to a given trained dictionary. Even then, overcompleteness means that several sparse decompositions may represent the same input. We only require that, once the dictionary is fixed, the encoder's choice among these decompositions depends on the dictionary-representable content of the input rather than on the residual ϵ_x. Hence, violations of idempotency reveal failures of internal self-consistency, whereas it is not expected to rule out all potential issues with SAE dictionary learning.

B Additional Idempotency Results

Figures 6 to 18 report idempotency metrics for all model and SAE combinations. Each figure follows the same format as Figures 1 and 2. The top row shows base re-encoding results and the bottom row shows results under norm-controlled re-encoding. Lines show the median over tokens and shaded bands the interquartile range, both averaged over layers. The ℓ_0 norm is not normalized by SAE width in these plots. In general, narrower SAEs exhibit greater stability under repeated re-encoding and therefore appear closer to idempotent behavior. The relationship with sparsity is less uniform. For Gemma Scope 2 SAEs, lower-ℓ_0 models tend to be more stable, whereas for SAEBench SAEs, greater stability is observed at higher ℓ_0 values.

[22]The difficulty of isolating whether issues stem from the encoder or the decoder makes tied SAEs an appealing alternative. However, while tied SAEs have also been reported to mitigate feature absorption (Chanin et al., 2024a) in some settings, they still exhibit distinct pathological behaviors, such as broken latents (Till and Chanin, 2024), limiting their utility as a straightforward solution.

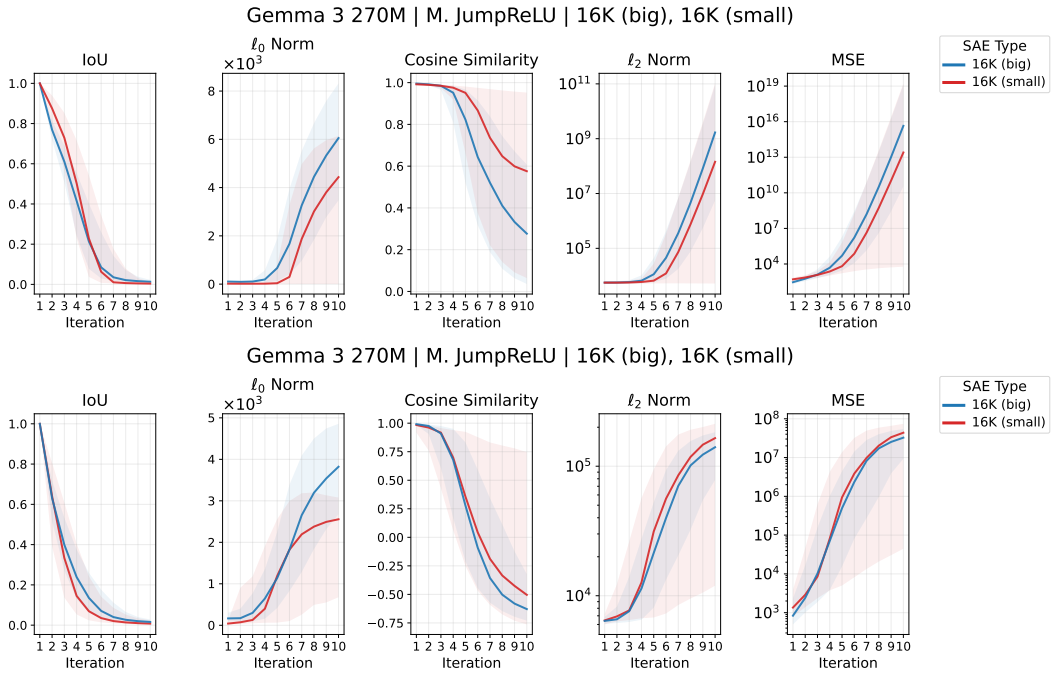


Figure 6: Idempotency metrics for Gemma 3 270M with 16K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

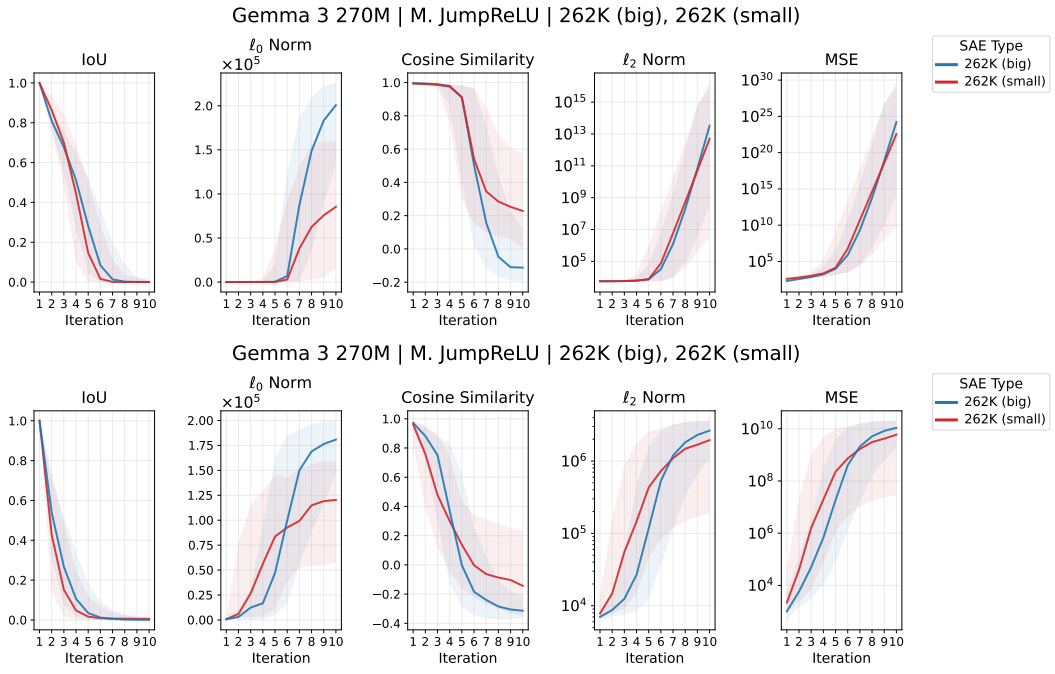


Figure 7: Idempotency metrics for Gemma 3 270M with 262K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

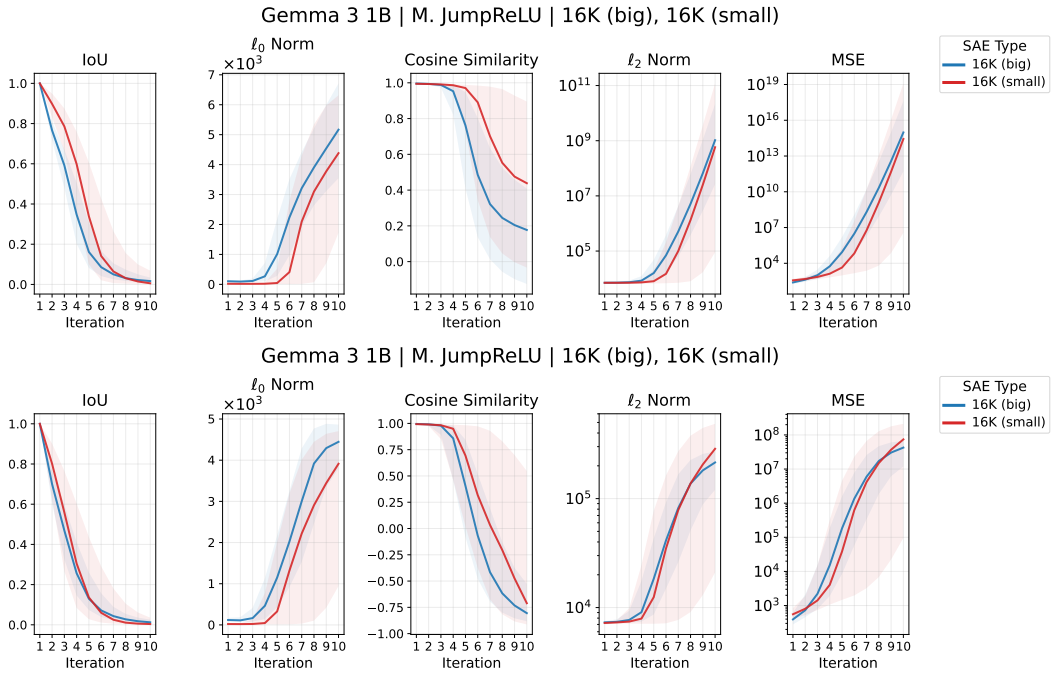


Figure 8: Idempotency metrics for Gemma 3 1B with 16K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

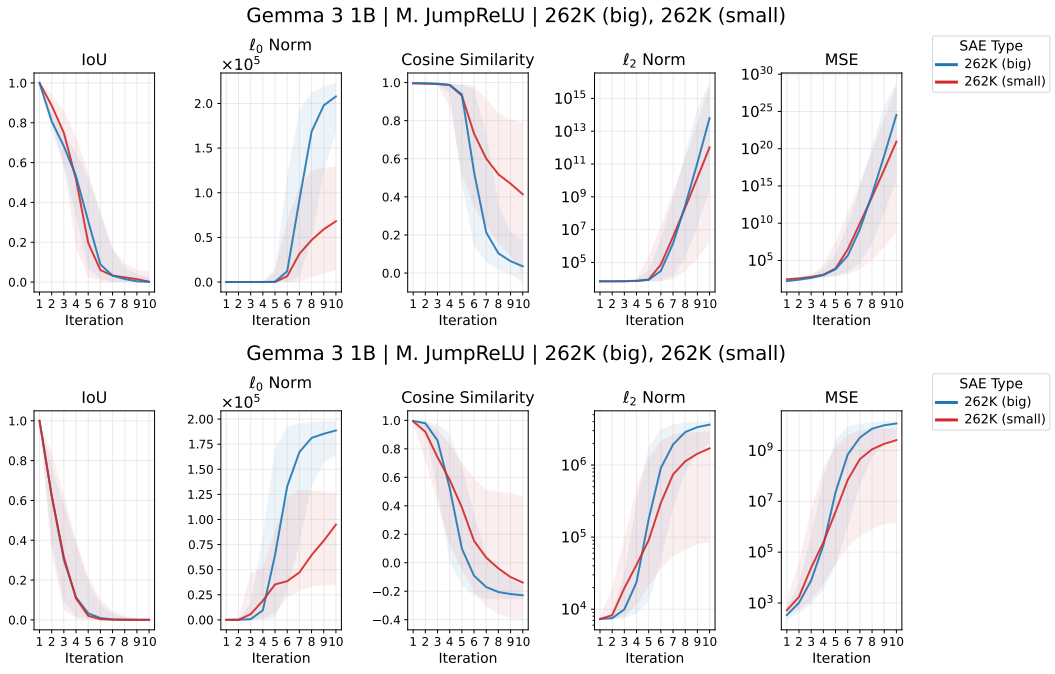


Figure 9: Idempotency metrics for Gemma 3 1B with 262K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

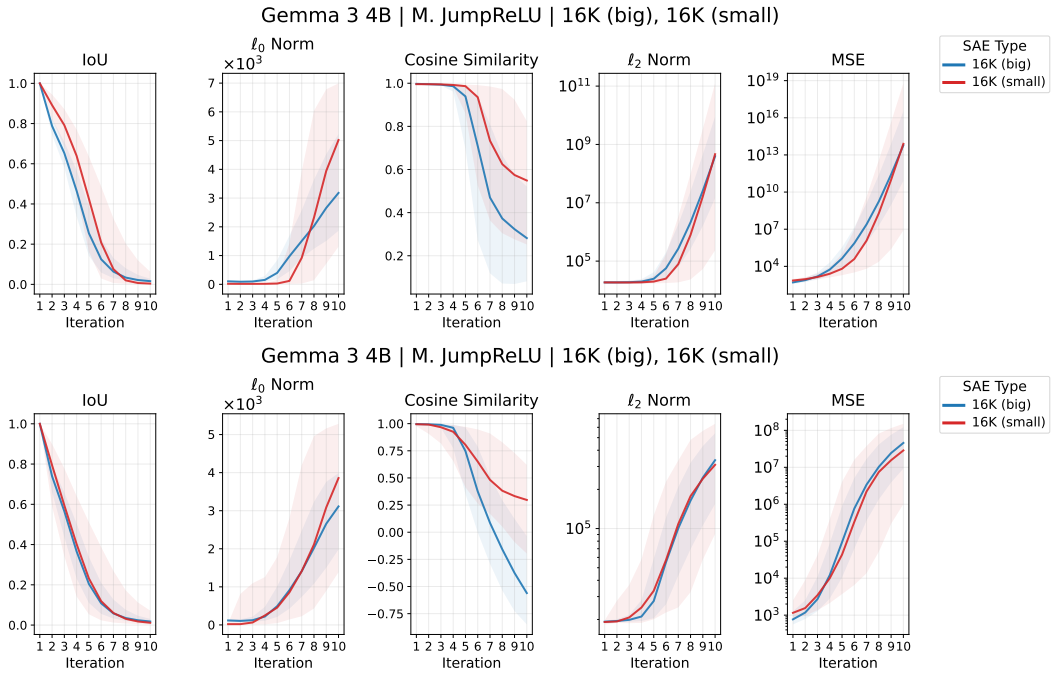


Figure 10: Idempotency metrics for Gemma 3 4B with 16K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

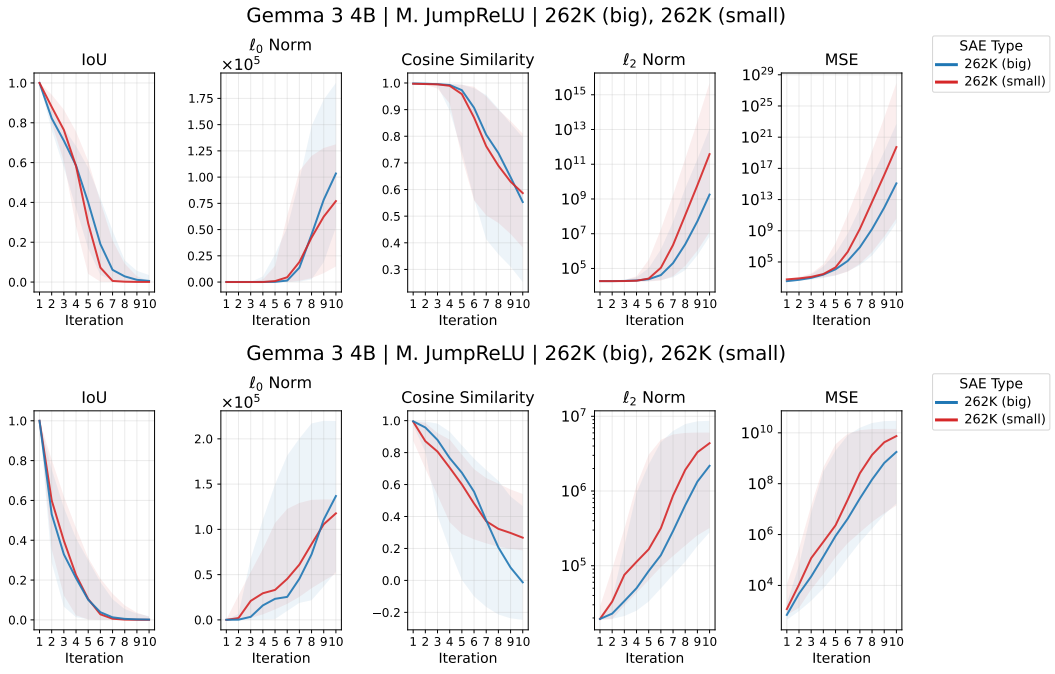


Figure 11: Idempotency metrics for Gemma 3 4B with 262K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

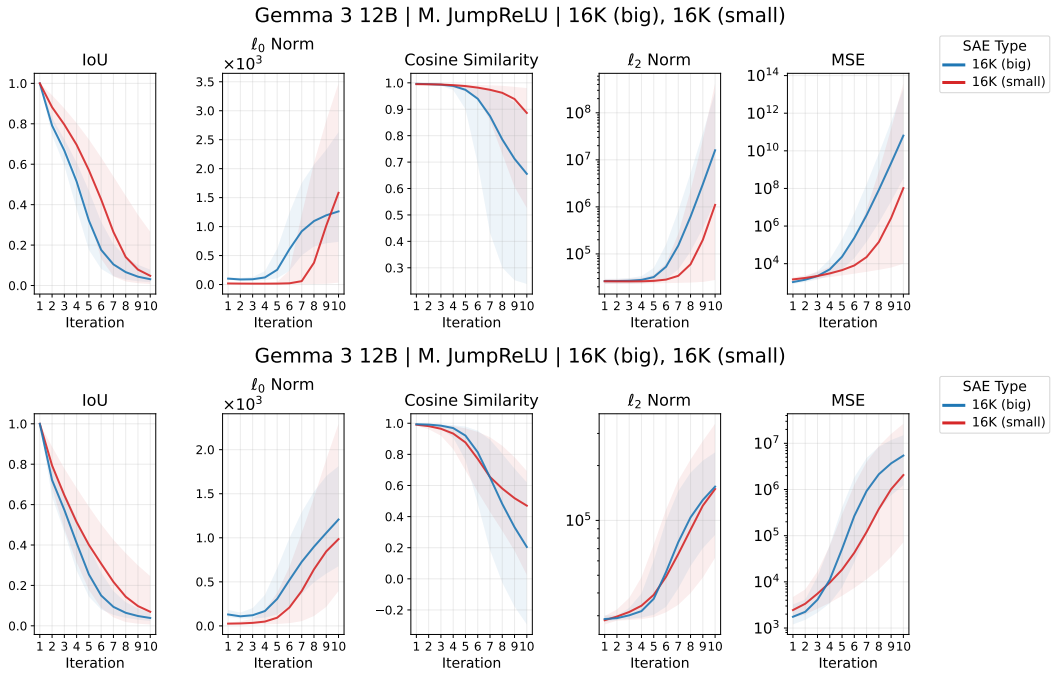


Figure 12: Idempotency metrics for Gemma 3 12B with 16K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

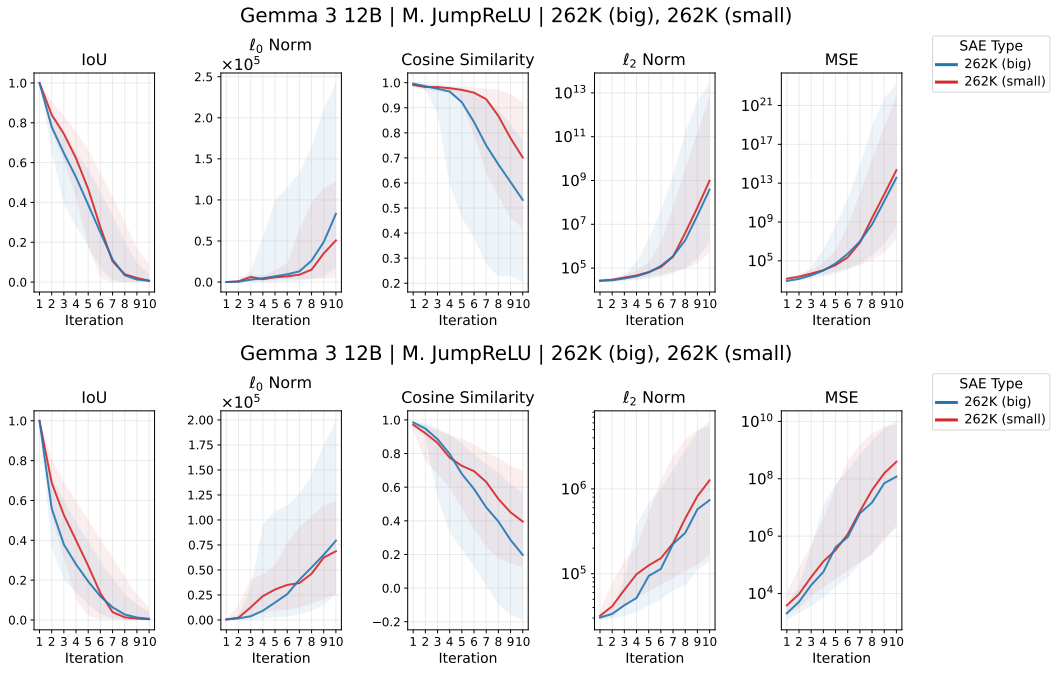


Figure 13: Idempotency metrics for Gemma 3 12B with 262K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

