Reconstruction Converges Before Decoder Reproducibility in Sparse Autoencoders

Anonymous

Abstract

Sparse autoencoders (SAEs) are often selected by reconstruction loss, but interpretability also requires stability across random seeds and, when ground truth is available, correctness. We present a null-calibrated multi-seed study of SAE decoder reproducibility on mod-113 transformers and selected layers of Pythia-70M/160M/410M. In a compute-limited cached-activation regime, reconstruction and decoder reproducibility separate sharply: TopK, ReLU, JumpReLU, and Batch-TopK SAEs reduce reconstruction loss while decoder-column similarity remains within 1.5% of the geometric random-dictionary null. With longer training, decoder metrics rise well above this null. On Pythia-410M layer 21, a 5-seed 500-epoch TopK audit reaches decoder MMCS 0.521 ($4.85\times$ random), with assignment-based decoder, activation-pattern, and firing-overlap metrics also above random baselines. Held-out audits on mod-113 and Pythia-410M show a consistent pattern: decoder matching rises strongly, while activation-pattern and firing-overlap agreement rise later and more weakly. A selected-layer streaming control partially tests cache-repetition effects. We also identify a failure mode: dead-neuron resampling can produce transient *phantom consistency*, so stability metrics should not be read as feature correctness by themselves. On a synthetic sparse-superposition benchmark with known ground truth, MMCS and GT-MCC rise together in one recoverable linear regime (correlation 0.997), providing a positive control for planted decoder-column recovery. These results argue for reporting reconstruction, nearest-neighbor decoder similarity, assignment-based matching, activation-level agreement, and ground-truth or downstream metrics together.

1 Introduction

Sparse Autoencoders aim to decompose neural network activations into human-interpretable features [Templeton et al., 2024, Gao et al., 2024]. The premise is that SAEs uncover "the true underlying features used by a model" [Elhage et al., 2022]. But do SAE features generalize across training runs, or are they artifacts of random initialization?

If features vary strongly with random seeds, interpretations based on single SAE instances may be incomplete: the features discovered could be one of many useful decompositions, with no claim to uniqueness. This concern was validated by Paulo and Belrose [2025], who found that only 30% of features are shared across independently trained SAEs on large language models, and by Song et al. [2025], who argued that feature consistency should be a primary evaluation criterion. Heap et al. [2025] showed that SAEs trained on randomly initialized transformers produce similar interpretability scores to those from trained models. Leask et al. [2025] demonstrated through stitching experiments that SAE features are neither complete nor atomic. Together, these findings raise a basic question: when do SAE latents reflect reproducible model structure?

Preprint.

We present a multi-seed stability analysis with null-calibrated controls on an algorithmic transformer (mod-113) and selected layers of Pythia-70M/160M/410M [Biderman et al., 2023]. Our aim is to reconcile, not overturn, recent stability results. Paulo and Belrose emphasize seed dependence at moderate budgets, while Song et al. show that stronger consistency is achievable with different metrics and longer or more strongly tuned training. We study the regime between these endpoints. Our central claim is narrower than "SAEs are unstable": in a compute-limited cached-activation regime, reconstruction can appear converged while decoder-column similarity remains near a calibrated random null for much longer. Our contributions are:

1. In a compute-limited cached-activation regime, reconstruction loss and decoder reproducibility occur on different timescales: four SAE architectures reduce reconstruction loss while MMCS remains near a calibrated random-dictionary null.
2. We make the metric scope explicit by reporting nearest-neighbor decoder MMCS alongside corrected assignment-based matching and held-out activation/firing/RSA diagnostics, including a Pythia-410M layer-21 audit.
3. We identify a stability-metric failure mode: cached dead-neuron resampling creates transient *phantom consistency* that later decays below the unregularized baseline.
4. We provide supporting calibration: longer training and selected Pythia scope checks raise decoder reproducibility, and a synthetic ground-truth positive control shows MMCS and GT-MCC rising together in one recoverable linear regime.

2 Related Work

SAEs for interpretability. SAEs are widely used for mechanistic interpretability [Cunningham et al., 2023, Bricken et al., 2023], with scaling to GPT-4 [Gao et al., 2024] and Claude 3 Sonnet [Templeton et al., 2024]. Two major architectural variants dominate: ReLU SAEs with L1 sparsity [Bricken et al., 2023] and TopK SAEs with hard sparsity [Gao et al., 2024], with subsequent innovations including JumpReLU [Rajamanoharan et al., 2024] and BatchTopK [Bussmann et al., 2024a].

Feature stability. Paulo and Belrose [2025] trained 9 SAEs with different random seeds on Pythia-160M and Llama 3 8B, finding that only 30–42% of features are consistently recovered. Song et al. [2025] elevated this to a position paper, arguing that mechanistic interpretability should prioritize feature consistency and demonstrating that high assignment-based PW-MCC is achievable with TopK SAEs and sufficient training on Pythia-160M.

SAE pathologies. O'Neill et al. [2024] study within-run amortized sparse inference and identify a gap where reconstruction can appear converged while the encoder remains suboptimal; this provides a plausible mechanism for, but is distinct from, our across-run decoder-reproducibility analysis. Bussmann et al. [2024b] documented feature absorption, where SAEs collapse semantically distinct features into shared neurons. Our work complements this line by showing that even when features are not absorbed, they may not be reproducible across seeds.

Evaluation and baselines. Korznikov et al. [2026] independently advocate random-baseline controls for SAE evaluation. Chanin and Garriga-Alonso [2026] introduce SynthSAEBench, demonstrating that reconstruction quality does not reliably predict feature quality. SAEBench [Karvonen et al., 2025] standardizes broader SAE evaluation, but does not make cross-seed reproducibility a primary metric; we add that axis. Jedryszek and Crook [2026] show that weight regularization during SAE training improves feature stability, offering a complementary approach to the longer-training mitigation we evaluate.

Identifiability theory. Cui et al. [2026] derive conditions for SAE identifiability: extreme sparsity of ground truth, sparse SAE activation, and sufficient hidden dimensions. When these conditions are violated, identifiability is not guaranteed and multiple decompositions may fit the same reconstruction objective. Our dense activation setting ($\text{erank}/d_{\text{model}} = 62.5\%$) places us outside the extreme-sparsity regime they analyze; since true generative factors are unknown for mod-113/Pythia activations, this is a heuristic consistency argument rather than a formal application of their theorem.

Sparse coding. Olshausen and Field [1996] observed that sparse codes for natural images converge to Gabor-like filters regardless of initialization, but only after extensive training on large datasets. Formal dictionary-learning work gives recovery guarantees only under stronger sparse-coding as-

sumptions than our SAE setting [Spielman et al., 2012, Gribonval and Schnass, 2010, Arora et al., 2015]. Our results suggest SAEs follow a related optimization trajectory, with the added complication that overcomplete neural-activation dictionaries introduce far more symmetries to break.

3 Methods

3.1 Experimental Setup

Algorithmic transformer. We trained a 2-layer transformer (d_{model}=128, n_{heads}=4, d_{mlp}=512) on modular addition (mod 113) to 100% accuracy through grokking [Power et al., 2022, Nanda et al., 2023]. Training used AdamW with learning rate 10^{-3} and weight decay 1.0 for 5,000 epochs on 30% of possible input pairs. We use a 2-layer architecture because it is not the single-layer Fourier-circuit setting analyzed by Nanda et al. [2023]; our stability claims do not rely on recovering that specific circuit.

SAE architectures. We trained four SAE variants on layer-1 residual stream activations:

- **TopK**: Hard sparsity with k=32 active features [Gao et al., 2024]
- **ReLU**: Soft sparsity with L1 penalty λ=10^{-3} [Bricken et al., 2023]
- **JumpReLU**: Learned per-feature thresholds with straight-through estimator [Rajamanoharan et al., 2024]
- **BatchTopK**: Batch-level top-k selection allowing variable per-sample sparsity [Bussmann et al., 2024a]

All used expansion factor $8\times$ (d_{sae}=1,024 for d_{model}=128).

Multi-seed protocol. For each architecture, 5 SAEs were trained with seeds $\{42, 123, 456, 789, 1011\}$, ensuring identical data, hyperparameters, and training duration (30 epochs, $\sim$1,500 gradient steps over 12,769 activation vectors). Only random initialization differed. The 20-seed robustness study (Section 4.1) used an expanded seed set of 20, yielding $\binom{20}{2} = 190$ pairwise comparisons. Mod-113 SAE training used Adam (lr $= 3\times10^{-4}$) with batch size 256.

Language model experiments. We trained SAEs on Pythia-70M, 160M, and 410M [Biderman et al., 2023], extracting 200K activation vectors from selected layers using wikitext-103. The Pythia-160M/410M cross-scale sweeps used $d_{\text{sae}} \in \{512, 1024, 2048, 4096, 8192\}$. The Pythia-70M multilayer scope check used a separate post-residual extraction run with $d_{\text{sae}} \in \{256, 512, 1024, 2048, 4096\}$. Extended training experiments (Section 4.3) used up to 200 epochs with batch size 4,096. Appendix J reports additional selected-layer held-out TopK audits, including Pythia-1B/1.4B layers, with 5 seeds and 100K–200K activation caches reported per row.

3.2 Evaluation Metrics

Mean Maximum Cosine Similarity (MMCS). For each decoder column in SAE_1, we compute its maximum absolute cosine similarity to any column in SAE_2, averaged symmetrically across both directions. MMCS is bounded by 0 and 1 but has a substantial dimension- and dictionary-size-dependent random baseline; all reported MMCS values are interpreted relative to this null. Because MMCS permits many-to-one matches, it measures nearest-neighbor decoder geometry rather than one-to-one feature identity. MMCS above random does not imply similar activations, similar causal effects, semantic correctness, or downstream usefulness.

Assignment-based PW-MCC. Song et al. [2025] define PW-MCC using Hungarian algorithm one-to-one matching, enforcing exclusive feature pairing. We therefore treat MMCS and PW-MCC as complementary metrics: MMCS is sensitive to whether different seeds learn nearby decoder directions even under splitting or redundancy; PW-MCC asks whether the full dictionary admits an exclusive assignment. We do not compare MMCS values directly to PW-MCC thresholds from prior work. Where the claim is specifically about the many-to-one statistic, we call it decoder-column MMCS rather than feature recovery. Because our main architecture, training-dynamics, and cross-scale sweeps are analyzed with MMCS, all main empirical claims in Sections 4.1–4.5 should be read

as claims about decoder-column geometry, not about one-to-one feature identity. Assignment-based metrics remain complementary checks rather than the main object of this paper.

Activation and functional diagnostics. On held-out samples, activation-pattern Hungarian matching computes one-to-one assignments over absolute Pearson correlations between latent activation vectors. Firing-overlap Hungarian uses one-to-one assignments over binary firing Jaccard similarities. RSA reports Spearman correlation between the two latent spaces' held-out sample-similarity matrices. Metric-specific random baselines run the same held-out evaluation on randomly initialized SAEs with the same architecture, seeds, data split, and matching convention; decoder Hungarian treats columns as features whenever decoder matrices share the same ambient row dimension.

Null calibration via extreme value theory. The inner product between two independent random unit vectors in $\mathbb{R}^d$ follows a scaled Beta distribution: $(z{+}1)/2 \sim \text{Beta}((d{-}1)/2,\ (d{-}1)/2)$. For m independent dictionary elements, the expected maximum absolute inner product can be computed by numerical integration over the corresponding order statistic. For $d{=}128,\ m{=}1{,}024$, the idealized prediction is 0.2989 and our observed random baseline is 0.2990. For Pythia-410M ($d{=}1{,}024,\ m{=}1{,}024$), predicted 0.1073 and observed 0.1074. These close matches make the null a useful calibrated reference, not a complete model of learned SAE dictionaries, whose columns may be correlated or anisotropic.