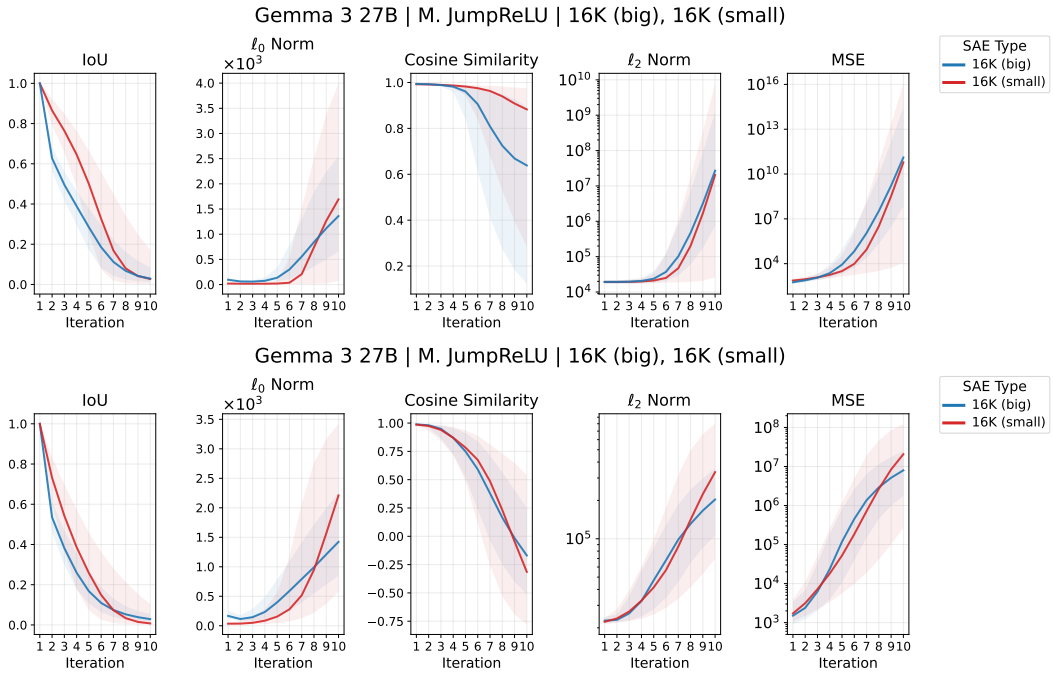


Figure 14: Idempotency metrics for Gemma 3 27B with 16K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

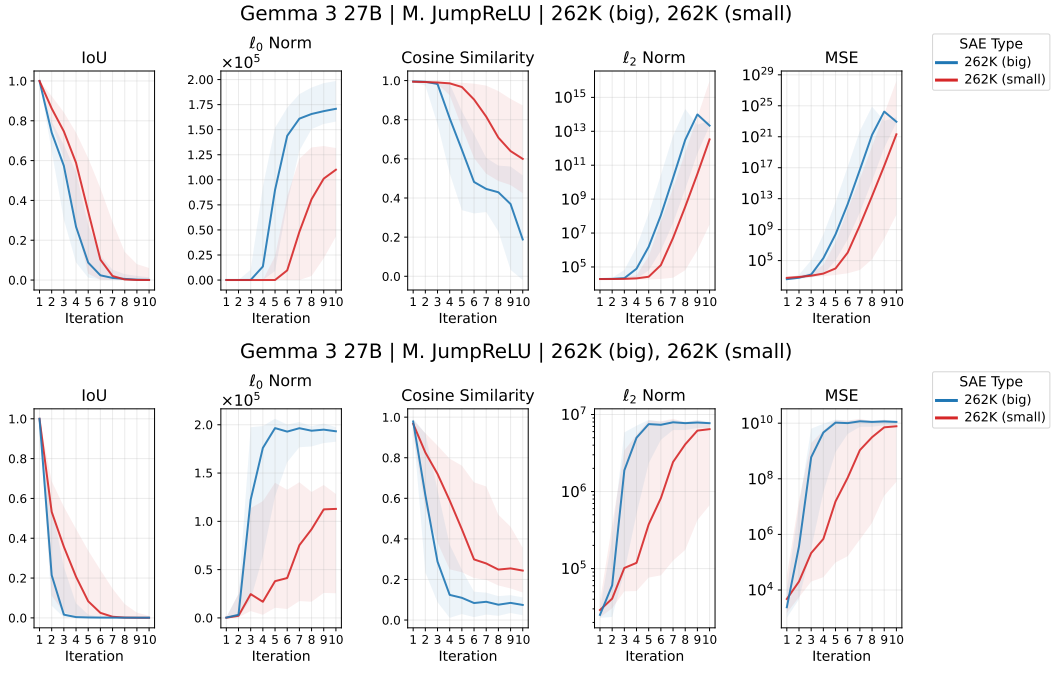


Figure 15: Idempotency metrics for Gemma 3 27B with 262K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

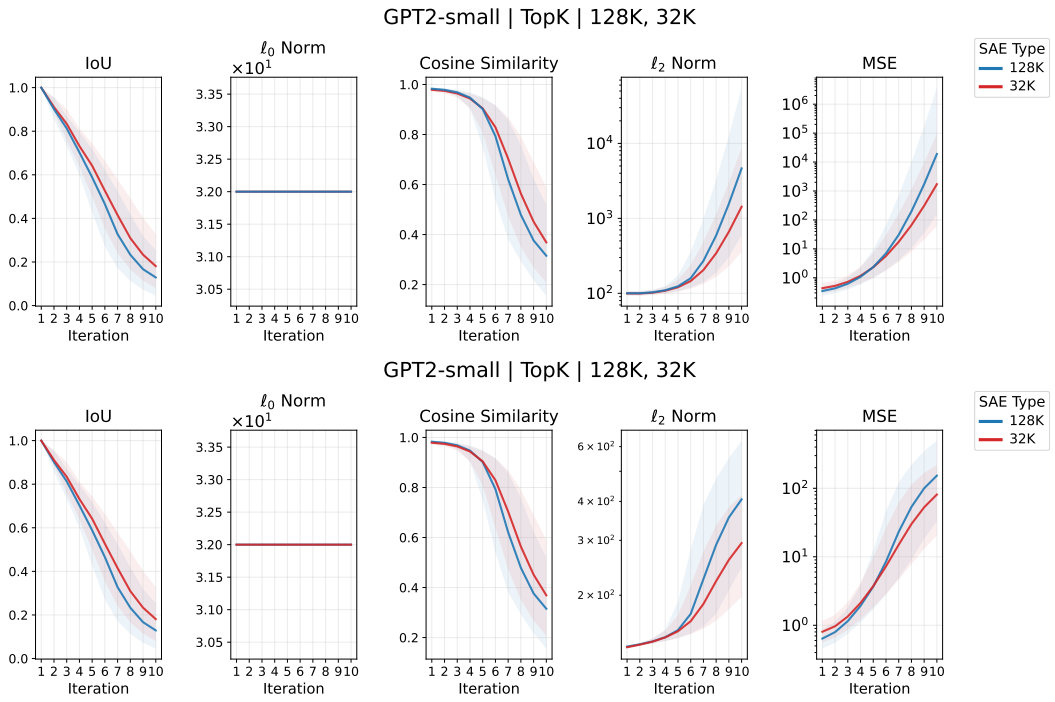


Figure 16: Idempotency metrics for GPT2-small with 128K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

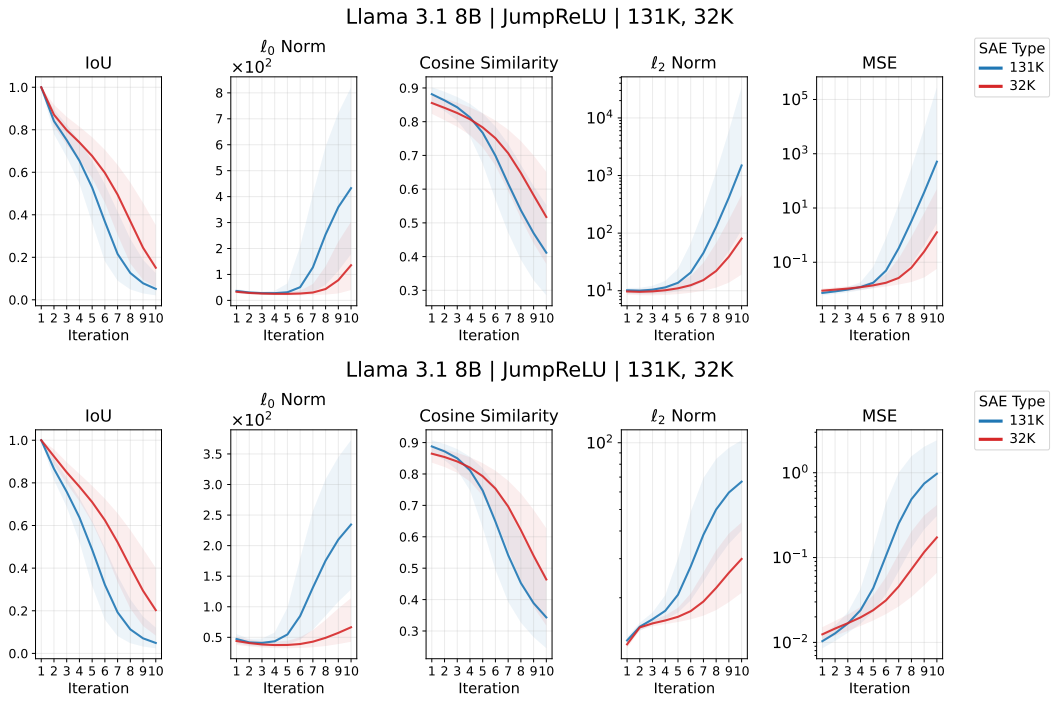


Figure 17: Idempotency metrics for Llama 3.1 8B with 131K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

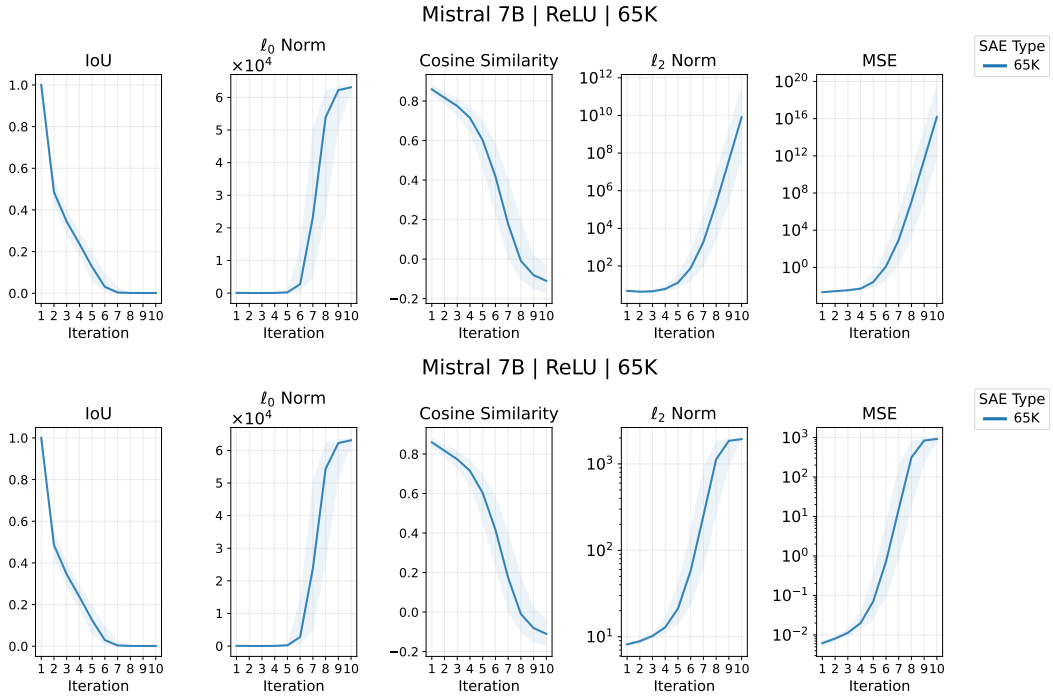


Figure 18: Idempotency metrics for Mistral 7B with 65K SAE. Top: base re-encoding. Bottom: norm-controlled re-encoding.

B.1 Layerwise Breakdown

In this section we break down the results for Experiment 1 and Experiment 3 by layer. Figures 19 to 21 show the first, middle, and last layer of each SAE from Figure 1. Here, each line shows the median over tokens at a single layer and the shaded band its interquartile range, without averaging over layers. We observe that these trends hold consistently across all layers of the model.

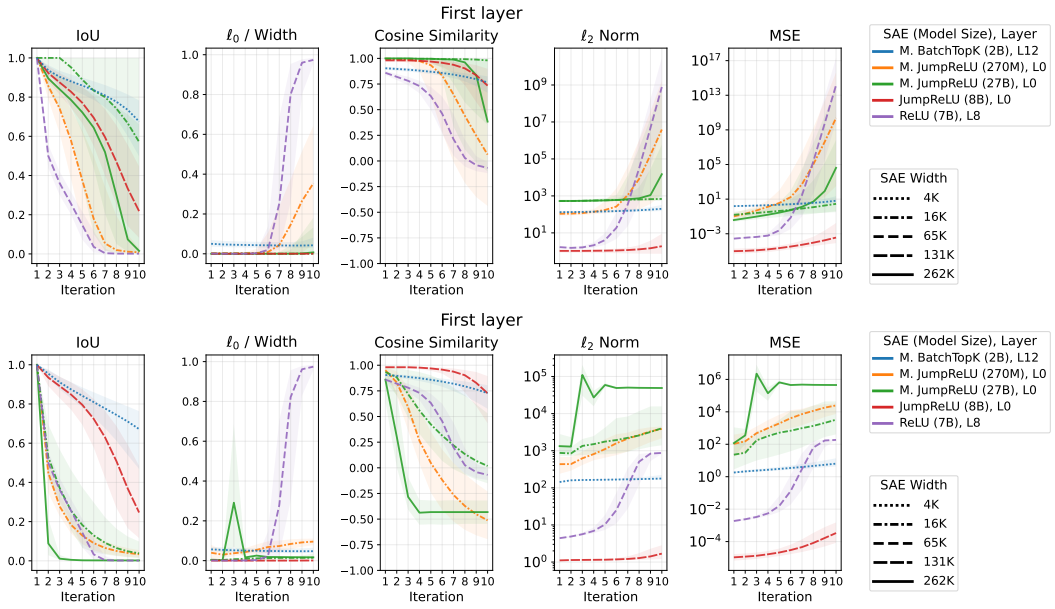


Figure 19: Idempotency metrics at the first layer of each SAE in Fig. 1. Top: base re-encoding. Bottom: norm-controlled re-encoding. Lines show the median over tokens, shaded bands the interquartile range.

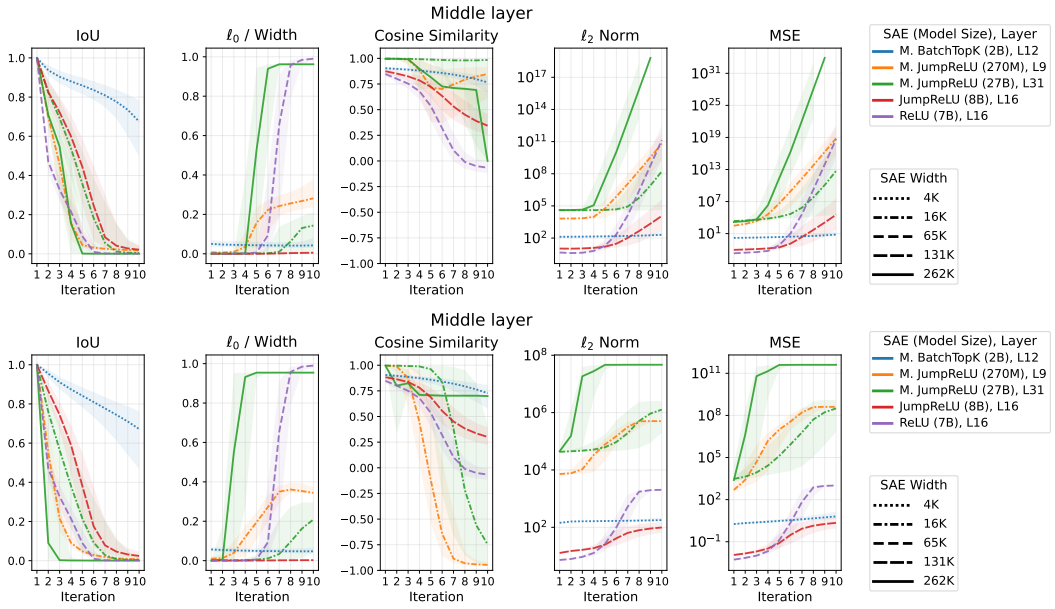


Figure 20: Idempotency metrics at the middle layer of each SAE in Fig. 1. Top: base re-encoding. Bottom: norm-controlled re-encoding. Lines show the median over tokens, shaded bands the interquartile range.

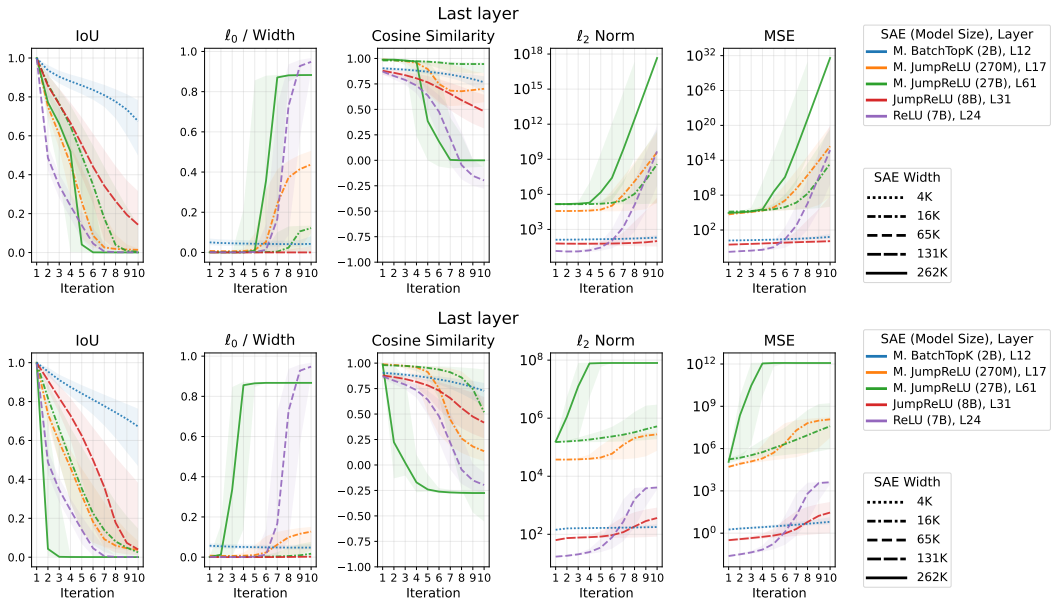


Figure 21: Idempotency metrics at the last layer of each SAE in Fig. 1. Top: base re-encoding. Bottom: norm-controlled re-encoding. Lines show the median over tokens, shaded bands the interquartile range.

B.2 Threshold Flip Control

To address concerns that the main text's results are an artifact of near-threshold activations we ran two variants of Experiment 1 on the same representative set of SAEs used in the main text.

First, we partitioned latents by their behavior across consecutive iterations into three populations—disappearing (active at iteration i, inactive at $i+1$), appearing (inactive at i, active at $i+1$), and remaining (active at both). We then compared how close each population's activations sit to the activation threshold. Representative results for Gemma 3 and Gemma 2 are in Tables 3 and 4.

At the median, the appearing and disappearing populations do sit significantly closer to the threshold than the remaining population. However, the 25th and 75th percentiles reveal substantial overlap between the three populations, so proximity to the threshold does not cleanly separate stable from unstable latents. Furthermore, if we fix the remaining population median at iteration one as the baseline for robustly-above-threshold activation magnitudes, we observe that in many cases the appearing and disappearing population median at later iterations (usually between 3 and 5) exceeds this value. At that point the effects can no longer be explained by thresholding sensitivity. More importantly, there is no correspondence between threshold proximity and idempotency across models. Matryoshka BatchTopK, the best performing model in our experiments, exhibits some of the closest-to-threshold latents of any SAE we tested (cf. Table 4). If near-threshold behavior were the driver of our instability results, this model should have been among the worst, not the best.

Because some latents nonetheless sit very close to the threshold, we ran a second, stronger control to rule out thresholding noise directly. We repeated the experiment, leaving the sequence of x_t unchanged, but restrict the metrics to latents with magnitude above the per-token median. This directly disregards any edge-case latents and focuses only on the unambiguous cases. The results are in Figure 22. Crucially, the trends remain unchanged. Even when we look exclusively at above-median latents, IoU collapses, L0 grows substantially, cosine similarity dwindles, and L2 and MSE grow by orders of magnitude.

We have further included the same above-median control for the GPT2 TopK SAEs. This type of SAEs operates entirely without a threshold and still displays consistent results.

Taken together, these results show that the instability we report is not an artifact of latents fluctuating around their threshold. It reflects a genuine reorganization of the representation that extends to its high-magnitude core, and it does not track threshold proximity across models.

Table 3: Activation magnitude minus firing threshold, per selected layer and iteration for JumpReLU Gemma 3 270M 16K (big) SAE.

Layer	Iteration	Remain. p50	Appear. p50	Disap. p50	Remain. p25	Appear. p75	Disap. p75
0	1	1.337	0.234	0.246	0.563	0.487	0.502
	2	1.500	0.466	0.266	0.673	1.103	0.578
	3	1.713	1.364	0.359	0.697	3.913	0.841
	4	2.362	3.313	0.804	0.825	10.425	2.463
	5	5.661	8.044	2.807	1.726	27.714	11.554
9	1	22.204	7.113	5.241	10.259	14.125	10.116
	2	27.975	19.274	5.551	11.835	40.792	10.902
	3	33.923	63.524	8.770	12.955	147.330	19.030
	4	98.282	148.798	45.345	34.843	365.555	107.969
	5	453.679	315.304	127.668	129.910	1010.059	350.454
17	1	265.101	68.105	61.644	125.125	137.857	118.363
	2	334.065	135.806	56.554	162.007	304.392	107.172
	3	401.121	375.812	59.365	167.895	904.687	117.822
	4	591.657	964.461	126.390	218.946	2453.508	377.048
	5	1636.029	2127.125	546.826	530.560	5878.570	1557.781

Table 4: Activation magnitude minus firing threshold, per selected layer and iteration for Matryoshka BatchTopK Gemma 2 2B 4K (target ℓ_0 of 80) SAE.