Statistical reporting. Pairwise seed-pair scores are descriptive because each SAE participates in multiple pairs. Our confirmatory analysis therefore uses the 20-seed study with whole-seed resampling of the pairwise MMCS statistic, together with a permutation test over seed labels. We set the practical-equivalence margin at $\varepsilon = 0.005$, a small absolute MMCS change relative to the effect sizes used for interpretability claims in this paper. We report pairwise standard deviations in figures as dispersion summaries, not as independent-sample uncertainty estimates.

Effective rank. Following Roy and Vetterli [2007], we define $\text{erank}(A) = \exp(-\sum_i \bar{\sigma}_i \log \bar{\sigma}_i)$, where $\bar{\sigma}_i = \sigma_i / \sum_j \sigma_j$ are the normalized singular values of the mean-centered activation matrix. This continuous measure equals the matrix rank when all nonzero singular values are equal and approaches 1 when a single component dominates.

Reconstruction metrics. Explained variance: $1 - \|x - \hat{x}\|^2 / \|x\|^2$. L0 sparsity: mean number of active features per input.

4 Results

4.1 Short-Budget Decoder MMCS Remains Practically Near the Random Baseline

At 30 epochs ($\sim$1,500 gradient steps), trained TopK SAE decoder-column MMCS is 0.3040 versus 0.3000 for random dictionaries, an absolute gap of 0.0040 (1.5%). In the 20-seed robustness study, whole-seed resampling of the pairwise statistic preserves the same qualitative conclusion: the gap is statistically detectable (one-sided seed-label permutation $p = 0.0001$; seed-resampled bootstrap 95% CI for the gap $[0.0037, 0.0043]$), as expected with more seeds, but remains within our pre-specified practical-equivalence margin $\varepsilon = 0.005$. Trained SAEs remain extremely close to the geometric null despite low reconstruction loss (Figure 2).

All four architectures remain within 1.5% of the random null (Figure 1):

Table 1: Architecture comparison at 30 epochs on mod-113 (d_{sae}=1,024). All four settings remain near the MMCS random null. The loss column is the final manifest-reported training loss and is not comparable across regularizers; ReLU and JumpReLU are also not sparsity-matched to TopK/BatchTopK, so this table is not an intrinsic architecture-quality comparison.

Architecture	MMCS	Over Random	L0	Loss
TopK	0.304 ± 0.001	$1.015\times$	32 (fixed)	21.3
BatchTopK	0.302 ± 0.001	$1.008\times$	32 (avg)	8.9
ReLU	0.300 ± 0.001	$1.002\times$	$\sim$328	0.31
JumpReLU	0.300 ± 0.001	$1.002\times$	$\sim$365	0.42

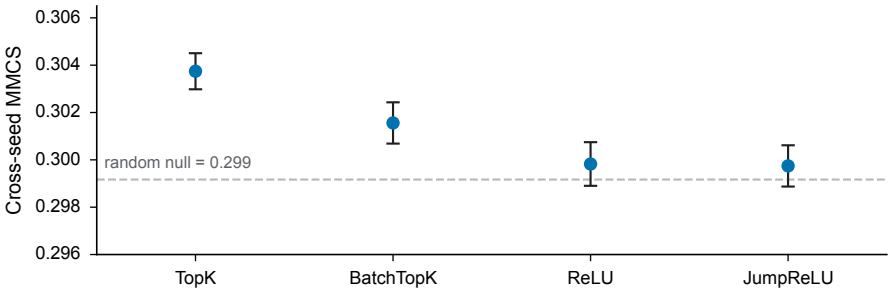

Figure 1: All four SAE settings remain within 1.5% of the MMCS random null at 30 epochs. Error bars show dispersion across pairwise seed comparisons and should not be interpreted as independent-sample confidence intervals.

We do not yet have a committed four-architecture Pythia-70M reproduction, so we restrict the cross-architecture comparison to mod-113. Unlike Paulo and Belrose [2025], who found TopK more unstable than ReLU on LLMs when controlling sparsity, our short-budget mod-113 runs show no practical separation. Because the ReLU-family L0 values here are much higher than TopK k=32, we treat this as evidence for the short-budget null result, not as a sparsity-controlled architecture ranking.

A complementary activation-space sanity check at the same short-budget TopK setting is mixed rather than decisively positive: decoder MMCS is $1.014\times$ random, activation-pattern MMCS is $1.085\times$ random, and representational-similarity analysis is below the random baseline (0.293 vs. 0.363). We therefore treat activation-level agreement as an important future metric, not as evidence that short-budget features have stable identities.

4.2 The Effective Rank Study

We swept d_{sae} from 16 to 1,024 on mod-113 (effective rank $\sim$80). Underparameterized SAEs ($d_{\text{sae}} <$ erank) achieve up to $2.87\times$ random stability but have higher reconstruction loss; overparameterized SAEs ($d_{\text{sae}} \gg$ erank) reduce reconstruction loss but collapse to random stability (full sweep in Appendix C). Section 4.3 shows that this apparent penalty is strongly training-budget dependent.

Our measured MMCS of 0.300 at short training is consistent with the identifiability theory of Cui et al. [2026], who predict low recoverability outside extreme-sparsity regimes. Because mod-113/Pythia activations lack known ground-truth sparse factors, this is only a qualitative consistency check. See Appendix B for a detailed comparison.

4.3 Training Dynamics Reveal Convergence

Extended training to 2,000 epochs on mod-113 reveals architecture-dependent convergence under MMCS (Figure 2). TopK MMCS asymptotes at $\sim$0.73 ($2.44\times$ random), reaching 0.723 by epoch 1,000 and gaining only $+0.009$ over the final 1,000 epochs. ReLU plateaus at $\sim$0.66 ($2.22\times$ random), confirmed from epoch 300 through 2,000. Reconstruction loss saturates within $\sim$20 epochs, but decoder-column MMCS continues climbing for 1,000+ epochs. This timescale separation is our main result.

To check whether this is only an MMCS artifact, we ran a held-out multi-metric TopK audit on an 80/20 split of the same mod-113 activation corpus through 2,000 epochs. At epoch 2,000, held-out explained variance is 0.996; decoder MMCS is 0.724 vs. a 0.300 random baseline, and corrected column-feature Hungarian matching is 0.678 vs. 0.289 random. Activation-pattern Hungarian matching also rises (0.776 vs. 0.547 random), while binary firing overlap rises later and remains weaker in absolute terms (0.540 vs. 0.361 random).

We also ran a functional sanity check using causal feature ablations. For the 24 most active held-out features in a reference SAE at each checkpoint, we matched features in the other four seeds by strict decoder Hungarian matching, ablated each feature's decoded contribution at the layer-1 residual hook, and compared the per-example true-logit drop vector against active-random target features sampled from the target SAE's high-activity features. Matched causal-effect correlations are higher than

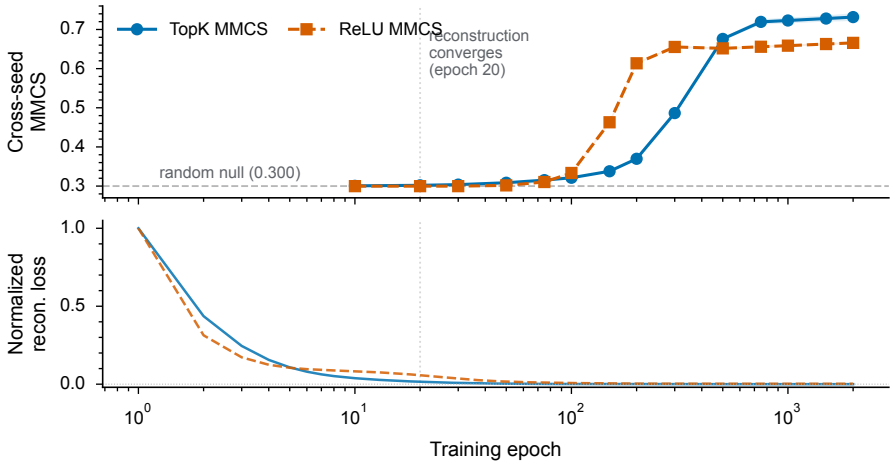

Figure 2: **Reconstruction converges before decoder reproducibility.** Cross-seed MMCS remains near the random null through early training, then continues climbing long after normalized reconstruction loss has converged. TopK asymptotes at $\sim$0.73; ReLU at $\sim$0.66.

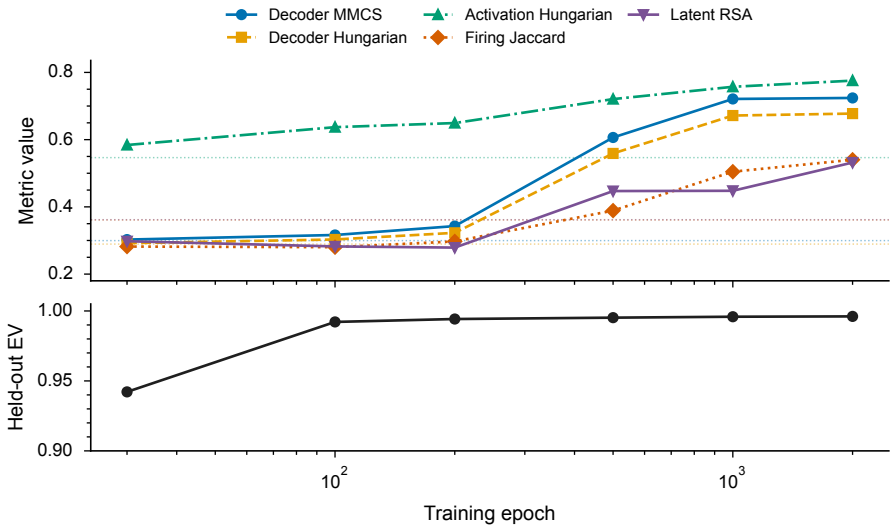

Figure 3: **Held-out multi-metric audit.** TopK SAEs were trained on 80% of mod-113 activations and evaluated on the held-out 20%. Decoder MMCS and corrected column-feature Hungarian matching rise strongly by 2,000 epochs; activation-pattern matching rises more slowly, and binary firing overlap becomes meaningfully above random only late. Dotted lines show metric-specific random SAE baselines.

this active-random baseline but remain modest in absolute terms: mean matched vs. active-random correlations are 0.037 vs. 0.003 at epoch 30, 0.183 vs. 0.001 at epoch 500, and 0.290 vs. -0.006 at epoch 2,000; the epoch-2,000 matched-minus-active-random gap is 0.296 (reference-feature-cluster bootstrap 95% CI [0.203, 0.389]). A 64-feature epoch-2,000 robustness run gives a similar active-random gap of 0.332 (95% CI [0.275, 0.392]). Even at epoch 2,000, the 0.290 matched causal-effect correlation remains an incomplete proxy for causal or functional agreement.

Phantom consistency of resampling. Under the cached dead-neuron resampling protocol we test, resampling produces a transient spike in MMCS (0.888 at epoch 10) that decays monotonically to 0.562 by epoch 2,000—below the unregularized TopK baseline for the remainder of training (Figure 4). Production resampling protocols vary, so this result is a warning about one common design pattern rather than a claim that all resampling is harmful.