Layer	Iteration	Remain. p50	Appear. p50	Disap. p50	Remain. p25	Appear. p75	Disap. p75
12	1	2.625	0.375	0.312	1.062	1.141	0.562
	2	3.250	0.094	0.250	1.531	0.281	0.469
	3	4.000	0.219	0.219	2.000	0.438	0.438
	4	4.750	0.312	0.219	2.438	0.625	0.406
	5	5.625	0.438	0.219	2.875	0.875	0.438

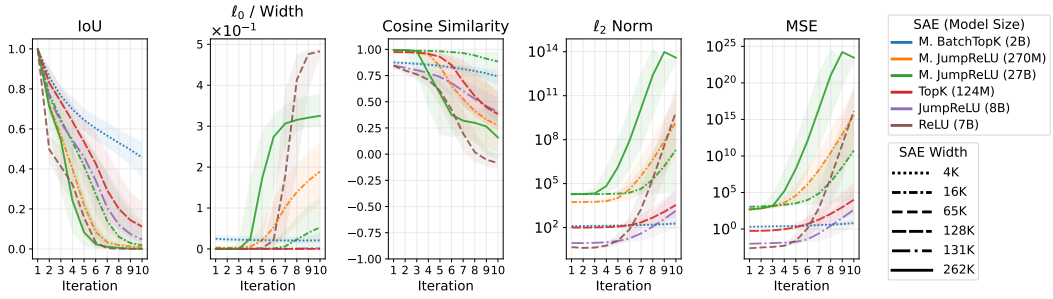


Figure 22: Idempotency metrics under base re-encoding for the SAEs in Fig. 1, restricted at each iteration to latents whose magnitude exceeds the per-token median. Lines show the median over tokens, shaded bands the interquartile range; both are averaged over layers.

C Full Input Scaling Results

This section reports the scaling experiments corresponding to Section 4.3 for all investigated model–SAE combinations. Tables 5–7 present the results for the scaling coefficients $c = 1.1$, $c = 1.2$, and $c = 1.3$, where c is defined according to Equation (7). As in Table 2, each entry reports the median [25th, 75th percentile] over tokens, averaged over layers, and the italic ℓ_0 entries of the Gemma 2 2B SAEs are relative to the median ℓ_0 of the unscaled input.

Table 5: Full Input Scaling Results ($c = 1.1$)

Model	SAE Type	SAE Size	$\|F(x)\|_0$	$\|F(\lambda(x))\|_0$	Identical Tokens	$\|\epsilon\|_2/\sigma$
Gemma 2 2B	JumpReLU	4K	*1.00* [*0.76*, *1.37*]	*1.21* [*0.93*, *1.61*]	92 [90, 100]%	0.09 [0.08, 0.10]
	JumpReLU id.	4K	*1.00* [*0.97*, *1.03*]	*1.02* [*0.98*, *1.05*]	97 [90, 100]%	0.12 [0.11, 0.14]
	M. BatchTopK	4K	*1.00* [*0.74*, *1.37*]	*1.18* [*0.88*, *1.60*]	92 [87, 98]%	0.10 [0.09, 0.11]
	M. BatchTopK id.	4K	*1.00* [*0.73*, *1.37*]	*1.15* [*0.86*, *1.58*]	90 [83, 97]%	0.14 [0.13, 0.15]
	JumpReLU	16K	*1.00* [*0.77*, *1.33*]	*1.29* [*1.00*, *1.66*]	90 [90, 100]%	0.09 [0.08, 0.10]
	JumpReLU id.	16K	*1.00* [*0.98*, *1.04*]	*1.02* [*0.99*, *1.06*]	98 [90, 100]%	0.12 [0.11, 0.14]
	M. BatchTopK	16K	*1.00* [*0.76*, *1.36*]	*1.22* [*0.93*, *1.62*]	92 [88, 100]%	0.09 [0.08, 0.10]
	M. BatchTopK id.	16K	*1.00* [*0.75*, *1.36*]	*1.20* [*0.90*, *1.59*]	92 [90, 100]%	0.10 [0.09, 0.11]
	JumpReLU	65K	*1.00* [*0.79*, *1.27*]	*1.34* [*1.08*, *1.65*]	92 [90, 100]%	0.09 [0.08, 0.10]
	JumpReLU id.	65K	*1.00* [*0.97*, *1.03*]	*1.02* [*0.99*, *1.08*]	97 [90, 100]%	0.12 [0.11, 0.14]
	M. BatchTopK	65K	*1.00* [*0.76*, *1.33*]	*1.26* [*0.99*, *1.63*]	92 [90, 100]%	0.09 [0.08, 0.10]
	M. BatchTopK id.	65K	*1.00* [*0.76*, *1.35*]	*1.25* [*0.96*, *1.65*]	90 [85, 98]%	0.10 [0.10, 0.11]
Gemma 3 270M	M. JumpReLU	16K (small)	19 [14, 26]	28 [21, 37]	98 [97, 100]%	0.08 [0.07, 0.10]
	M. JumpReLU	16K (big)	106 [86, 130]	135 [112, 161]	98 [97, 100]%	0.08 [0.07, 0.10]
	M. JumpReLU	262K (small)	19 [15, 25]	28 [22, 37]	98 [97, 100]%	0.08 [0.07, 0.10]
	M. JumpReLU	262K (big)	104 [83, 132]	122 [100, 150]	98 [97, 100]%	0.09 [0.07, 0.10]
Gemma 3 1B	M. JumpReLU	16K (small)	19 [15, 25]	26 [21, 34]	99 [93, 100]%	0.09 [0.07, 0.10]
	M. JumpReLU	16K (big)	104 [84, 131]	133 [108, 165]	98 [93, 100]%	0.09 [0.08, 0.10]
	M. JumpReLU	262K (small)	19 [15, 23]	29 [22, 39]	99 [93, 100]%	0.09 [0.07, 0.10]
	M. JumpReLU	262K (big)	104 [85, 130]	110 [90, 137]	99 [93, 100]%	0.09 [0.08, 0.11]
Gemma 3 4B	M. JumpReLU	16K (small)	18 [14, 24]	24 [19, 31]	100 [99, 100]%	0.05 [0.04, 0.07]
	M. JumpReLU	16K (big)	104 [85, 130]	126 [104, 155]	100 [99, 100]%	0.05 [0.04, 0.07]
	M. JumpReLU	262K (small)	18 [14, 23]	43 [21, 149]	100 [99, 100]%	0.05 [0.04, 0.07]
	M. JumpReLU	262K (big)	104 [86, 128]	97 [81, 119]	100 [99, 100]%	0.06 [0.05, 0.07]
Gemma 3 12B	M. JumpReLU	16K (small)	18 [14, 24]	23 [18, 30]	99 [93, 100]%	0.04 [0.03, 0.05]
	M. JumpReLU	16K (big)	102 [81, 128]	122 [99, 153]	99 [93, 100]%	0.04 [0.03, 0.05]
	M. JumpReLU	262K (small)	27 [15, 83]	27 [19, 44]	99 [93, 100]%	0.04 [0.03, 0.05]
	M. JumpReLU	262K (big)	113 [88, 168]	99 [81, 123]	99 [90, 100]%	0.04 [0.03, 0.05]
Gemma 3 27B	M. JumpReLU	16K (small)	20 [16, 25]	25 [21, 32]	100 [99, 100]%	0.03 [0.02, 0.03]
	M. JumpReLU	16K (big)	95 [77, 124]	114 [93, 147]	100 [99, 100]%	0.03 [0.02, 0.03]
	M. JumpReLU	262K (small)	21 [18, 26]	61 [27, 134]	100 [98, 100]%	0.03 [0.03, 0.04]
	M. JumpReLU	262K (big)	117 [102, 135]	134 [113, 171]	100 [98, 100]%	0.04 [0.03, 0.05]
GPT2-small	TopK	32K	32 [32, 32]	32 [32, 32]	100 [93, 100]%	0.10 [0.09, 0.11]
	TopK	128K	32 [32, 32]	32 [32, 32]	100 [93, 100]%	0.10 [0.09, 0.11]
Llama 3.1 8B	JumpReLU	32K	33 [27, 40]	39 [31, 47]	100 [100, 100]%	0.07 [0.06, 0.07]
	JumpReLU	131K	36 [29, 43]	43 [35, 51]	100 [100, 100]%	0.07 [0.06, 0.07]
Mistral 7B	ReLU	65K	86 [75, 96]	34981 [32623, 37632]	67 [47, 77]%	0.07 [0.07, 0.07]

Table 6: Full Input Scaling Results ($c = 1.2$)

Model	SAE Type	SAE Size	$\|F(x)\|_0$	$\|F(\lambda(x))\|_0$	Identical Tokens	$\|\epsilon\|_2/\sigma$
Gemma 2 2B	JumpReLU	4K	*1.00* [*0.76*, *1.37*]	*1.43* [*1.12*, *1.88*]	90 [82, 100]%	0.18 [0.16, 0.20]
	JumpReLU id.	4K	*1.00* [*0.97*, *1.03*]	*1.03* [*0.99*, *1.07*]	90 [87, 100]%	0.24 [0.21, 0.27]
	M. BatchTopK	4K	*1.00* [*0.74*, *1.37*]	*1.37* [*1.03*, *1.84*]	87 [78, 95]%	0.20 [0.18, 0.22]
	M. BatchTopK id.	4K	*1.00* [*0.73*, *1.37*]	*1.32* [*0.99*, *1.78*]	83 [73, 92]%	0.28 [0.26, 0.30]
	JumpReLU	16K	*1.00* [*0.77*, *1.33*]	*1.61* [*1.29*, *2.03*]	90 [80, 98]%	0.18 [0.16, 0.20]
	JumpReLU id.	16K	*1.00* [*0.98*, *1.04*]	*1.05* [*1.01*, *1.09*]	92 [88, 98]%	0.24 [0.22, 0.27]
	M. BatchTopK	16K	*1.00* [*0.76*, *1.36*]	*1.47* [*1.13*, *1.92*]	88 [80, 97]%	0.18 [0.16, 0.21]
	M. BatchTopK id.	16K	*1.00* [*0.75*, *1.36*]	*1.43* [*1.09*, *1.88*]	90 [82, 95]%	0.20 [0.18, 0.22]
	JumpReLU	65K	*1.00* [*0.79*, *1.27*]	*1.77* [*1.48*, *2.11*]	90 [82, 98]%	0.18 [0.16, 0.20]
	JumpReLU id.	65K	*1.00* [*0.97*, *1.03*]	*1.07* [*1.02*, *1.12*]	92 [87, 100]%	0.24 [0.22, 0.27]
	M. BatchTopK	65K	*1.00* [*0.76*, *1.33*]	*1.61* [*1.29*, *2.03*]	90 [82, 98]%	0.18 [0.16, 0.21]
	M. BatchTopK id.	65K	*1.00* [*0.76*, *1.35*]	*1.62* [*1.24*, *2.14*]	85 [73, 90]%	0.21 [0.19, 0.23]
Gemma 3 270M	M. JumpReLU	16K (small)	19 [14, 26]	40 [32, 52]	98 [96, 98]%	0.17 [0.14, 0.19]
	M. JumpReLU	16K (big)	106 [86, 130]	168 [142, 198]	97 [96, 98]%	0.17 [0.14, 0.20]
	M. JumpReLU	262K (small)	19 [15, 25]	50 [35, 155]	97 [96, 98]%	0.17 [0.14, 0.19]
	M. JumpReLU	262K (big)	104 [83, 132]	145 [121, 174]	97 [96, 98]%	0.17 [0.15, 0.20]
Gemma 3 1B	M. JumpReLU	16K (small)	19 [15, 25]	36 [29, 46]	93 [88, 100]%	0.17 [0.15, 0.20]
	M. JumpReLU	16K (big)	104 [84, 131]	166 [138, 204]	94 [88, 100]%	0.18 [0.15, 0.21]
	M. JumpReLU	262K (small)	19 [15, 23]	59 [39, 100]	93 [88, 100]%	0.17 [0.15, 0.20]
	M. JumpReLU	262K (big)	104 [85, 130]	123 [101, 152]	93 [89, 100]%	0.18 [0.16, 0.21]
Gemma 3 4B	M. JumpReLU	16K (small)	18 [14, 24]	31 [25, 40]	99 [99, 100]%	0.10 [0.08, 0.13]
	M. JumpReLU	16K (big)	104 [85, 130]	150 [126, 183]	99 [99, 100]%	0.10 [0.08, 0.13]
	M. JumpReLU	262K (small)	18 [14, 23]	510 [193, 890]	99 [99, 100]%	0.10 [0.08, 0.13]
	M. JumpReLU	262K (big)	104 [86, 128]	103 [87, 123]	99 [99, 100]%	0.12 [0.09, 0.15]
Gemma 3 12B	M. JumpReLU	16K (small)	18 [14, 24]	30 [24, 39]	97 [89, 99]%	0.07 [0.06, 0.09]
	M. JumpReLU	16K (big)	102 [81, 128]	144 [118, 179]	96 [89, 99]%	0.07 [0.06, 0.09]
	M. JumpReLU	262K (small)	27 [15, 83]	44 [28, 89]	98 [88, 99]%	0.08 [0.06, 0.10]
	M. JumpReLU	262K (big)	113 [88, 168]	98 [81, 122]	97 [88, 99]%	0.08 [0.06, 0.11]
Gemma 3 27B	M. JumpReLU	16K (small)	20 [16, 25]	32 [26, 40]	100 [98, 100]%	0.05 [0.04, 0.06]
	M. JumpReLU	16K (big)	95 [77, 124]	133 [109, 173]	100 [98, 100]%	0.05 [0.04, 0.06]
	M. JumpReLU	262K (small)	21 [18, 26]	266 [103, 455]	99 [97, 100]%	0.07 [0.06, 0.08]
	M. JumpReLU	262K (big)	117 [102, 135]	1658 [380, 5513]	99 [97, 99]%	0.08 [0.07, 0.09]
GPT2-small	TopK	32K	32 [32, 32]	32 [32, 32]	95 [92, 100]%	0.19 [0.17, 0.21]
	TopK	128K	32 [32, 32]	32 [32, 32]	96 [92, 100]%	0.19 [0.17, 0.21]
Llama 3.1 8B	JumpReLU	32K	33 [27, 40]	45 [36, 54]	100 [98, 100]%	0.14 [0.12, 0.15]
	JumpReLU	131K	36 [29, 43]	50 [40, 60]	100 [99, 100]%	0.14 [0.12, 0.15]
Mistral 7B	ReLU	65K	86 [75, 96]	45014 [44340, 46793]	37 [23, 53]%	0.14 [0.14, 0.15]

Table 7: Full Input Scaling Results ($c = 1.3$)

Model	SAE Type	SAE Size	$\|F(x)\|_0$	$\|F(\lambda(x))\|_0$	Identical Tokens	$\|\epsilon\|_2/\sigma$
Gemma 2 2B	JumpReLU	4K	$1.00\,[0.76, 1.37]$	$1.69\,[1.34, 2.17]$	90 [80, 90]%	0.26 [0.24, 0.30]
	JumpReLU id.	4K	$1.00\,[0.97, 1.03]$	$1.05\,[1.01, 1.09]$	90 [82, 93]%	0.36 [0.32, 0.40]
	M. BatchTopK	4K	$1.00\,[0.74, 1.37]$	$1.58\,[1.21, 2.10]$	85 [72, 90]%	0.30 [0.27, 0.33]
	M. BatchTopK id.	4K	$1.00\,[0.73, 1.37]$	$1.51\,[1.14, 2.02]$	77 [65, 87]%	0.42 [0.39, 0.46]
	JumpReLU	16K	$1.00\,[0.77, 1.33]$	$2.02\,[1.66, 2.46]$	88 [78, 90]%	0.26 [0.24, 0.30]
	JumpReLU id.	16K	$1.00\,[0.98, 1.04]$	$1.08\,[1.04, 1.13]$	88 [83, 97]%	0.37 [0.33, 0.41]
	M. BatchTopK	16K	$1.00\,[0.76, 1.36]$	$1.76\,[1.38, 2.24]$	87 [77, 90]%	0.27 [0.25, 0.31]
	M. BatchTopK id.	16K	$1.00\,[0.75, 1.36]$	$1.69\,[1.31, 2.19]$	85 [77, 90]%	0.29 [0.27, 0.33]
	JumpReLU	65K	$1.00\,[0.79, 1.27]$	$2.32\,[1.99, 2.69]$	88 [78, 90]%	0.27 [0.24, 0.30]
	JumpReLU id.	65K	$1.00\,[0.97, 1.03]$	$1.12\,[1.06, 1.19]$	90 [85, 97]%	0.36 [0.33, 0.41]
	M. BatchTopK	65K	$1.00\,[0.76, 1.33]$	$2.10\,[1.72, 2.60]$	88 [78, 90]%	0.27 [0.24, 0.31]
	M. BatchTopK id.	65K	$1.00\,[0.76, 1.35]$	$2.40\,[1.75, 3.36]$	78 [65, 88]%	0.31 [0.29, 0.34]
Gemma 3 270M	M. JumpReLU	16K (small)	19 [14, 26]	58 [46, 73]	96 [93, 98]%	0.25 [0.21, 0.29]
	M. JumpReLU	16K (big)	106 [86, 130]	205 [176, 240]	96 [93, 98]%	0.25 [0.22, 0.29]
	M. JumpReLU	262K (small)	19 [15, 25]	124 [56, 733]	96 [93, 98]%	0.25 [0.22, 0.29]
	M. JumpReLU	262K (big)	104 [83, 132]	173 [145, 206]	96 [93, 98]%	0.26 [0.22, 0.30]
Gemma 3 1B	M. JumpReLU	16K (small)	19 [15, 25]	50 [41, 63]	92 [86, 97]%	0.26 [0.22, 0.30]
	M. JumpReLU	16K (big)	104 [84, 131]	204 [171, 247]	92 [85, 97]%	0.26 [0.23, 0.31]
	M. JumpReLU	262K (small)	19 [15, 23]	117 [73, 199]	92 [86, 97]%	0.26 [0.22, 0.30]
	M. JumpReLU	262K (big)	104 [85, 130]	141 [116, 174]	92 [86, 97]%	0.27 [0.24, 0.32]
Gemma 3 4B	M. JumpReLU	16K (small)	18 [14, 24]	42 [34, 52]	99 [98, 100]%	0.15 [0.12, 0.19]
	M. JumpReLU	16K (big)	104 [85, 130]	176 [150, 213]	99 [98, 100]%	0.15 [0.12, 0.20]
	M. JumpReLU	262K (small)	18 [14, 23]	1106 [637, 1531]	99 [97, 100]%	0.15 [0.12, 0.20]
	M. JumpReLU	262K (big)	104 [86, 128]	113 [96, 133]	99 [97, 100]%	0.17 [0.14, 0.22]
Gemma 3 12B	M. JumpReLU	16K (small)	18 [14, 24]	38 [30, 49]	92 [88, 99]%	0.11 [0.08, 0.14]
	M. JumpReLU	16K (big)	102 [81, 128]	168 [139, 208]	92 [88, 99]%	0.11 [0.08, 0.14]
	M. JumpReLU	262K (small)	27 [15, 83]	86 [44, 199]	92 [87, 99]%	0.11 [0.09, 0.15]
	M. JumpReLU	262K (big)	113 [88, 168]	101 [84, 125]	89 [87, 99]%	0.13 [0.10, 0.16]
Gemma 3 27B	M. JumpReLU	16K (small)	20 [16, 25]	40 [33, 50]	99 [95, 100]%	0.08 [0.06, 0.10]
	M. JumpReLU	16K (big)	95 [77, 124]	154 [127, 200]	99 [95, 100]%	0.08 [0.07, 0.10]
	M. JumpReLU	262K (small)	21 [18, 26]	517 [244, 800]	98 [96, 99]%	0.10 [0.08, 0.12]
	M. JumpReLU	262K (big)	117 [102, 135]	10917 [4507, 15578]	98 [94, 99]%	0.12 [0.10, 0.14]
GPT2-small	TopK	32K	32 [32, 32]	32 [32, 32]	93 [90, 100]%	0.29 [0.26, 0.32]
	TopK	128K	32 [32, 32]	32 [32, 32]	93 [90, 100]%	0.29 [0.26, 0.32]
Llama 3.1 8B	JumpReLU	32K	33 [27, 40]	51 [41, 61]	100 [94, 100]%	0.21 [0.19, 0.22]
	JumpReLU	131K	36 [29, 43]	57 [46, 68]	100 [94, 100]%	0.21 [0.19, 0.22]
Mistral 7B	ReLU	65K	86 [75, 96]	50236 [47612, 55093]	27 [17, 40]%	0.22 [0.21, 0.22]

D Individual Latent Idempotency Results

Figures 23 to 35 provide extended results for Section 4.4. While the results in Section 4.4 aggregate metrics over all layers for which SAEs are available within a given SAE class, the plots shown here are left unaggregated: each line shows the median over latents for a single layer, and the shaded band its interquartile range. This allows us to inspect the behavior of individual SAEs. While the overall trend is already captured by Figure 3, the unaggregated plots reveal individual SAEs that deviate from this trend and exhibit a higher degree of stability on at least one metric than is typical. We refer to these cases as outliers from the general trend.