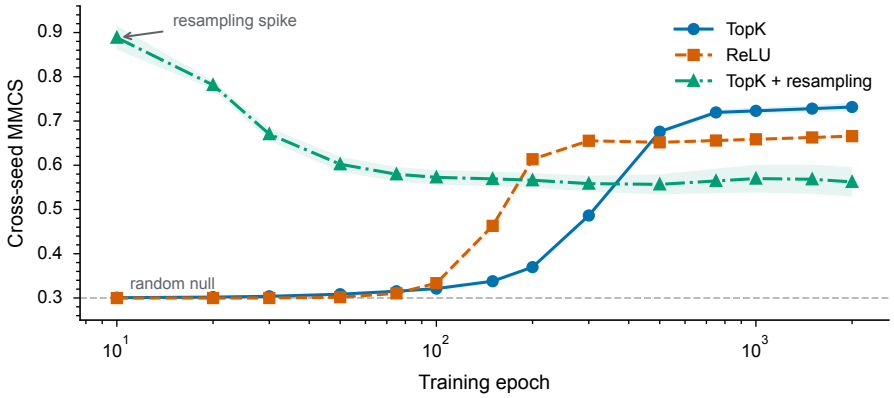

Figure 4: **Phantom consistency.** TopK with resampling (green) spikes to 0.888 at epoch 10, then decays below standard TopK (blue). In this cached run, resampling creates transient decoder agreement but does not provide reliable long-horizon stability.

We tested one simple mechanism: shared resampling pools. In a threshold-matched 50-epoch follow-up, replacing the shared resampling pool with seed-specific resampling pools did not remove the early spike (epoch-10 MMCS 0.524 shared pool, 0.543 seed-specific pool, 0.301 without resampling). Thus shared-pool anchoring alone is not the explanation, and the remaining mechanism is unresolved. The tracked 2,000-epoch cached run also destabilizes reconstruction loss; together with a Pythia streaming-resampling stress test, this shows that resampling can manufacture decoder agreement while worsening reconstruction, so the spike should not be interpreted as useful stability or functional agreement.

The d_{sae}/effective-rank threshold is training-budget sensitive. To test whether overparameterized regimes are fundamentally unstable or merely undertrained, we swept $d_{\text{sae}} \in \{512, 1024, 2048, 4096\}$ on Pythia-410M layer 21 with 200 epochs of extended training (3 seeds each):

Table 2: All d_{sae} values reach 4.3–4.7× random at 200 epochs on Pythia-410M layer 21, despite different overparameterization ratios. This layer-level sweep shows that the short-budget instability penalty can disappear with longer training.

d_{sae}	d_{sae}/erank	10-ep MMCS (ratio)	200-ep MMCS	200-ep ratio
512	1.3	0.248 (2.45×)	0.479	4.73×
1024	2.5	0.230 (2.14×)	0.458	4.27×
2048	5.0	0.213 (1.88×)	0.523	4.63×
4096	10.0	0.183 (1.55×)	0.537	4.54×

The $d_{\text{sae}} = 4{,}096$ condition (ratio 10.0), which showed the weakest 10-epoch stability in this sweep, reaches 4.54× random at 200 epochs—close to the $d_{\text{sae}} = 512$ condition. Thus the threshold should not be interpreted as a fixed property of the optimization landscape without accounting for training budget.

4.4 Selected-Layer Streaming Control for Cache Repetition

To test whether convergence reflects memorization of cached activations (200 epochs over 200K cached samples), we trained Pythia-410M SAEs in two modes: *cached* (repeated fixed-cache protocol, 3 seeds) and *streaming* (fresh activations extracted online from wikitext-103 for each step-equivalent rather than repeated fixed-cache passes, 3 seeds). Streaming MMCS matches cached through 100 step-equivalents (0.313 vs. 0.316; 2.92× vs. 2.94× random). At 200 step-equivalents, cached reaches 0.458 (4.26×) while streaming reaches 0.418 (3.89×; Δ=0.039, or 11% of the cached above-random signal). This reduces but does not eliminate the cache-memorization explanation: streaming SAEs still reach 3.89× random from a 1.0× starting point, while cached training retains a measurable advantage.

For dead-neuron resampling specifically, our resampling-pool control rejects shared-pool anchoring, and a separate Pythia streaming-resampling stress test shows the harder failure mode: resampling can increase decoder consistency while damaging held-out reconstruction. We therefore treat resampling-induced consistency as an artifact risk, not functional agreement.

4.5 Cross-Scale Scope Checks on Pythia

On Pythia-410M layer 21 (200 epochs, 5 seeds), both TopK ($4.33\times$ random, MMCS $= 0.465 \pm 0.016$) and ReLU ($4.41\times$ random, MMCS $= 0.473 \pm 0.005$) are still climbing and continue improving over the observed training horizon. On Pythia-160M layer 10, ReLU reaches $5.11\times$ random (MMCS $= 0.632$). These runs use roughly 40M token-equivalent SAE update exposures over 200K unique cached activations, far below production-scale SAE training [Gao et al., 2024, Templeton et al., 2024]. We use them as qualitative scope checks rather than benchmark-matched frontier validation or direct cross-paper architecture comparisons.

We also ran the corrected held-out multi-metric audit on a fresh Pythia-410M layer-21 TopK run ($d_{\text{sae}} = 1{,}024$, $k = 32$, 5 seeds, 200K Wikitext token activations with an 80/20 split). At epoch 500, held-out explained variance is 0.947; decoder MMCS is 0.521 vs. 0.107 random, and corrected decoder Hungarian matching is 0.453 vs. 0.104 random. Activation-pattern and firing-overlap Hungarian scores also rise above their random baselines (0.560 vs. 0.166; 0.618 vs. 0.091), though this selected-layer audit still does not establish causal or semantic feature identity. Appendix J gives additional completed selected-layer Pythia held-out audits through Pythia-1.4B; they are scope checks, not a harmonized scaling law.