We classify outliers using a set of heuristics derived from manual inspection of the data. These criteria are intended to identify SAEs that retain stability over several re-encoding iterations, in contrast to the dominant pattern of exponential deviation observed for most SAE classes relative to the first encoding. For each metric, we define a heuristically stable regime, evaluated on the median over latents. A re-encoding iteration x_t is considered heuristically stable if the corresponding metric remains within a fixed tolerance of its value after the first encoding, x_1, where x_i are defined according to Equation (6). We define heuristic stability criteria as follows:

1. MSE is heuristically stable if its relative deviation from the first iteration remains below 20%,
$$\frac{|\mathrm{MSE}_t - \mathrm{MSE}_1|}{\mathrm{MSE}_1} < 0.2. \tag{12}$$
2. The ℓ_2 norm is heuristically stable if it increases by less than one order of magnitude relative to the first iteration,
$$\frac{\|x_t\|_2}{\|x_1\|_2} < 10. \tag{13}$$
3. Cosine similarity is heuristically stable if the angular similarity to the first iterate remains high,
$$\cos(x_1, x_t) > 0.85. \tag{14}$$
4. IoU is heuristically stable if the active latent support remains largely preserved,
$$\mathrm{IoU}(S_1, S_t) > 0.85, \tag{15}$$
where S_t denotes the active latent set at iteration t.
5. The ℓ_0 norm is heuristically stable if the support size changes by less than 20% relative to the first iteration,
$$\frac{||S_t| - |S_1||}{|S_1|} < 0.2. \tag{16}$$

We classify an SAE as an outlier if it satisfies the corresponding heuristic stability criterion for all ten observed re-encoding iterations on at least one metric. This contrasts with the dominant behavior, where most SAEs fail to remain within the heuristically stable regime for any metric across the full re-encoding trajectory. Such SAEs therefore deviate from the dominant instability pattern and are analyzed separately. A more detailed classification and discussion of these outliers is provided in Appendix E.

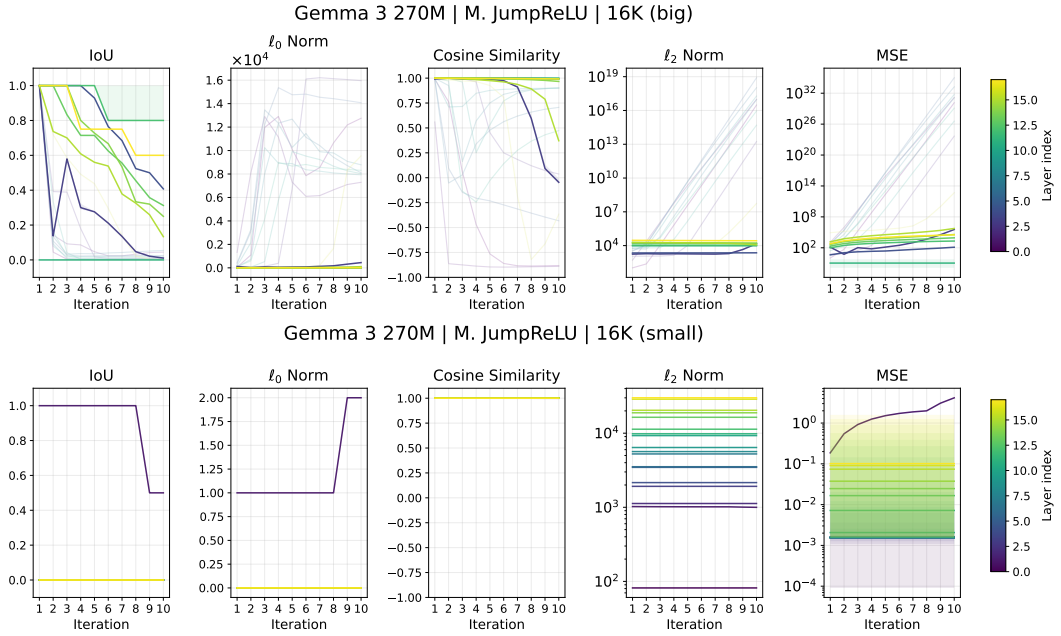


Figure 23: Idempotency metrics for decoder latents of Gemma 3 270M with 16K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

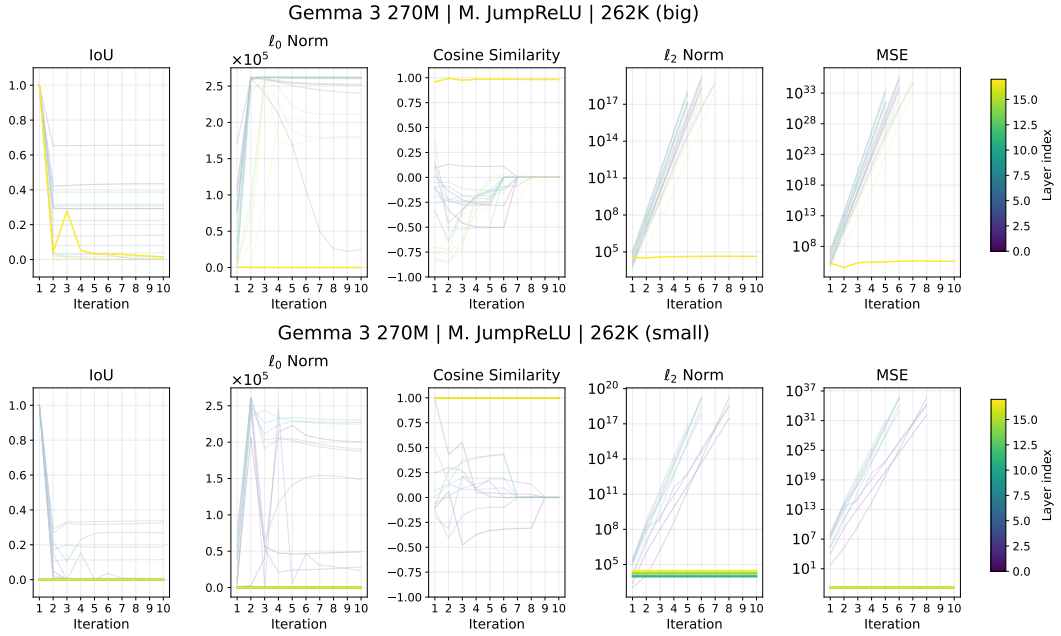


Figure 24: Idempotency metrics for decoder latents of Gemma 3 270M with 262K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

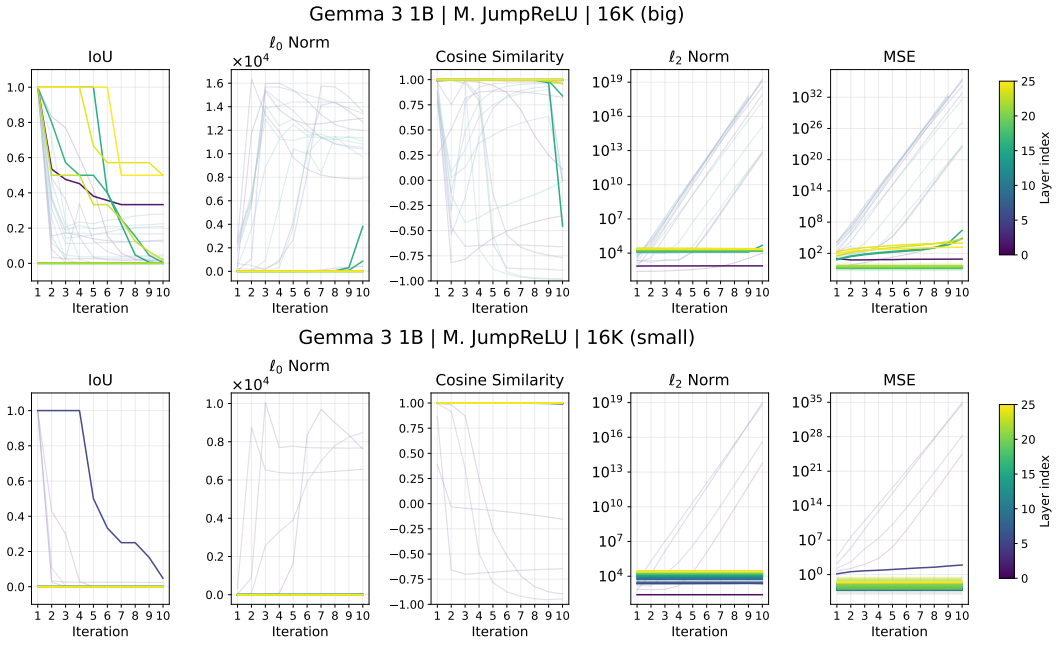


Figure 25: Idempotency metrics for decoder latents of Gemma 3 1B with 16K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

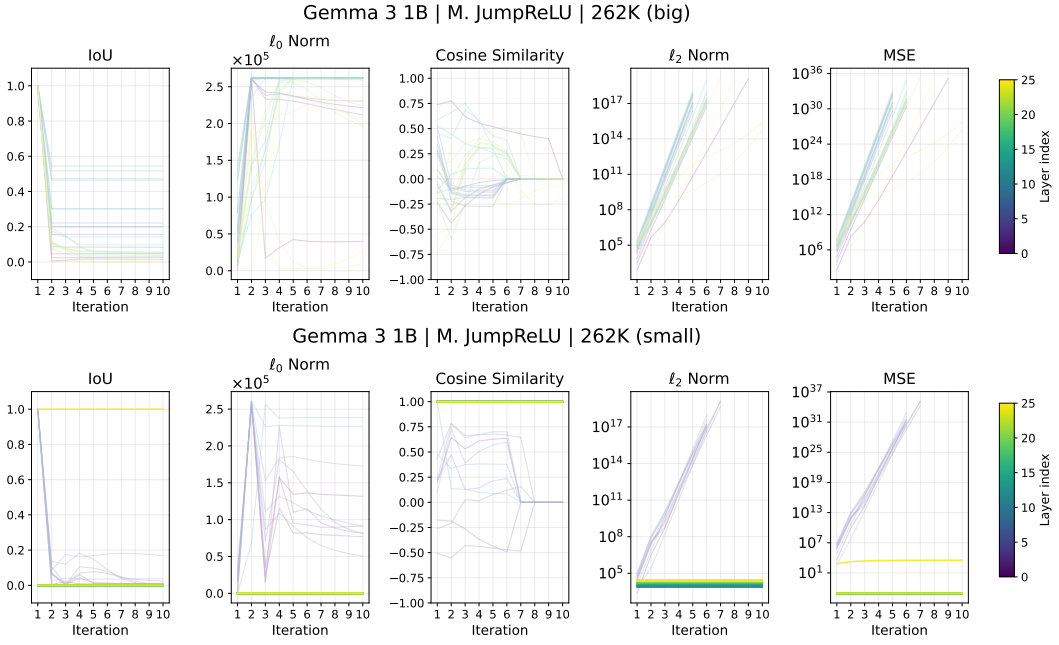


Figure 26: Idempotency metrics for decoder latents of Gemma 3 1B with 262K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

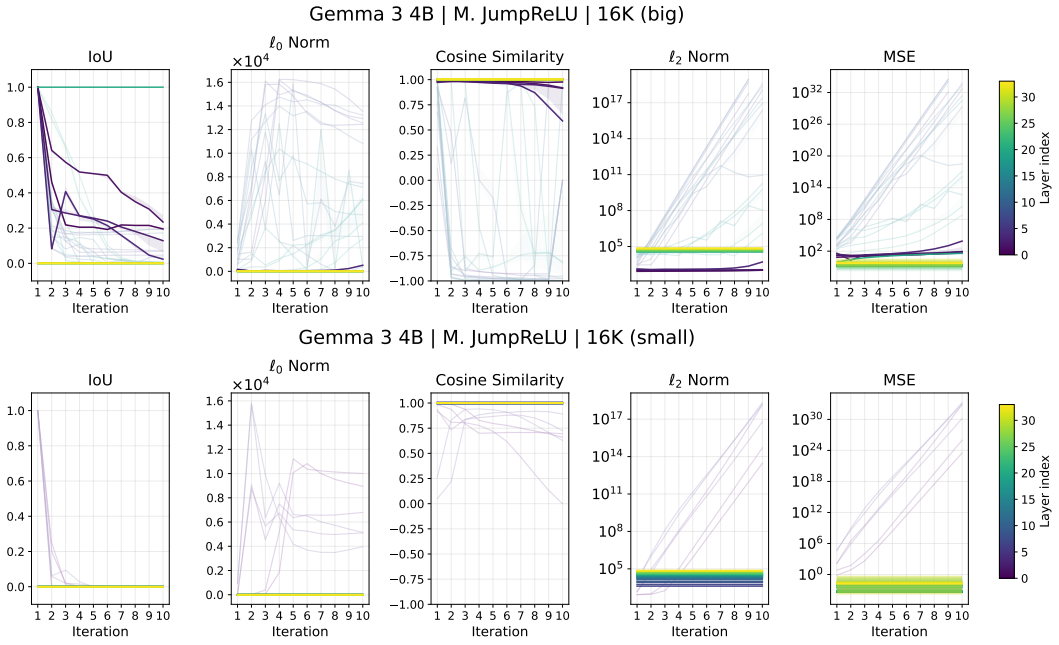


Figure 27: Idempotency metrics for decoder latents of Gemma 3 4B with 16K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

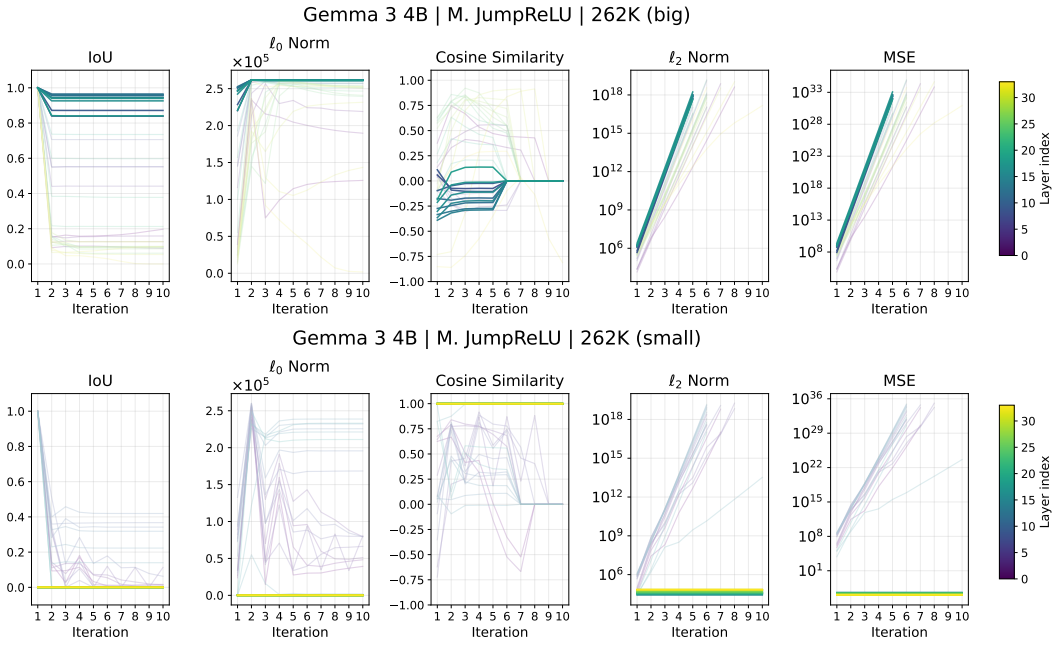


Figure 28: Idempotency metrics for decoder latents of Gemma 3 4B with 262K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

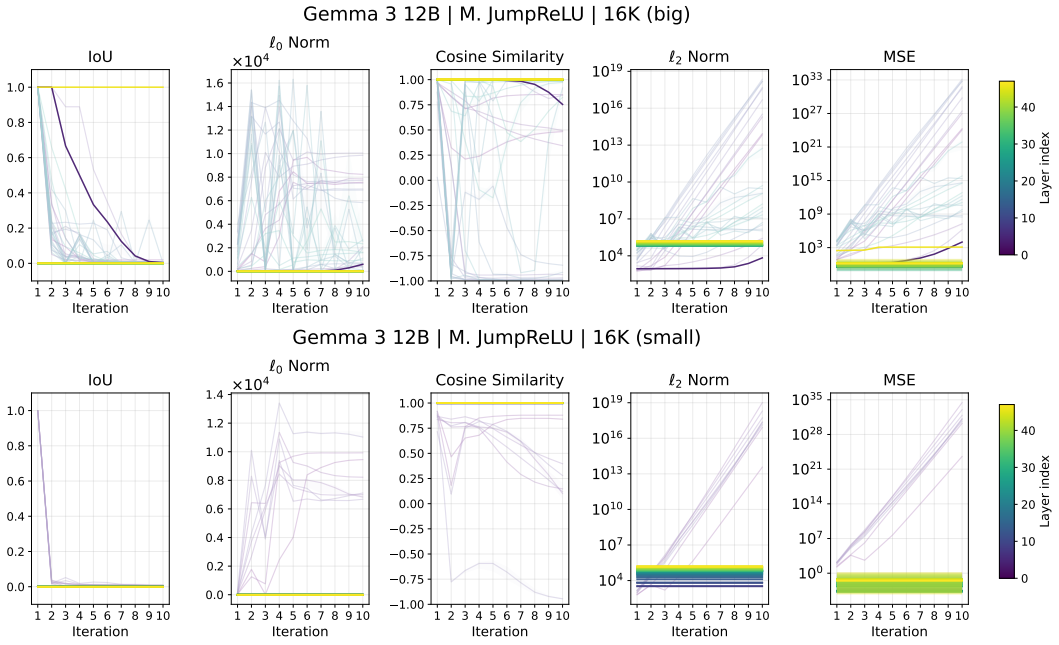


Figure 29: Idempotency metrics for decoder latents of Gemma 3 12B with 16K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

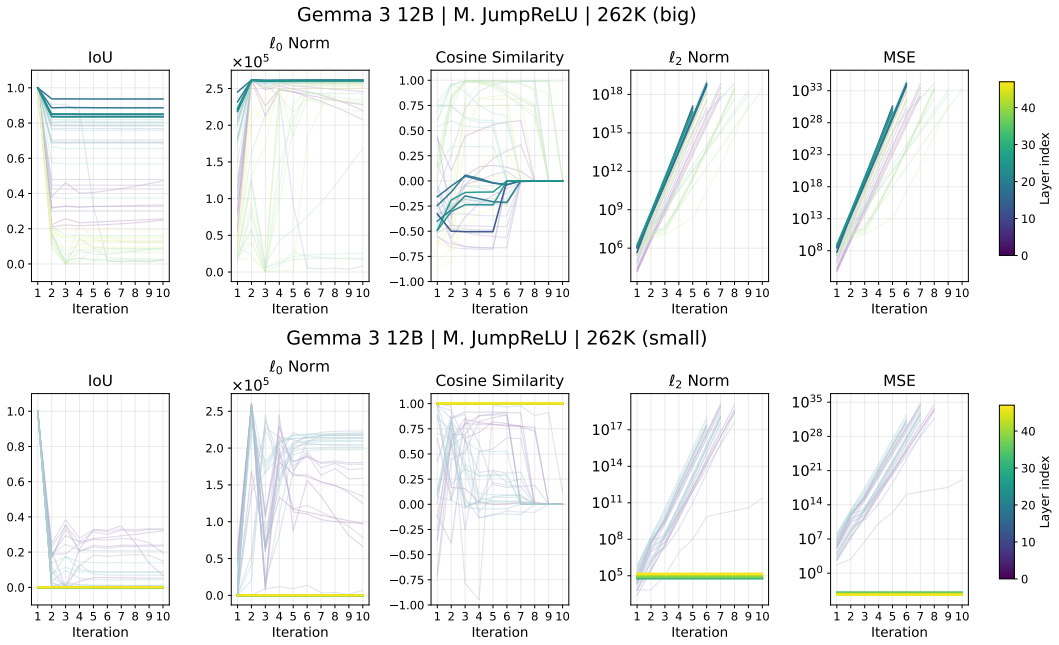


Figure 30: Idempotency metrics for decoder latents of Gemma 3 12B with 262K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

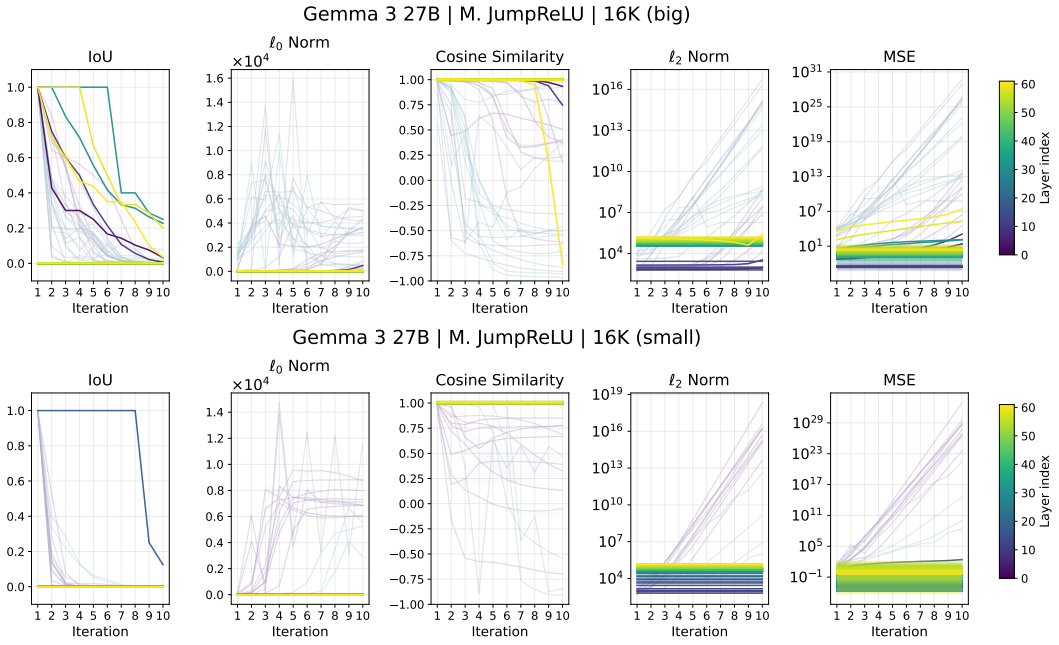


Figure 31: Idempotency metrics for individual decoder latents of Gemma 3 27B with 16K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

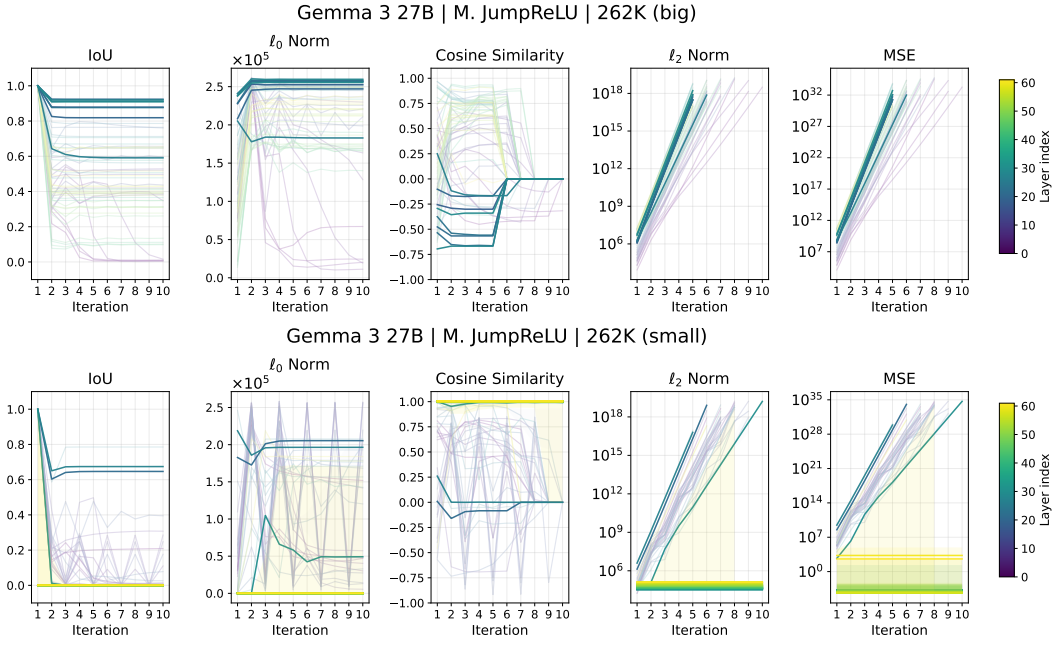


Figure 32: Idempotency metrics for decoder latents of Gemma 3 27B with 262K SAE. Top: high ℓ_0 (big). Bottom: low ℓ_0 (small).

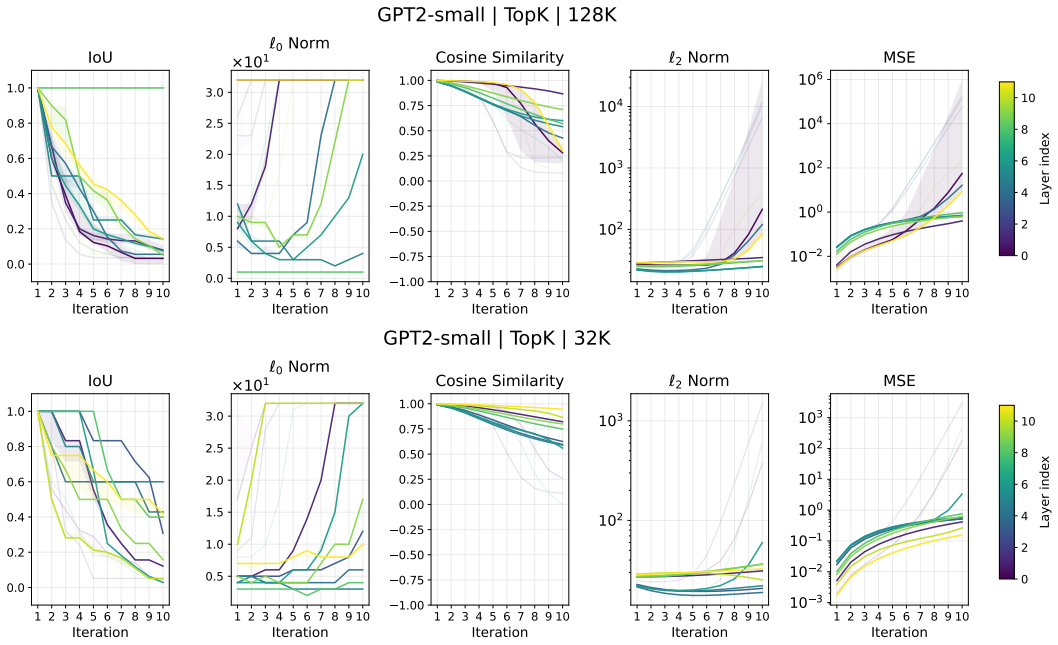


Figure 33: Idempotency metrics for decoder latents of GPT2-small. Top: 128K SAE. Bottom: 32K SAE.

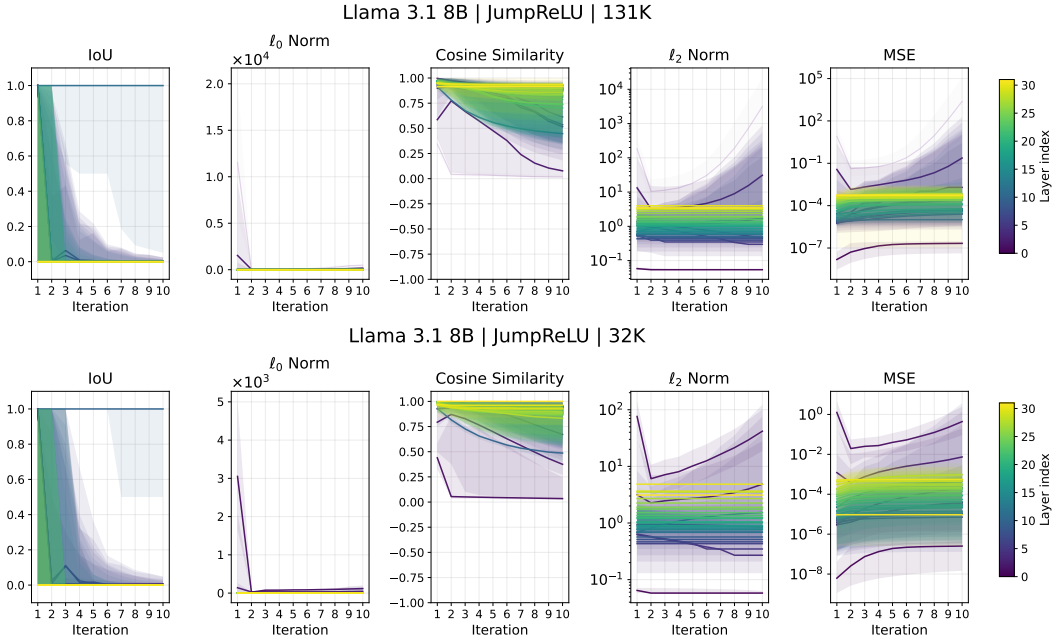


Figure 34: Idempotency metrics for decoder latents of Llama 3.1 8B. Top: 131K SAE. Bottom: 32K SAE.

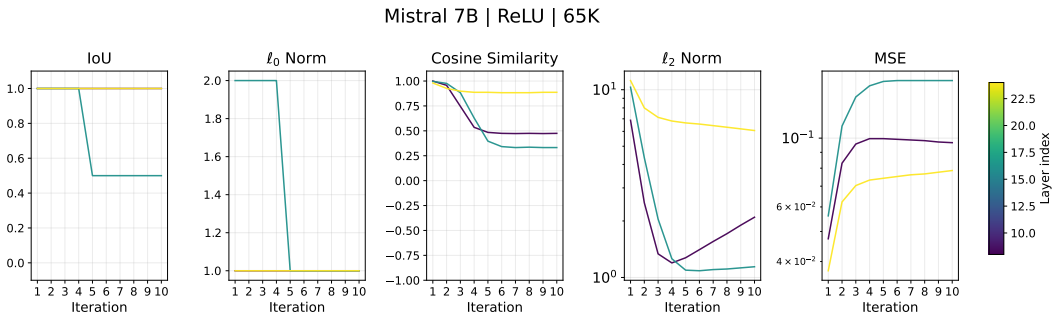


Figure 35: Idempotency metrics for decoder latents of Mistral 7B with 65K SAE.

E Outlier analysis

The results in Section 4.4 suggest that no SAE class is idempotent on its latents when inspected in aggregate. Here, aggregate refers to metrics whose median and interquartile range are first computed over latents and then averaged over all layers for which SAEs of the same class are available. However, when individual SAEs are inspected separately, i.e., before aggregating across SAEs within a class, we observe a substantial number of outliers that deviate from this general pattern. We define an SAE as an outlier if it remains within the heuristically stable regime defined in Appendix D for at least one metric across ten re-encoding iterations. The behavior of these outliers on the investigated metrics is highlighted in Figures 23 to 35. In this section, we discuss the characteristics of different outlier classes and their prevalence across the overall ensemble of SAEs.

An SAE is classified as an outlier if it remains stable according to at least one of the heuristic thresholds. However, stability on a single metric alone is not sufficient to indicate approximate idempotency. We therefore further categorize outliers according to the combinations of metrics for which they remain heuristically stable within certain regimes of the observed re-encoding range. This yields the six outlier types described below. Together, these categories capture almost all relevant outlier behavior, excluding only SAEs that are stable on a single metric and a small residual class of otherwise unexplained outliers.

Table 8: Outlier types defined by heuristic stability criteria defined in Appendix D at re-encoding iterations $t = 5$ and $t = 10$. A checkmark indicates that the metric remains within the corresponding heuristically stable regime, while a cross indicates instability beyond the set threshold limit. A dash indicates that the corresponding metric is not used to define the classification pattern.

	IoU		ℓ_0		Cosine		ℓ_2		MSE	
Type	$t=5$	$t=10$	$t=5$	$t=10$	$t=5$	$t=10$	$t=5$	$t=10$	$t=5$	$t=10$
Outlier 1	×	–	–	–	✓	×	✓	×	–	–
Outlier 2	✓	×	✓	×	✓	✓	✓	✓	–	–
Outlier 3	< 0.5	< 0.5	–	< 0.5	✓	✓	✓	✓*	–	–
Outlier 4	< 0.5	< 0.5	–	< 0.5	✓	✓	✓	✓	–	–
Outlier 5	✓	✓	–	$[0.5, 10]$	–	×	✓	✓	–	–
Outlier 6	✓	✓	–	$[0.5, 10]$	✓	✓	✓	✓	–	–

Table 8 summarizes the classification patterns used to distinguish the six outlier types. A checkmark indicates that the corresponding metric satisfies the heuristic stability criterion at the specified re-encoding iteration $t = 5$, or $t = 10$, while a cross indicates that the criterion is violated. A dash denotes that the metric is not part of the defining criterion for the respective outlier type. The asterisk on the ℓ_2 entry of Outlier 3 at $t = 10$ indicates that, while the median ℓ_2 norm remains stable, some inputs reach extremely high ℓ_2 norms. Entries given as numerical ranges or thresholds are also part of the classification pattern. Thus, unlike dashed entries, they impose an explicit condition on the corresponding metric. For example, for the last two outlier classes, where the ℓ_0 deviation at iteration $t = 10$ is required to lie in the range $[0.5, 10]$. Since the ground-truth construction has support size $\ell_0 = 1$, this interval is treated as an approximately valid regime. It excludes collapsed representations with effectively no active latents, while still allowing deviations from the ground-truth support. By contrast, entries of the form $\ell_0 < 0.5$ indicate that the initial active latent set has effectively vanished, which we interpret as a functionally dead regime. If an SAE satisfies multiple classification patterns, we assign it to the class with the more restrictive criteria. Note that MSE stability is not used as an explicit criterion for assigning SAEs to any outlier class. However, all SAEs that remained stable in MSE were assigned to one of the defined outlier categories based on the requirements on other metrics. Thus, MSE-stable cases are captured implicitly by the classification scheme, even though MSE stability itself is not a defining requirement. The resulting distribution across outlier classes is shown in Figure 36.

Representative examples from the outlier categories are shown in Figure 37. A full breakdown of Experiment 4 by SAE, with outliers highlighted, is provided in Appendix D. In this Section we will

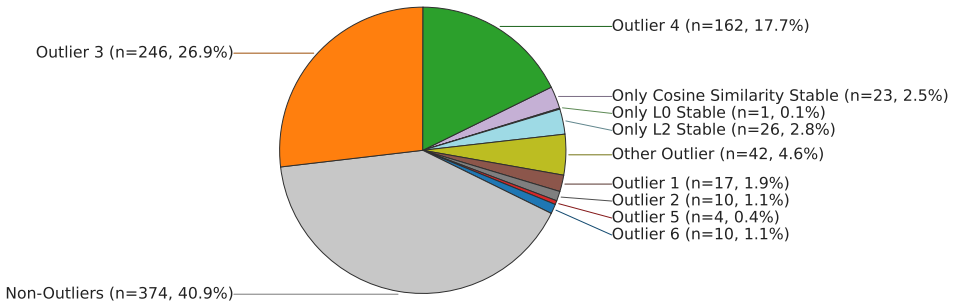


Figure 36: Overview of the outlier classification. SAEs are classified as outliers if they remain within the heuristically stable regime for at least one metric across all ten observed re-encoding iterations. The resulting outlier types are defined by the combinations of metrics for which heuristic stability is preserved, defined in Table 8. The category "Non-Outliers" contains the more typical SAEs that do not satisfy the outlier criterion for any metric.

discuss each type of outlier we could identify in our analysis and how close they came to achieving the criteria for idempotency defined in 4.1.

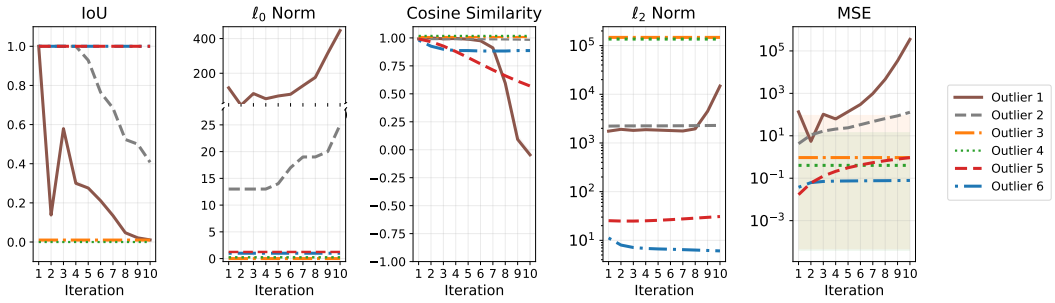


Figure 37: Experiment 4 results for six individual SAEs. Each one is representative of one type of common outlier. Outlier 1: Gemma 3 270M 16K (big) Layer 3; Outlier 2: Gemma 3 270M 16K (big) Layer 4; Outlier 3: Gemma 3 27B 16K (small) Layer 60; Outlier 4: Gemma 3 27B 16K (small) Layer 61; Outlier 5: GPT2-small 128K Layer 8; Outlier 6: Mistral 7B 65K Layer 24. Lines show the median over latents, shaded bands the interquartile range.

Outliers 1 & 2. The first two outlier classes both exhibit a distinctive heuristically stable phase in ℓ_2 norm and cosine similarity. In the first class case, the ℓ_2 norm remains nearly constant for approximately six iterations with cosine similarity close to 1, before the typical exponential instability sets in. In the second class case this heuristically stable phase extends for all 10 iterations for the ℓ_2 norm and cosine similarity. In both cases, however, the IoU tells a starkly different story. In both cases the active feature set is unstable and IoU eventually collapses. For outlier class 1 this collapse is almost immediate, whereas outlier class 2 delays it by about 5 iterations. Together these two classes demonstrate that activation-level and latent-level stability are independent properties, and that the latter can fail silently without any visible degradation in reconstruction quality. Since both SAEs in this example are trained with a Matryoshka loss, we can further break down the results of ℓ_0 by latent index, shown in Figure 38. During early iterations, low-index abstract latents exhibit significantly lower ℓ_0 than higher-index latents, consistent with hierarchical feature behavior; specific features activate their abstract parents, but not vice versa. With further iterations this relationship inverts, and low-index latents begin to exhibit dramatically spiking ℓ_0 metric, suggesting a cascading instability originating in the most abstract layers of the hierarchy. This effect is qualitatively similar across both outlier classes but considerably more pronounced in the second.

Outliers 3 & 4. The third and fourth outlier classes present what appears at first glance to be the most directionally stable behavior of any outlier type. The cosine similarity remains at one

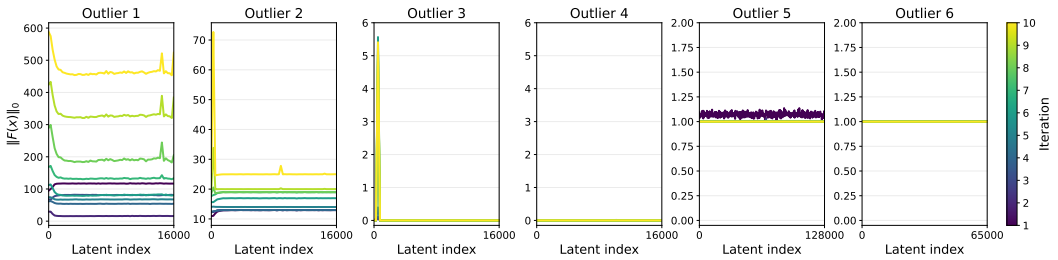


Figure 38: Breakdown of the ℓ_0-norm results by latent index and iteration for the same set of outlier SAEs as in Figure 37. Outlier 1-4 are trained using a Matryoshka loss. This implicitly sorts the latents by level of abstraction. More abstract features are assigned low-index latents and very specific features are assigned high-index latents. Outlier 5 and 6 were trained without a Matryoshka loss; the order of their latents is arbitrary.

throughout all iterations. However, this is entirely deceptive. Both classes are characterized by an ℓ_0 of approximately zero, meaning the SAE fails to activate even the single latent present by construction. The apparent directional stability is not a sign of faithful reconstruction but an artifact of the decoder bias dominating the output — with no active latents the reconstruction collapses to b_d, so cosine similarity effectively measures $\cos(d_i + b_d,\ b_d)$, which approaches 1 when $\|b_d\| \gg \|d_i\|$. Therefore, the SAE is most likely not recovering the input, but ignoring it entirely. The distinction between the two classes only becomes apparent when examining the breakdown of ℓ_0 by latent index (Figure 38). In the third class, a small fraction of very low-index latents do activate despite the near-zero median. The inputs for which these low-index latents activate undergo the usual explosion in ℓ_2 and MSE. However, they lie outside the interquartile range and thus cannot be seen in these plots. In the fourth class the ℓ_0 is truly zero across all latent indices, representing a degenerate but internally consistent fixed point in which the SAE has effectively collapsed to a constant function.

Outliers 5 & 6. The fifth and sixth outlier classes are the most well-behaved of all, and stand in stark contrast to the pathological behavior observed elsewhere. Both cases maintain an ℓ_0 score close to 1 throughout all observed iterations – fractionally above 1 for the fifth class and exactly 1 for the sixth – with correspondingly stable IoU values of just below and exactly 1 respectively. The ℓ_2 norm is also relatively stable. More precisely, they are not perfectly constant but close to it. This means that both classes successfully extract the single correct latent from the synthetic input and maintain it consistently across all observed re-encoding iterations, a property that no other SAEs achieve. The distinction between the two lies in the cosine similarity. In the case representing the fifth outlier type, cosine similarity declines steadily across iterations, indicating that while the correct latent is identified the reconstructed activation drifts directionally with each application of the SAE. In the case representing the sixth class cosine similarity stabilizes at approximately 0.9, indicating that the directional drift is bounded and the reconstruction converges to a stable fixed point.

This makes the sixth outlier class the closest approximation to genuinely idempotent behavior observed in our experiments. Among all inspected SAEs, ten fall into this category. One is a Mistral-7B SAE (Engels et al., 2025a), which is noteworthy because it belongs to the simplest SAE class considered. Three are from the Gemma Scope 2 corpus (McDougall et al., 2025), thus being Matryoshka SAEs. Two further cases come from the SAEBench corpus (Karvonen et al., 2025) and are both Matryoshka BatchTopK SAEs with width 65K on Gemma-2. The remaining cases are self-trained SAEs according to Appendix F, including JumpReLU and Matryoshka BatchTopK variants on Gemma-2.

F Training Details

We follow the SAE training protocol from Karvonen et al. (2025), using directly comparable training conditions across architectures and build on the original codebase[23][24]. As in the SAEBench setup,

[23] `https://github.com/adamkarvonen/dictionary_learning_demo`
[24] `https://github.com/saprmarks/dictionary_learning`

we train SAEs on layer 12 and target several sparsity levels. The hyperparameters are referenced in Table 9.

Table 9: Hyperparameters for SAE training. Unless stated otherwise, these follow the SAE training setup from Karvonen et al. (2025).

Hyperparameter	Value
Layer	12
Target ℓ_0 values	$\{20, 40, 80, 160, 320, 640\}$
SAE widths	4K, 16K, 65K
Tokens processed	500M
Learning rate	3×10^{-4}
Learning rate warmup	1,000 steps
Sparsity penalty warmup	5,000 steps
Learning rate decay	Last 20% of training
Dataset	The Pile (Gao et al., 2020)
Batch size	2,048
LLM context length	1,024
Activation buffer size	250,000 — replenished when half empty (Nanda, 2023)
Decoder initialization	Transpose of encoder — untied during training

In contrast to the SAEBench training protocol, we introduce an additional auxiliary loss term designed to encourage stability on the active latent set under re-encoding, with the goal of promoting more idempotency-aligned behavior. This term is weighted by a scalar coefficient ζ, which controls its contribution to the total training objective. The auxiliary loss itself is described below.