The TopK-ReLU divergence on Pythia-160M (TopK $2.09\times$ vs. ReLU $5.11\times$) contrasts with their convergence on Pythia-410M (4.33 vs. $4.41\times$). One possible explanation: TopK's hard sparsity constraint creates more local optima in lower-dimensional spaces ($d_{\text{model}} = 768$), causing earlier plateaus, while ReLU's soft sparsity allows gradual redistribution of activation mass. We cannot rule out that TopK on Pythia-160M would resume climbing with more training.

The short-budget overparameterization-stability pattern appears across all three model scales we tested (Appendix H provides a detailed multilayer analysis on Pythia-70M).

Two patterns emerge (Table 11, Appendix I). First, deeper layers with lower effective rank show higher stability, consistent with more structured representations being easier to consistently decompose. Second, absolute MMCS remains low with short training: even at the most favorable setting (Pythia-410M, layer 21, $d_{\text{sae}} = 512$), MMCS is only 0.248 at 10 epochs.

4.6 Subspace Stability

The top-8 principal decoder subspace is $2.98\times$ more stable than random (overlap 0.187 vs. 0.063), even when individual features match random baseline. This ratio decays with subspace rank: $2.03\times$ at rank 16, $1.43\times$ at rank 32, and $1.11\times$ at rank 64. SAEs learn a partially shared low-rank subspace, but the absolute overlap remains modest (0.187 at rank 8), so this should be read as a weak structural signal rather than a stable feature-level basis. Whether subspace-level interpretability is more robust remains a hypothesis for future work.

4.7 Synthetic Ground-Truth Validation

To calibrate whether cross-seed consistency (MMCS) can track actual feature recovery in a setting where ground truth is available, we constructed a synthetic sparse-superposition benchmark with known features. We generated 50,000 samples from $x = W_{\text{gt}} z + \epsilon$ where $W_{\text{gt}} \in \mathbb{R}^{128\times 256}$ is a fixed random unit-norm dictionary, z is 5%-sparse with positive coefficients, and $\epsilon \sim \mathcal{N}(0, 0.01^2 I)$. We trained TopK SAEs ($d_{\text{sae}}$=256, k=12, 5 seeds) and measured both MMCS (cross-seed) and GT-MCC (ground-truth recovery via Hungarian matching against W_{gt}) at 8 checkpoints from 5 to 1,000 epochs.

The result is a positive control: **MMCS and GT-MCC rise together**. By epoch 1,000, MMCS reaches $3.21\times$ random and GT-MCC reaches $3.49\times$ random; the Pearson correlation across training horizons is 0.997. Thus, in this specific recoverable linear regime, MMCS rises in step with GT-MCC-measured recovery of planted decoder columns. Because assignment-based GT-MCC is sensitive to decoder-axis conventions, Appendix E records the metric-layout audit used to validate

this result, including a permutation sanity test that scores a column-permuted overcomplete dictionary at GT-MCC $= 1.0$.

Epoch	MMCS	GT-MCC	MMCS/rand	GT/rand
5	0.334	0.424	1.26×	1.67×
20	0.583	0.680	2.20×	2.68×
100	0.839	0.880	3.16×	3.46×
1000	0.852	0.887	3.21×	3.49×

Table 3: Synthetic benchmark positive control: MMCS (cross-seed consistency) and GT-MCC-measured recovery of planted decoder columns rise together in a recoverable linear sparse-superposition setting. Random baselines are MMCS $= 0.2655$ and GT-MCC $= 0.2539$.

This section is a calibration check, not an external validation of feature correctness on language-model activations. It validates the GT-MCC pipeline and shows that MMCS can be informative when the planted decoder columns are recoverable by the SAE objective. It does not prove that MMCS establishes correctness on real model activations, where ground truth is unknown and learned decoder geometry may agree for reasons unrelated to semantic or causal feature identity.

5 Discussion

We observe a dissociation between reconstruction convergence and decoder reproducibility: reconstruction loss alone does not determine decoder-column geometry on the training horizons we study. This is consistent with, but does not prove, a multiplicity of near-equivalent sparse reconstructions in dense regimes. Our "extended" Pythia runs are far below frontier budgets, so we treat them as selected-layer scope checks rather than frontier validation.

Dead neuron resampling can undermine long-horizon stability under the cached protocol we test, but our resampling-pool follow-up did not support the simplest shared-pool mechanism. The $d_{\text{sae}}/\text{erank}$ instability threshold is also reduced with extended training, so short-budget thresholds should be treated as budget-dependent rather than fixed laws.

Consistency is not correctness by itself. Our synthetic benchmark (Section 4.7) shows that MMCS can track GT-MCC-measured recovery of planted decoder columns in one controlled setting, but on real model activations high consistency confirms decoder-geometry agreement, not semantic, causal, or interventional correctness. This boundary is visible in our mod-113 causal audit: decoder-Hungarian matches at epoch 2,000 have only modest causal-effect correlation (0.290), so geometric and functional alignment can separate substantially. Practitioners should report cross-seed stability alongside downstream, causal, auto-interpretability, or ground-truth metrics.

Implications for safety. Seed sensitivity creates an auditability risk for safety-motivated interpretability: conclusions from one SAE may fail to transfer across runs. We do not measure deception detection directly; this is a caution about evidentiary standards, not a demonstrated failure of a specific safety audit.

6 Limitations

Scale and training budget. Our primary experiments use $d_{\text{model}}{=}128$; Pythia audits are selected-layer checks through 1B/1.4B, below frontier scale and not a harmonized scaling law. Pythia asymptotes are unknown.

Metric. MMCS is many-to-one decoder geometry; Hungarian matching is exclusive but has a substantial random baseline in overcomplete dictionaries. Both are geometric proxies and can miss activation-level or causal agreement. Our held-out and causal diagnostics are limited; we do not include Pythia causal, semantic, human-label, or downstream-task validation.

Task dependence. Dense mod-113 activations may exaggerate instability. Our Pythia streaming control addresses repeated-cache exposure during training, not generalization to new activation distributions or layers.

References

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Simple, efficient, and neural algorithms for sparse coding. In *Conference on Learning Theory*, pages 113–149, 2015.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, 2023.

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

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

Bart Bussmann et al. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *NeurIPS*, 2024b.

David Chanin and Adrià Garriga-Alonso. SynthSAEBench: Evaluating sparse autoencoders on scalable realistic synthetic data. *arXiv preprint arXiv:2602.14687*, 2026.

Jingyi Cui, Qi Zhang, Yifei Wang, and Yisen Wang. On the limits of sparse autoencoders: A theoretical framework and reweighted remedy. *ICLR*, 2026.

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

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

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

Rémi Gribonval and Karin Schnass. Dictionary identification: Sparse matrix-factorization via ℓ_1-minimization. *IEEE Transactions on Information Theory*, 56(7):3523–3539, 2010.

Thomas Heap, Thomas Lawson, Lukas Farnik, and Laurence Aitchison. Sparse autoencoders can interpret randomly initialized transformers. *arXiv preprint arXiv:2501.17727*, 2025.

Patryk Jedryszek and Oliver M Crook. Stable and steerable SAEs with weight regularization. *arXiv preprint arXiv:2603.04198*, 2026.

Adam Karvonen et al. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*, 2025.

Artem Korznikov et al. Sanity checks for SAEs: Do SAEs beat random baselines? *arXiv preprint arXiv:2602.14111*, 2026.

Patrick Leask, Bart Bussmann, et al. Sparse autoencoders do not find canonical units of analysis. *ICLR*, 2025.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *ICLR*, 2023.

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

Charles O'Neill et al. Compute optimal inference and provable amortisation gap in sparse autoencoders. *arXiv preprint arXiv:2411.13117*, 2024.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with JumpReLU sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference (EUSIPCO)*, pages 606–610, 2007.

Xiangchen Song, Aashiq Muhamed, Yujia Zheng, Lingjing Kong, Zeyu Tang, Mona T Diab, Virginia Smith, and Kun Zhang. Position: Mechanistic interpretability should prioritize feature consistency in sparse autoencoders. *arXiv preprint arXiv:2505.20254*, 2025.

Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. *arXiv preprint arXiv:1206.5882*, 2012.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

A Dictionary Pinning

Freezing decoder columns from a reference SAE achieves high stability but at steep reconstruction cost:

Fraction Pinned	MMCS	Recon Loss	Recon Loss Δ
0%	0.304	21.3	–
6.25% (64 cols)	0.347	20.0	-6%
25% (256 cols)	0.478	60.3	$+183\%$
50% (512 cols)	0.653	123.4	$+479\%$
75% (768 cols)	0.828	231.1	$+985\%$
100% (1024 cols)	1.000	4210.4	–

Table 4: Dictionary pinning. At low fractions (6.25%), stability improves with no reconstruction cost; above 25%, reconstruction degrades sharply.

B Theoretical Grounding

Our empirical MMCS values are consistent with the identifiability framework of Cui et al. [2026]. Their analysis indicates that SAEs fail to recover ground-truth features unless: (1) the ground truth is extremely sparse, (2) SAE activation is sparse, and (3) hidden dimensions are sufficient. Our dense activation subspace (62.5% of d_{model}) violates Condition 1. Extended training does not resolve non-identifiability in the formal sense; rather, it suggests that optimization can select increasingly similar solutions within a potentially non-identifiable family.

C Effective Rank Sweep

D Experimental Details

The released environment pins PyTorch 2.5.1; some H100 Pythia scope-check manifests record torch 2.4.1+cu124. Individual manifests record CPU/CUDA device, and mixed precision was not uniformly enabled or recorded. Mod-113 SAE experiments used Adam (lr $= 3\times10^{-4}$) with batch size 256. Pythia extended/sweep experiments used AdamW (lr $= 3\times10^{-4}$) with batch size 4,096 unless otherwise specified in the manifest. Main Pythia dsae-sweep runs covered 200 epochs; held-out Pythia multi-metric scope checks used 500 or 1,000 epochs as reported in the appendix tables. The

Table 5: Stability-reconstruction tradeoff as a function of d_{sae} on mod-113 (30 epochs).

Regime	d_{sae}	k	MMCS	Ratio	Recon Loss
Under	16	4	0.513	2.87×	1.124
Under	32	8	0.454	2.22×	0.587
Matched	64	16	0.373	1.62×	0.203
Matched	128	32	0.304	1.23×	0.052
Over	512	32	0.295	1.04×	0.028
Over	1024	32	0.304	1.02×	0.034

mod-113 experiments ($d_{\text{model}} = 128$, 2,000 epochs, 5 seeds) complete in minutes on a single GPU. The Pythia-410M experiments ($d_{\text{model}} = 1024$, 200 epochs over 200K cached activations) require approximately 1–3 GPU-hours per seed on an NVIDIA B200. Full hyperparameter tables and JSON manifests are available in the anonymous supplement.

E Synthetic GT-MCC Metric Audit

The synthetic benchmark uses decoder matrices with shape $[d_{\text{model}}, d_{\text{sae}}] = [128, 256]$, where columns are feature directions. Assignment-based GT-MCC is therefore computed by Hungarian matching over decoder columns against the columns of W_{gt}. This convention matters: a legacy shape heuristic that transposed overcomplete matrices would instead treat the 128 rows as features, artificially driving GT-MCC to the random baseline even for permuted copies of a recoverable dictionary. We fixed the metric to default to columns as features whenever both matrices share the same ambient row dimension and added a regression test: a permuted overcomplete dictionary now scores GT-MCC $= 1.0$. Under this corrected convention, the random GT-MCC baseline is 0.2539 and epoch-1000 GT-MCC is 0.8869.