For JumpReLU SAEs, we swept $\zeta \in \{0.5, 1, 2, 4\}$ on 4K-width models. Since the auxiliary term was intended to enforce high overlap on the active-set, we selected $\zeta = 2$; this setting achieved IoU above 0.99 and, among the high-IoU runs, yielded the best reconstruction loss. However, the auxiliary term also affected sparsity. Across the sweep, the trained SAEs were substantially more dense than intended. The realized ℓ_0 was approximately twice the declared target value. Therefore, in subsequent JumpReLU training runs, we set the target ℓ_0 to half of the intended value, which brought the realized ℓ_0 back into the desired range.

For Matryoshka BatchTopK SAEs, we similarly swept $\zeta \in \{0.5, 1, 2, 4, 20\}$. The results were overall robust to this hyperparameter. Reconstruction losses were similar across the sweep, while $\zeta = 1$ gave the best IoU score. We therefore selected $\zeta = 1$ for the final Matryoshka BatchTopK training configuration. No additional sparsity correction was required for these models, since sparsity is controlled directly through the BatchTopK configuration.

With this adjustment, SAEs were trained for all target ℓ_0 values at layer 12 and widths 4K, 16K, and 65K.

Auxiliary Intersection Loss For training, we introduce an additional auxiliary loss term designed to encourage stability of the active latent support under re-encoding and thus encourage better performance on idempotency metrics. For JumpReLU SAEs, this term takes the form

$$\mathcal{L}^{\text{JumpReLU}}_{\text{Intersection}} = -\zeta \cdot \frac{\|\mathcal{H}(F(x)) \odot \mathcal{H}(F(\hat{x}))\|_0}{\ell_0^{(\text{target})}}, \tag{17}$$

where $\mathcal{H}$ denotes the Heaviside step function, $\odot$ denotes element-wise multiplication, and $\hat{x} = (G \circ F)(x)$ is the SAE reconstruction of x. The term rewards overlap between the active latent set of the original input and that of the re-encoded reconstruction. Since the overall objective is minimized, the negative sign encourages this overlap to be large, favoring reconstructions whose re-encoding preserves the initial active support.[25]

[25]Note that this term is natural to implement for JumpReLU SAEs. Their active support is already defined through Heaviside thresholding, which is optimized using pseudo-gradients in standard JumpReLU training. The auxiliary term therefore reuses the same threshold-based support definition.

Similarly, for Matryoshka BatchTopK SAEs, we use an auxiliary overlap term defined on the selected BatchTopK latent activations. Let $z = F(x)$ and $\hat{z} = F(\hat{x})$, where again $\hat{x}$ denotes the SAE reconstruction of x. We define

$$\mathcal{L}_{\text{Intersection}}^{\text{MatryoshkaBatchTopK}} = -\zeta \cdot \frac{1}{\ell_0^{\text{target}}} \sum_{i \in \mathcal{K}_{\text{BatchTopK}}} \frac{z_i \hat{z}_i}{\|z\| \|\hat{z}\|}, \tag{18}$$

where $\mathcal{K}_{\text{BatchTopK}}$ denotes the set of latent indices selected by the BatchTopK operator, rewarding overlap between the latent activations selected for the original input and those selected after reconstruction. As above, the negative sign encourages this overlap to increase under minimization of the total loss.

Both terms operate on the active support rather than on the reconstruction itself, consistent with our position that support stability, not signal fidelity, is the property of interest (Section 4.1). An alternative would be an ℓ_2-based term that penalizes the distance between the reconstruction and its re-encoding, $\|(G \circ F)(\hat{x}) - \hat{x}\|_2^2$. We have not evaluated this variant and leave it to future work.

G Additional Results for Idempotency-Loss SAEs

This section reports additional results for the SAEs trained with the idempotency-encouraging auxiliary loss. We evaluate these SAEs on the experimental setups from Experiments 1–4 and further report their performance on SAEBench (Karvonen et al., 2025) reference metrics. With the exception of Experiment 2, whose results are reported in Appendix C, the results are presented in Figures 39–56. In each figure, the top row shows the SAEs trained with the additional idempotency-encouraging auxiliary loss, while the bottom row shows the corresponding SAEBench baseline SAEs. Each line corresponds to one sparsity target and shows the median over tokens (Experiments 1 and 3) or latents (Experiment 4), with the interquartile range as a shaded band. As discussed in Section 4.5, the auxiliary loss improves IoU, ℓ_0, and cosine similarity, with the strongest gains observed for JumpReLU SAEs and more moderate improvements for Matryoshka BatchTopK SAEs, while the norm-based metrics do not improve.

However, the SAEBench metrics (Figures 57–61) reveal a more mixed picture, indicating that improved idempotency can come with architecture-dependent trade-offs in other aspects of dictionary quality.

The corresponding results are reported below and sorted by their respective experimental configuration.

G.1 Experiment 1

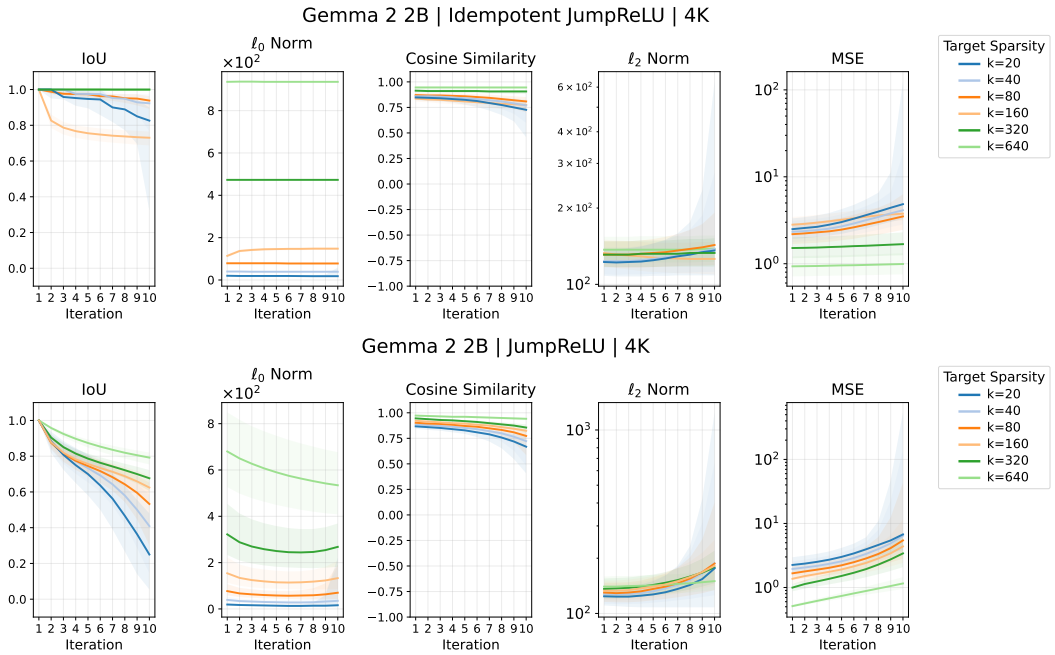


Figure 39: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 4K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 1 in Section 4.2.

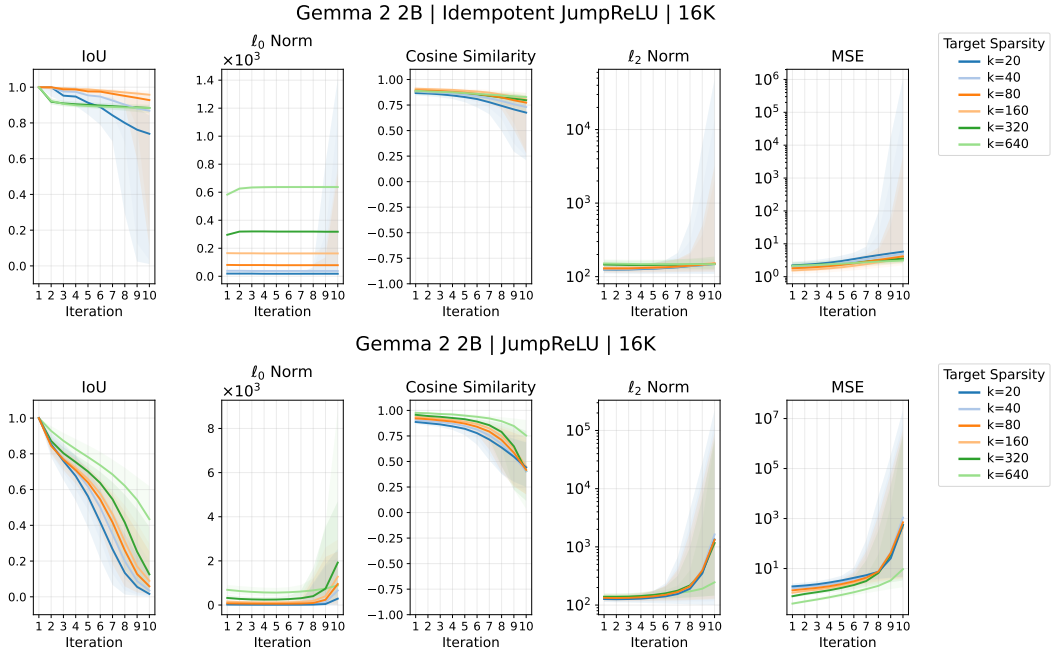


Figure 40: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 16K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 1 in Section 4.2.

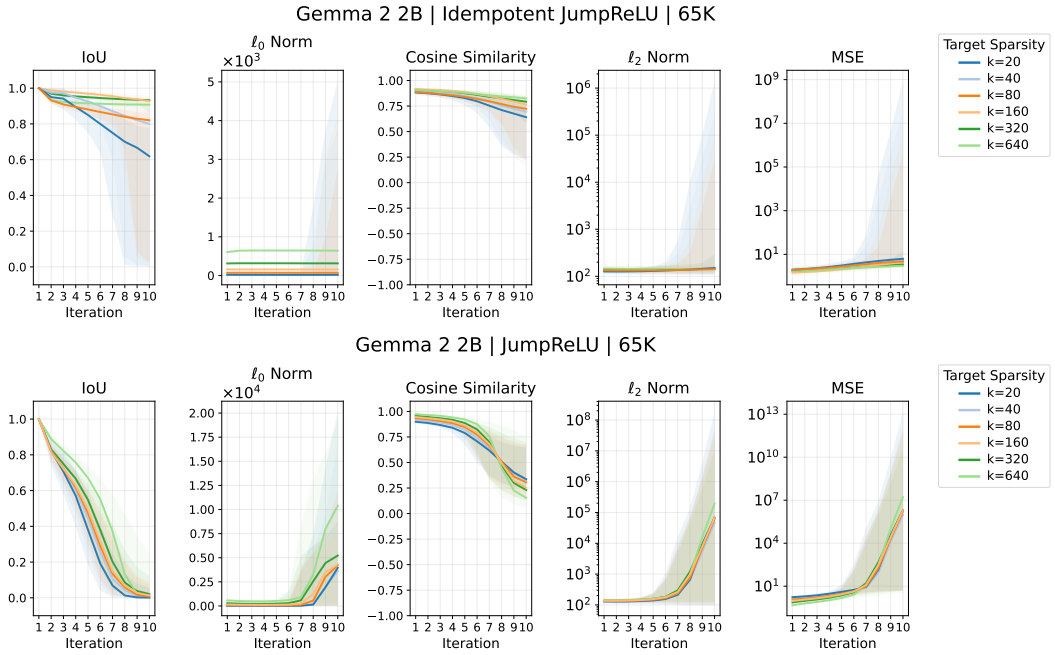


Figure 41: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 65K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 1 in Section 4.2.

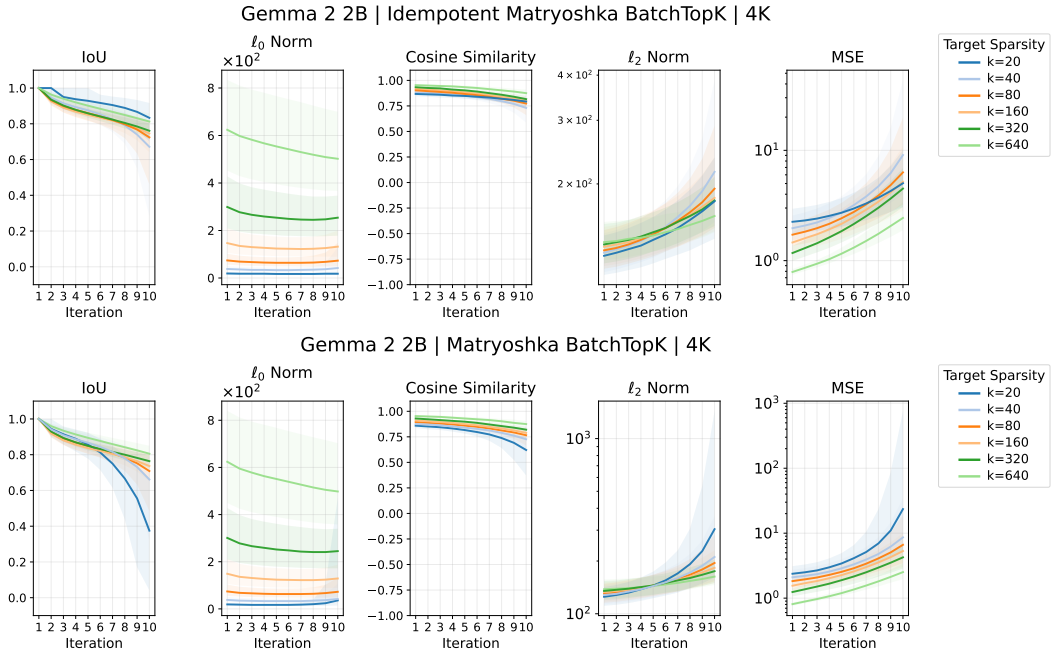


Figure 42: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatch-TopK SAEs with 4K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 1 in Section 4.2.

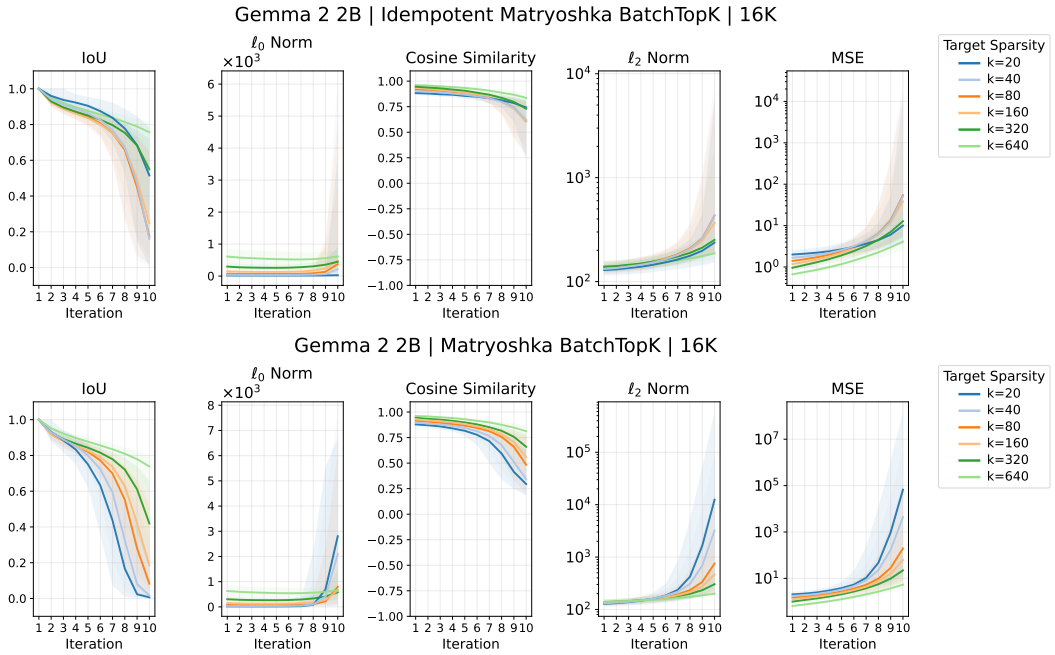


Figure 43: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatchTopK SAEs with 16K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 1 in Section 4.2.

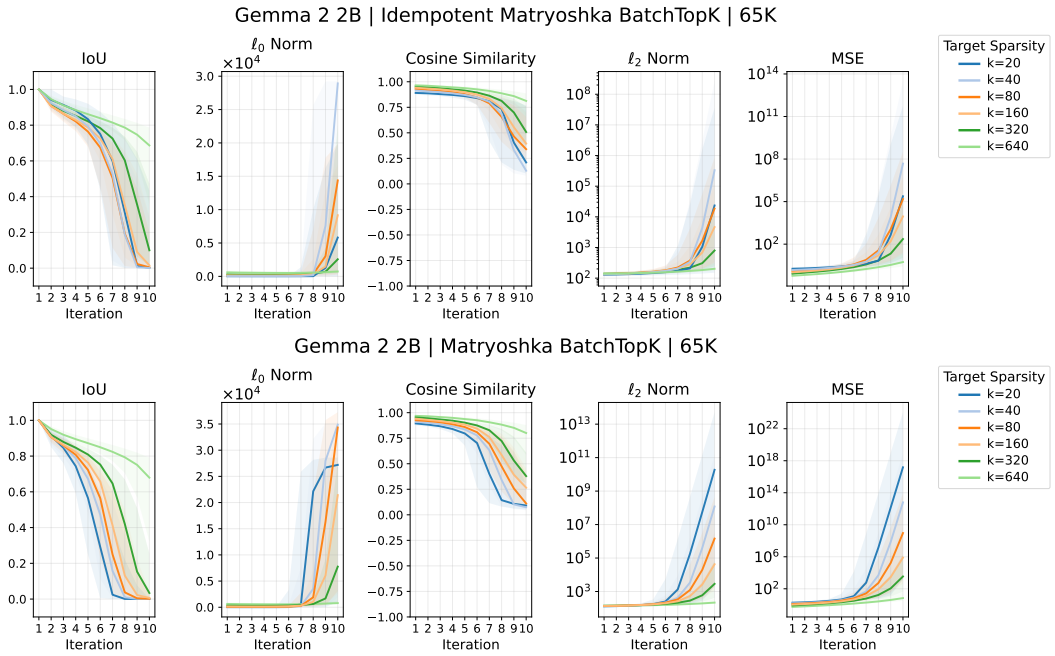


Figure 44: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatchTopK SAEs with 65K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 1 in Section 4.2.

G.2 Experiment 2

The results for Experiment 2 on the idempotent and non-idempotent SAEBench SAEs are included in the full results in Appendix C.

G.3 Experiment 3

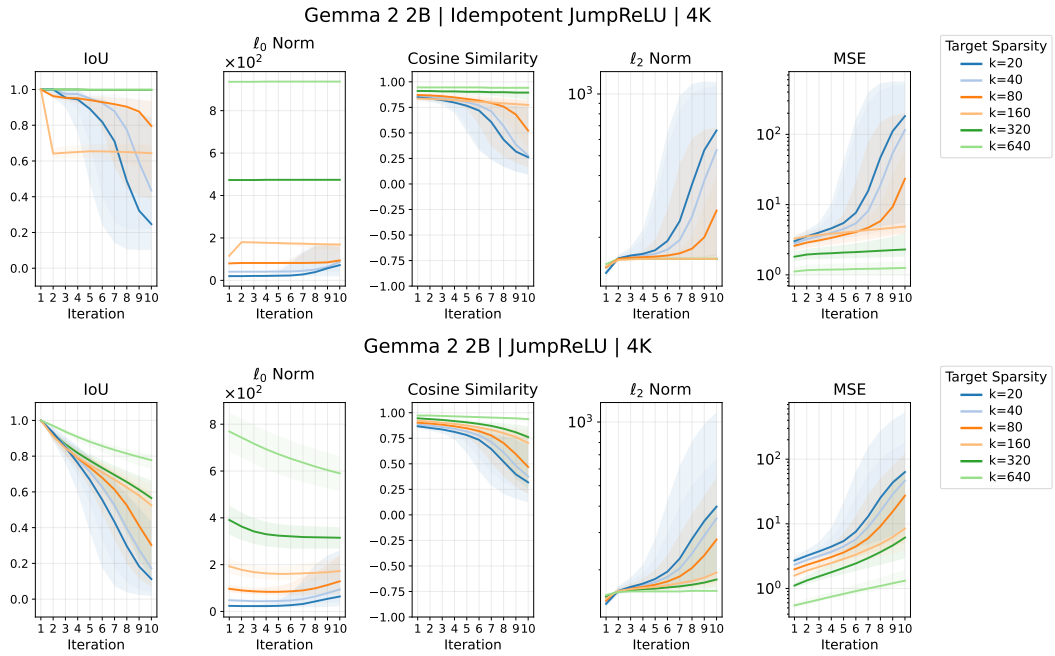


Figure 45: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 4K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 3 in Section 4.3.2.

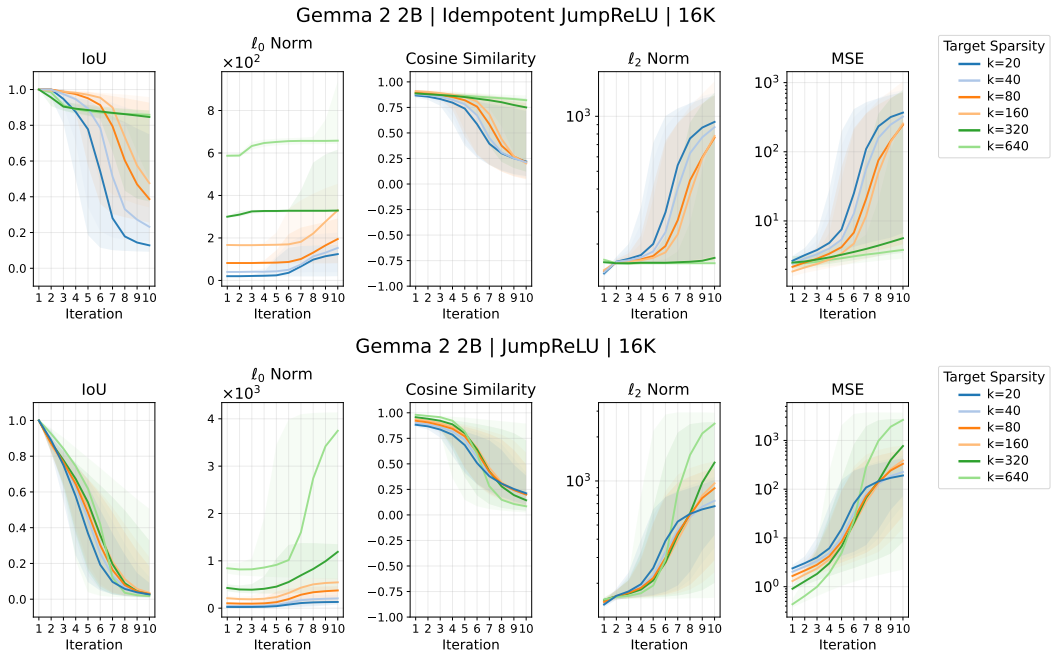


Figure 46: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 16K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 3 in Section 4.3.2.

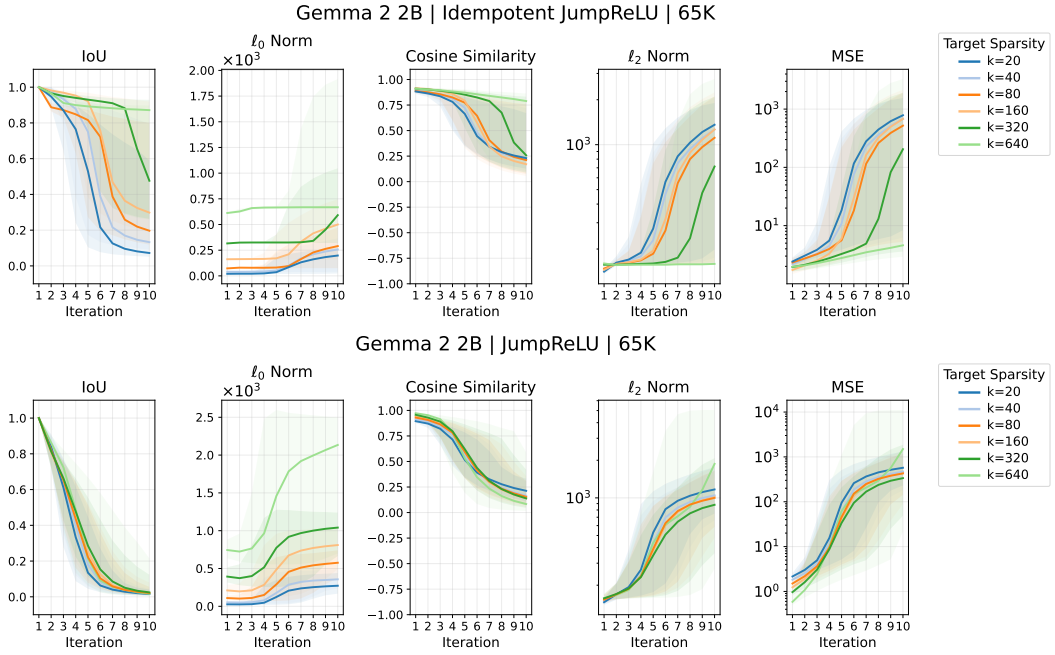


Figure 47: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 65K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 3 in Section 4.3.2.

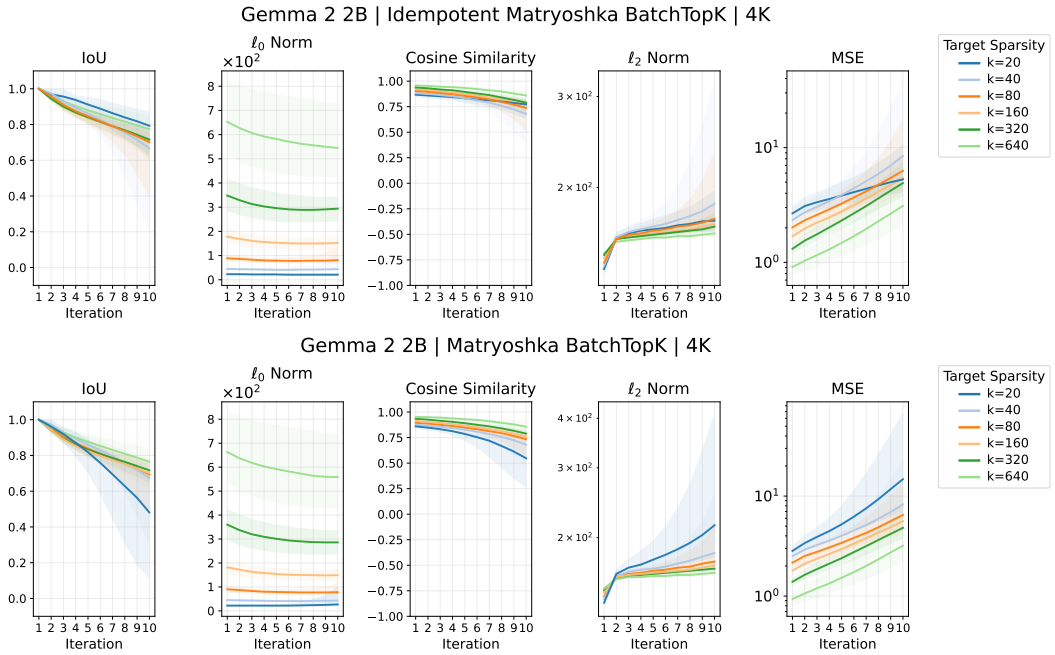


Figure 48: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatchTopK SAEs with 4K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 3 in Section 4.3.2.

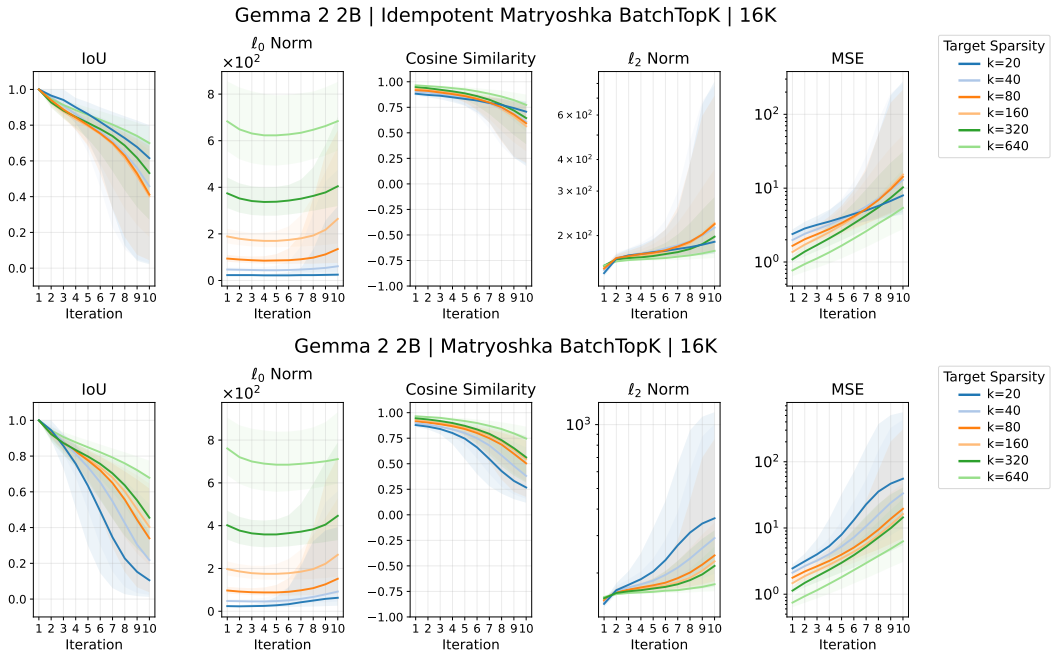


Figure 49: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatchTopK SAEs with 16K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 3 in Section 4.3.2.

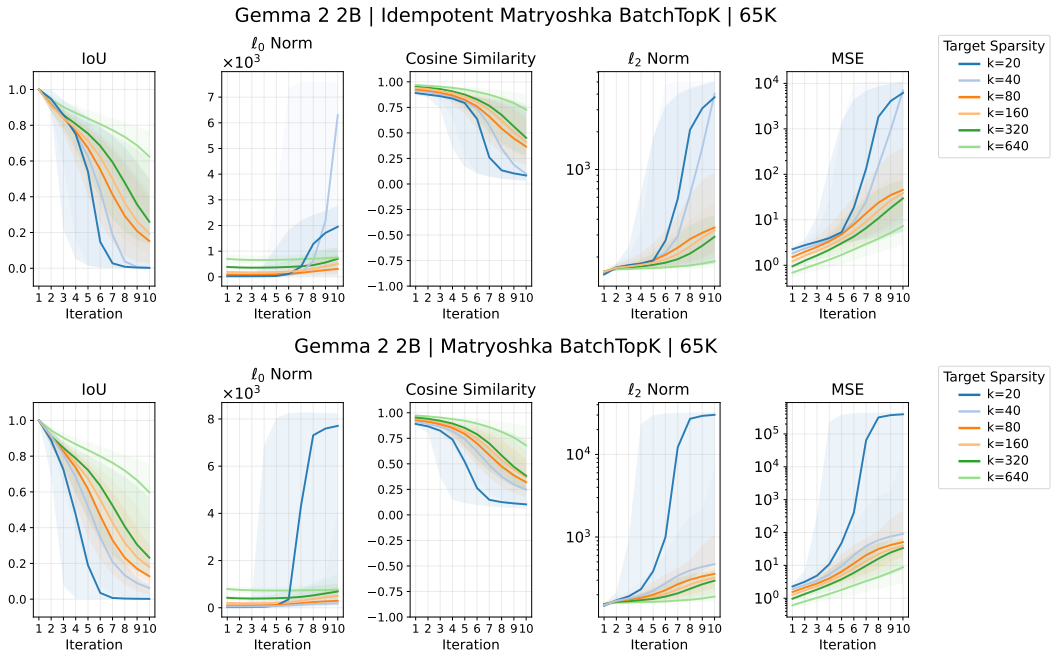


Figure 50: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatchTopK SAEs with 65K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 3 in Section 4.3.2.

G.4 Experiment 4

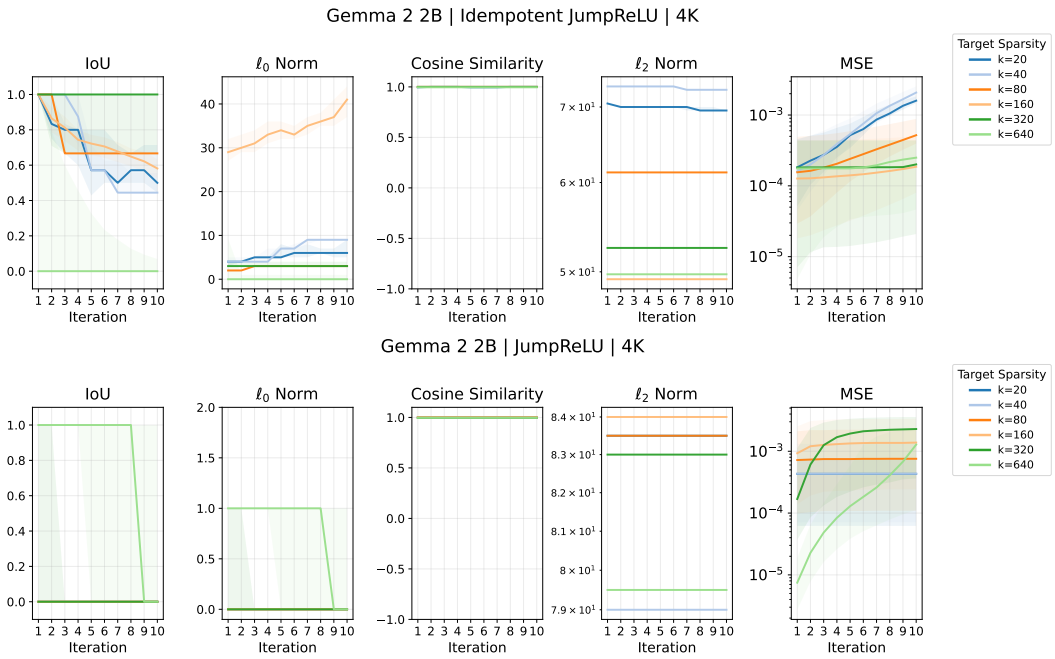


Figure 51: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 4K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 4 in Section 4.4.

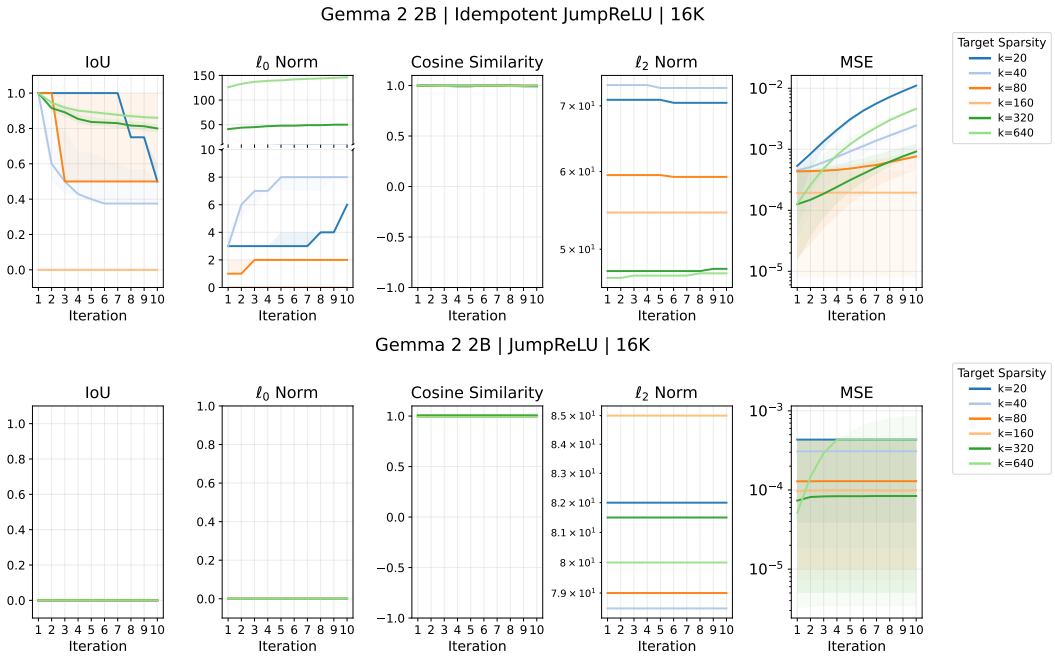


Figure 52: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 16K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 4 in Section 4.4.

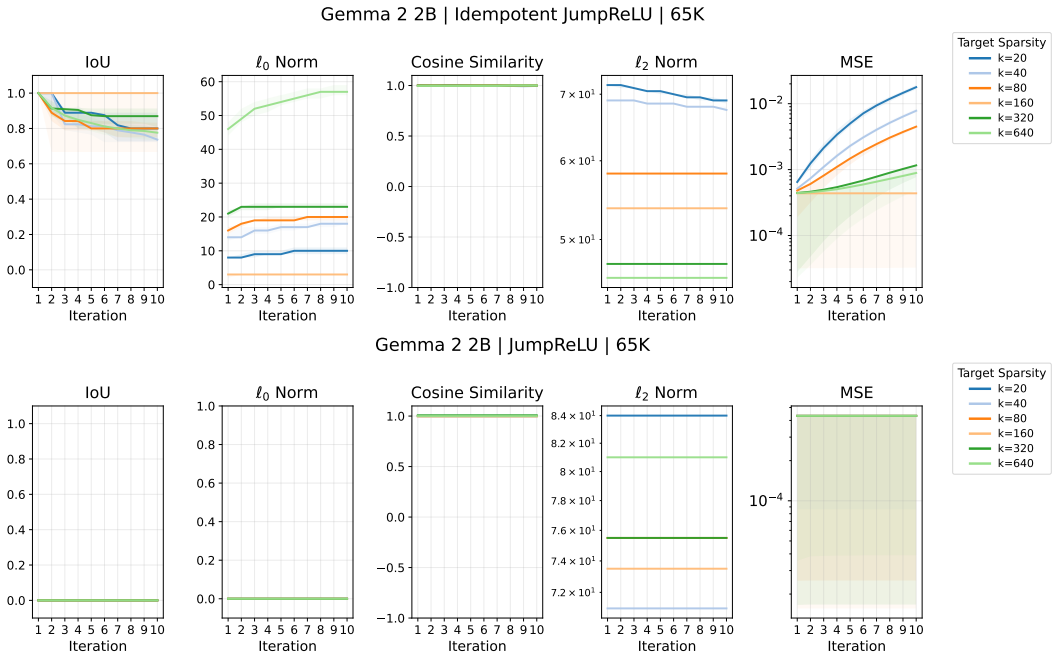


Figure 53: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) JumpReLU SAEs with 65K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 4 in Section 4.4.

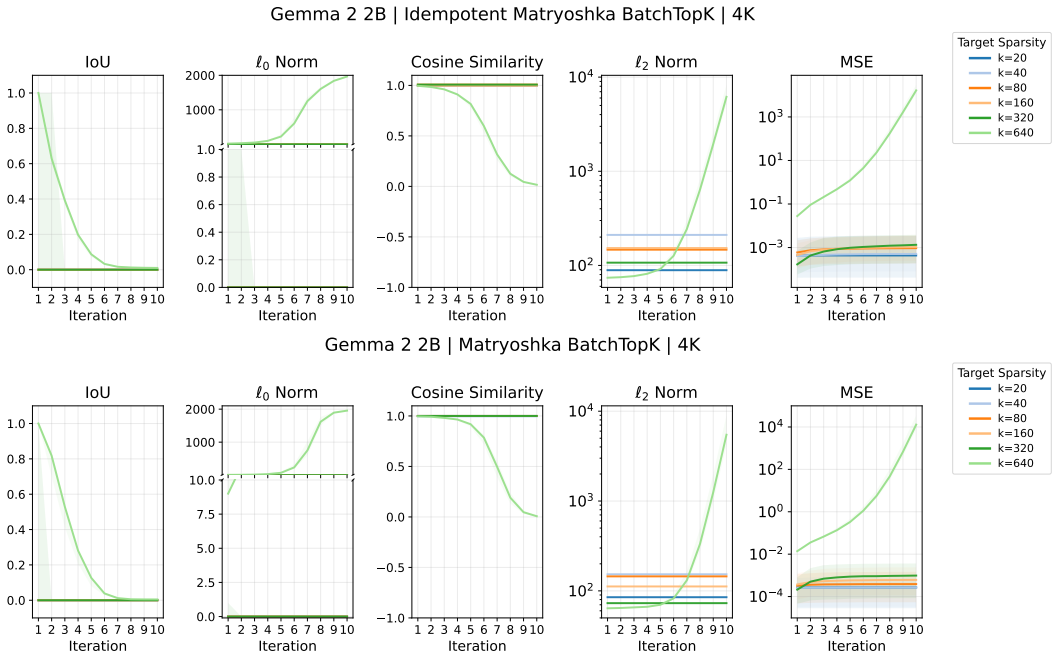


Figure 54: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatchTopK SAEs with 4K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 4 in Section 4.4.

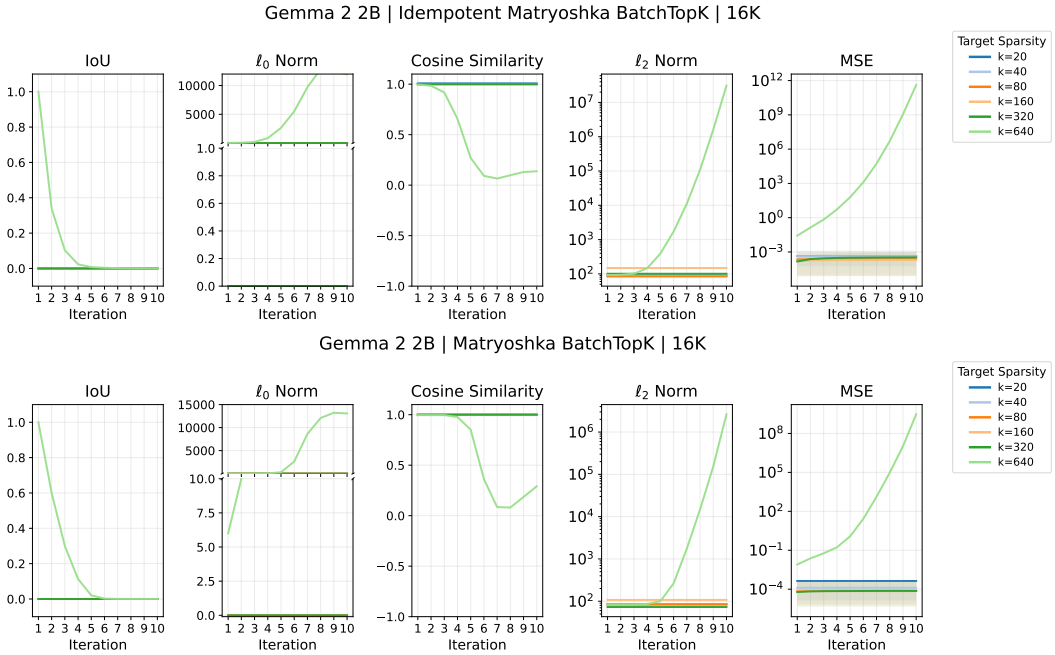


Figure 55: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatchTopK SAEs with 16K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 4 in Section 4.4.

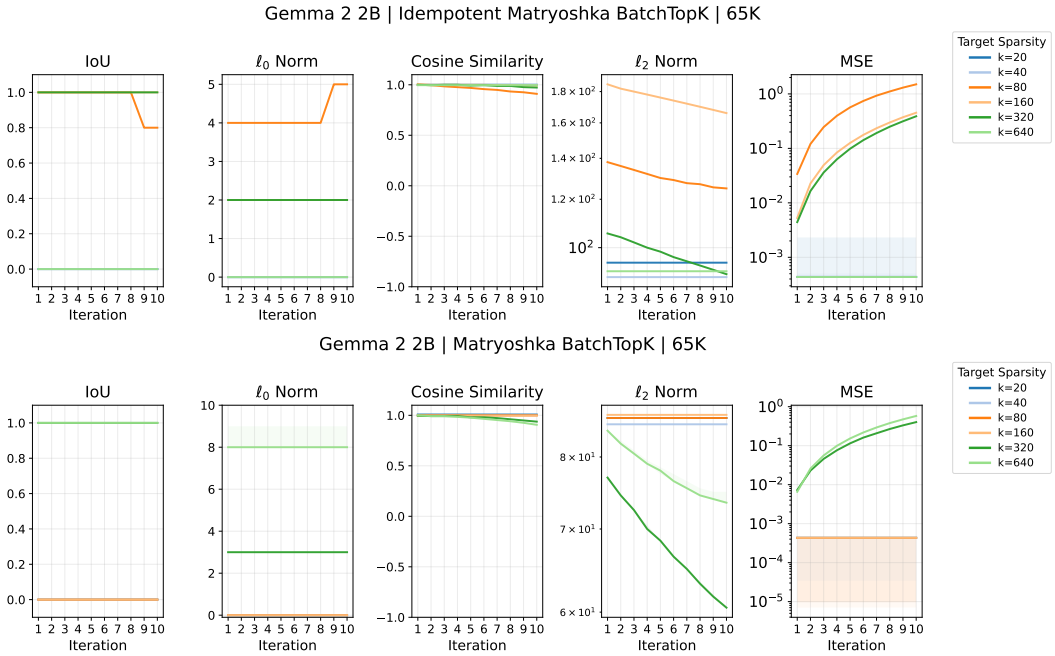


Figure 56: Comparison of idempotency auxiliary-loss (top) and baseline (bottom) MatryoshkaBatchTopK SAEs with 65K dictionaries on Gemma-2-2B. Results are computed according to the setup of Experiment 4 in Section 4.4.

G.5 SAEBench

The scaling behavior with respect to SAE width is mixed. Some metrics improve from 4K to 16K, but deteriorate again at 65K. We speculate that this may partly reflect the use of a fixed weight for the auxiliary-loss across widths. As described in Appendix F, ζ was not tuned separately for each width. Therefore, this hyperparameter choice might not be optimal for 65K width.

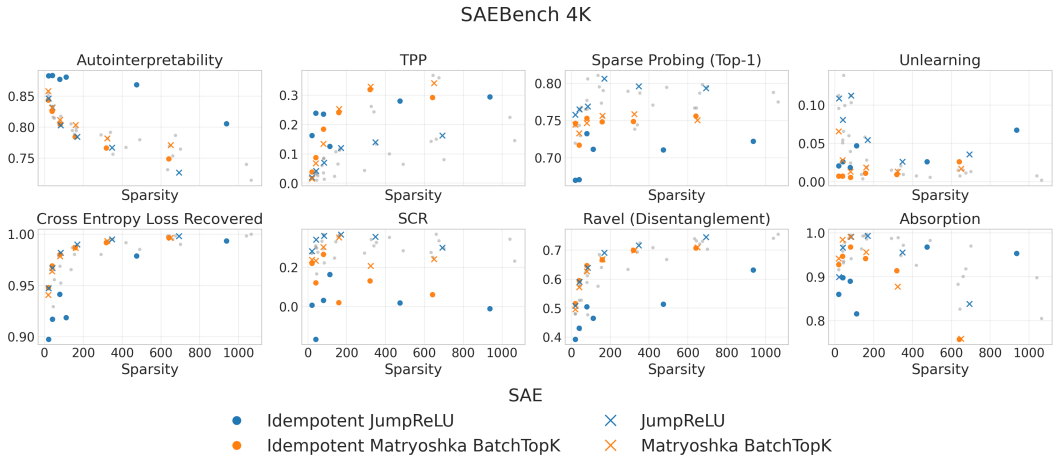


Figure 57: Comparison between the 4K-width SAEs trained in this work and the corresponding SAEBench baselines. Gray points show the full set of SAEs released in SAEBench and are de-emphasized to improve readability, while the highlighted points enable a direct comparison between matched SAE variants. As in Figure 5, the x-axis ("Sparsity") shows the ℓ_0 of each SAE.

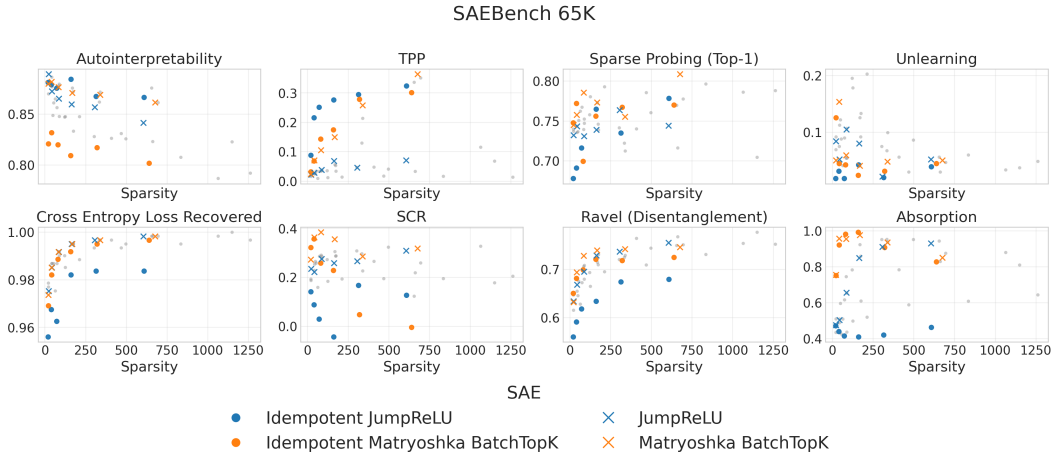


Figure 58: Comparison between the 65K-width SAEs trained in this work and the corresponding SAEBench baselines. Gray points show the full set of SAEs released in SAEBench and are de-emphasized to improve readability, while the highlighted points enable a direct comparison between matched SAE variants. As in Figure 5, the x-axis ("Sparsity") shows the ℓ_0 of each SAE.

G.5.1 SAEBench Results by Individual SAE Classes

The data shown in Figures 59–61 corresponds to the same evaluation as in Figure 57–58 and Figure 5. Above, we highlight only the newly trained SAEs and their SAEBench counterparts trained without the auxiliary loss, in order to improve readability. For completeness, we additionally provide versions in which all SAE classes from SAEBench are highlighted, enabling comparison against specific alternative SAE types.

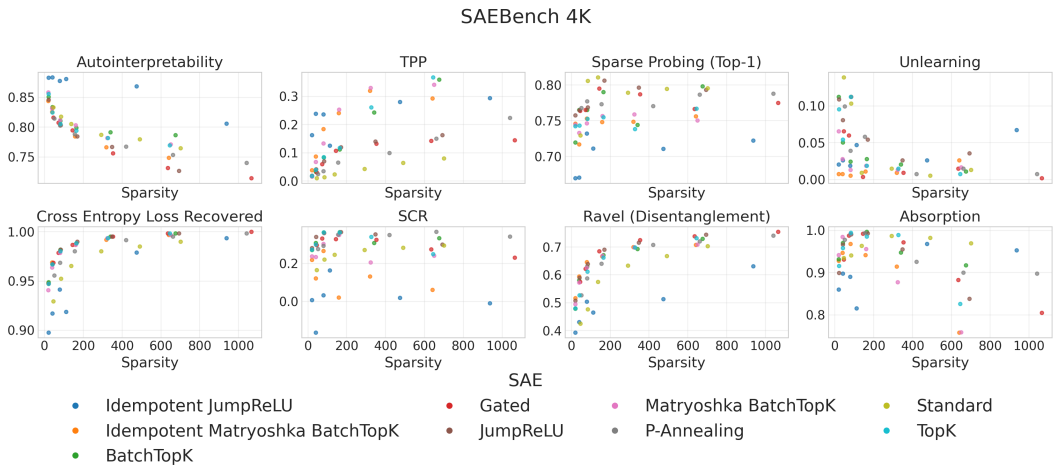


Figure 59: Expanded visualization of the data shown in Figure 57, with all SAE variants highlighted. The plot compares the 4K-width SAEs trained in this work against the full set of SAEs released in SAEBench. As in Figure 5, the x-axis ("Sparsity") shows the ℓ_0 of each SAE.

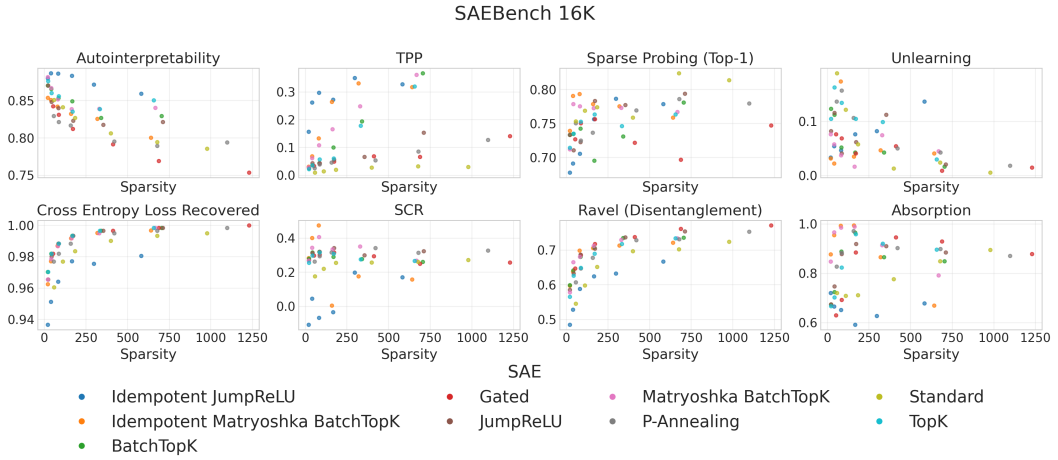


Figure 60: Expanded visualization of the data shown in Figure 5, with all SAE variants highlighted. The plot compares the 16K-width SAEs trained in this work against the full set of SAEs released in SAEBench. As in Figure 5, the x-axis ("Sparsity") shows the ℓ_0 of each SAE.

G.6 Outliers analysis in SAEBench

We also analyzed whether the outlier SAEs that performed well under Experiment 4 show better performance in SAEBench. No such connection could be established. Figure 62 highlights the SAEs that were classified as outlier type six in Appendix E.

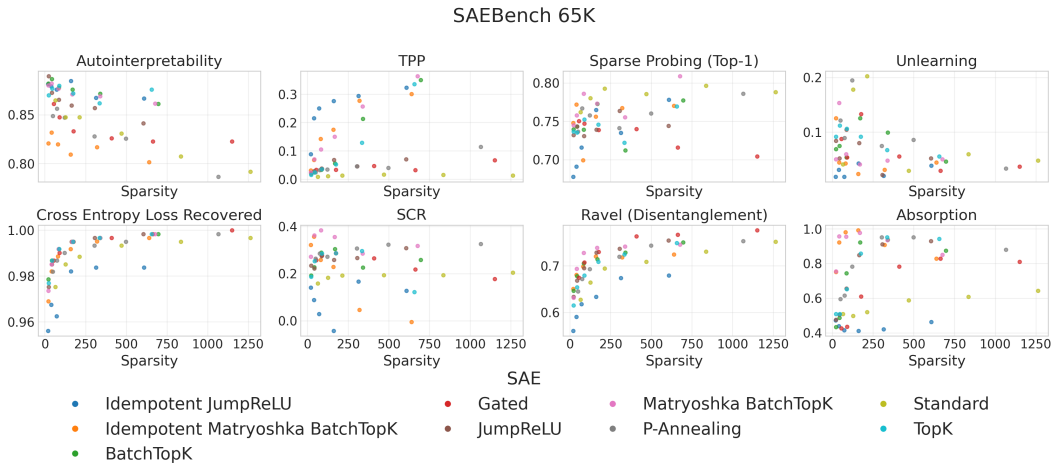


Figure 61: Expanded visualization of the data shown in Figure 58, with all SAE variants highlighted. The plot compares the 65K-width SAEs trained in this work against the full set of SAEs released in SAEBench. As in Figure 5, the x-axis ("Sparsity") shows the ℓ_0 of each SAE.

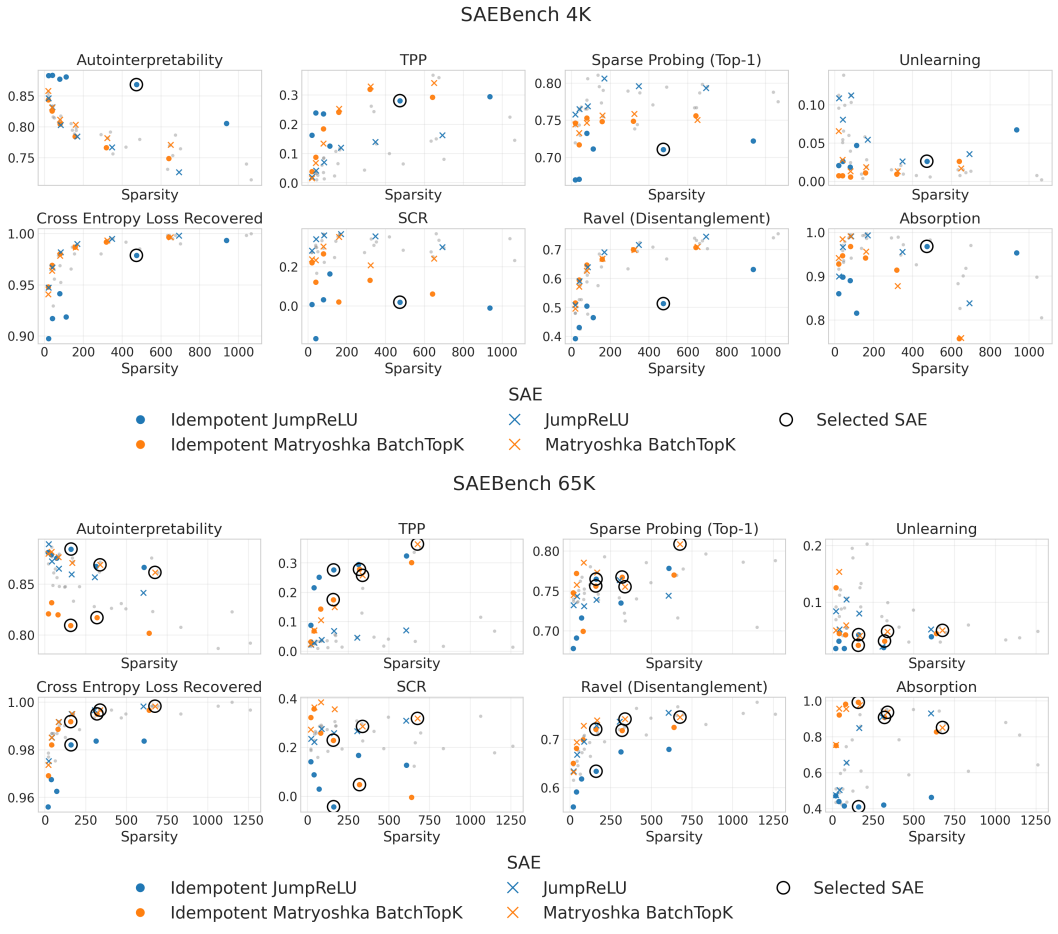


Figure 62: Expanded visualization of the data shown in Figure 57 and Figure 58, with all SAEs highlighted that the outlier analysis in Appendix E identified as idempotent on single-latent inputs. As in Figure 5, the x-axis ("Sparsity") shows the ℓ_0 of each SAE.

H Compute Resources

All experiments were performed on a node of four Nvidia RTX3090s. Experiments 1–4 consumed approximately one, two, one, and three days of compute respectively. Training and evaluating the idempotent SAEs in Experiment 5 required a further six days, with an estimated three to five days for preliminary experiments.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the paper's scope. We propose idempotency as a self-consistency criterion for sparse autoencoders (SAEs), show that current SAEs typically violate this criterion, and evaluate the degree of degradation in empirical studies. We additionally train SAEs with an Intersection-based auxiliary loss to improve idempotency and compare their behavior on our metrics and against standard SAE baselines.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses the limitations of our work, including the experimental scope, methodological assumptions, and the extent to which our findings generalize beyond the evaluated models, SAE releases, and datasets.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: Our main contributions are empirical. The paper does not present formal theoretical results in the form of theorems requiring complete proofs. We introduce the relevant conceptual assumptions in Section 2 and use them to motivate idempotency as a self-consistency criterion in Section 3 and Appendix A. These discussions are intended to clarify the assumptions and interpret the empirical findings.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose the information needed to reproduce the main experimental results, including formal definitions for all investigated metrics in Section 4.1, the evaluated SAEs in Section 4.2, hardware resources in Appendix H and hyperparameter settings during training in Appendix F. The experimental setups and protocols are described in their respective sections.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All experimental code is publicly available at `https://github.com/aip-hd-research/sae_idempotency`. The externally used SAEs and models are publicly available, and we additionally release the SAEs trained in this work, linked from the repository.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The configurations and protocols for the experiments are lined out in their respective section. We provide the relevant details of the SAE training procedure in Appendix F, including the training objective, auxiliary intersection loss, used data, hyperparameters, and evaluation criteria.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For Experiments 1–4 (Sections 4.2–4.4), we report aggregate metrics with error bars corresponding to one standard deviation across the evaluated models and SAE variants, and specify the corresponding aggregation procedure. For the SAEBench results (Section 4.5 and Appendix G) we plot all available datapoints as a grayed-out point cloud to given an idea of the overall variance. The outlier analyses in Appendices D and E focus on individual SAEs. These figures are therefore reported without error bars, since variance statistics over metric scores of individual latents were not collected separately.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   We describe the computational infrastructure and the approximate compute time required to run our experiments in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: We acknowledge and comply with the NeurIPS Code of Ethics. To the best of our assessment, this work does not involve human subjects, crowdsourcing, personally identifiable information, surveillance data, or safety-critical deployment, and we document the relevant models, SAEs, datasets, licenses, compute resources, and limitations.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper does not provide a separate discussion of positive and negative societal impacts. Our contribution is a self-consistency criterion for SAEs intended to support future SAE evaluation and thereby improve the reliability of mechanistic interpretability methods. The implications of deviations from this self-consistency requirement are discussed throughout the main text. However, idempotency alone likely does not solve the broader challenges of making the internal computations of transformer models transparent, so a detailed societal-impact discussion of transparent transformers would be overstated relative to the scope of the work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work does not release a high-risk pretrained model, image generator, scraped dataset, or safety-critical deployment artifact. The pretrained models used in our experiments are existing external resources, potentially subject to their own access

restrictions and licenses, while the artifacts released by this work consist of diagnostic code and SAE-related experimental results intended for reproducibility, as well as the trained SAEs used for analysis. These SAEs are not standalone generative models and do not provide new model capabilities beyond the underlying pretrained models. Rather, they are auxiliary representations of internal activations intended for interpretability. We therefore do not consider them to pose the type of misuse risk targeted by this question.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credit the creators and original owners of all assets used in this work, including prior literature, externally released SAEs, pretrained models, and datasets. The experimental code was written by us. Existing codebases that we modified to incorporate the auxiliary SAE training loss and to evaluate the resulting SAEs are explicitly referenced and used in accordance with their respective terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new assets introduced by this work consist of the experimental code, modified SAE-training code, training and evaluation configurations, and the SAEs trained with the auxiliary intersection loss. The code, together with documentation, reproduction instructions, and links to the trained SAEs, is available at `https://github.com/aip-hd-research/sae_idempotency`.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments, paid annotation work, or research with human subjects. All experiments are conducted on existing models, SAE releases, and publicly available data.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, direct interaction with human participants, or human-subject data collection. Consequently, no study participants incurred risks and IRB approval or equivalent review was not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

Justification: The core method development does not use LLMs as an original or non-standard methodological component. Pretrained language models are used as experimental subjects and are documented in the experimental setup. Any LLM assistance was limited to writing, editing, formatting, or debugging support and did not affect the scientific methodology, results, or claims. Therefore, no additional LLM usage declaration is required in the main text or appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.