F Held-Out Multi-Metric Check

Table 6: Held-out TopK multi-metric dynamics on an 80/20 mod-113 activation split. Cells report absolute metric value with ratio to the metric-specific random baseline in parentheses. Hungarian columns use the corrected column-feature convention from Appendix E.

Epoch	Held-out EV	Dec. MMCS	Dec. Hung.	Act. Hung.	Fire Hung.	RSA
Random	–	0.300	0.289	0.547	0.361	0.362
30	0.942	0.303 (1.01×)	0.292 (1.01×)	0.584 (1.07×)	0.282 (0.78×)	0.298 (0.82×)
100	0.992	0.316 (1.06×)	0.303 (1.05×)	0.637 (1.17×)	0.280 (0.78×)	0.282 (0.78×)
200	0.994	0.343 (1.14×)	0.323 (1.12×)	0.649 (1.19×)	0.297 (0.82×)	0.279 (0.77×)
500	0.995	0.606 (2.02×)	0.559 (1.93×)	0.721 (1.32×)	0.389 (1.08×)	0.447 (1.24×)
1000	0.996	0.721 (2.41×)	0.672 (2.32×)	0.757 (1.39×)	0.505 (1.40×)	0.448 (1.24×)
2000	0.996	0.724 (2.42×)	0.678 (2.34×)	0.776 (1.42×)	0.540 (1.50×)	0.532 (1.47×)

We repeated the same audit on a fresh Pythia-410M layer-21 TopK run using 200K Wikitext token activations, a deterministic 80/20 split, $d_{\text{sae}} = 1{,}024$, $k = 32$, AdamW, and 5 seeds. Padding tokens were excluded during HuggingFace activation extraction, yielding effective rank 438.9 for this fresh cache. RSA has a high random-SAE baseline on this activation distribution (0.829), so its absolute gap should be interpreted conservatively.

Table 7: Held-out Pythia-410M layer-21 TopK multi-metric dynamics. Cells report absolute metric value with ratio to the metric-specific random SAE baseline in parentheses. This is a limited language-model scope check, not frontier-scale or causal validation.

Epoch	Held-out EV	Dec. MMCS	Dec. Hung.	Act. Hung.	Fire Hung.	RSA
Random	–	0.107	0.104	0.166	0.091	0.829
30	0.854	0.282 (2.62×)	0.240 (2.32×)	0.105 (0.63×)	0.057 (0.63×)	0.961 (1.16×)
100	0.956	0.315 (2.94×)	0.269 (2.60×)	0.145 (0.88×)	0.072 (0.79×)	0.960 (1.16×)
200	0.964	0.410 (3.82×)	0.356 (3.44×)	0.284 (1.71×)	0.126 (1.38×)	0.969 (1.17×)
500	0.947	0.521 (4.85×)	0.453 (4.37×)	0.560 (3.38×)	0.618 (6.81×)	0.957 (1.15×)

G Resampling-Pool Follow-Up

To stress-test the shared-cache explanation for phantom consistency, we ran a threshold-matched follow-up that kept the same training cache and resampling schedule but changed the resampling pool. In condition A, all seeds resampled dead features from the full shared cache. In condition B, each seed resampled from its own fixed 50% subset of the cache while still training on the full cache. The spike persisted, so this control rejects the narrow shared-pool mechanism rather than confirming it.

Table 8: Resampling-pool follow-up on mod-113 TopK SAEs ($d_{\text{sae}} = 1024$, $k = 32$, 5 seeds, dead threshold 50). Values are absolute decoder MMCS. Seed-specific resampling pools do not remove the early spike, so shared-pool anchoring alone is insufficient.

Epoch	Shared resampling pool	Seed-specific resampling pools	No resampling
5	0.327	0.325	0.300
10	0.524	0.543	0.301
20	0.381	0.447	0.302
50	0.350	0.418	0.309

As a separate stress test, we reran the resampling question on Pythia-70M layer 4 with online activation extraction. This was not tuned as a production recipe; it is a failure-mode check for whether resampling-induced consistency should be trusted.

Table 9: Pythia streaming-resampling stress test at epoch 50 ($d_{\text{sae}} = 1024$, $k = 32$, 3 seeds). The large negative explained-variance values indicate catastrophic optimization instability under this online resampling stress test, not a usable reconstruction tradeoff; explained variance is unbounded below once reconstruction error greatly exceeds held-out activation variance. Resampling can nevertheless increase decoder MMCS, so resampling-induced consistency is not evidence of functional agreement.

Condition	Held-out EV	Decoder MMCS	Decoder Hung.	Resampled at ep. 50
Cached, no resampling	0.671	0.260 ($1.72\times$)	0.228	0.0
Cached, resampling	-1.98×10^5	0.264 ($1.75\times$)	0.228	20.0
Streaming, no resampling	0.667	0.235 ($1.55\times$)	0.208	0.0
Streaming, resampling	-5.94×10^6	0.322 ($2.13\times$)	0.188	152.7

H Multilayer Analysis on Pythia-70M

We swept layers $\{0, 2, 4, 5\}$ of Pythia-70M (6 layers total, $d_{\text{model}} = 512$) with $d_{\text{sae}} \in \{256, 512, 1024, 2048, 4096\}$:

Table 10: Pythia-70M multilayer analysis at 10 epochs. Deeper layers with lower effective rank show higher stability at small d_{sae}. At the largest dictionary tested ($d_{\text{sae}} = 4096$), all layers are near $1.1\times$ random.

Layer	Eff. Rank	Best MMCS/Random	Best d_{sae}	At d_{sae}=4096
0	391.4	$1.60\times$	256	$1.07\times$
2	280.6	$1.77\times$	256	$1.09\times$
4	348.9	$1.57\times$	256	$1.10\times$
5	349.3	$1.71\times$	512	$1.11\times$

The Pythia-70M row above comes from the post-residual multilayer scope check in Appendix H. The Pythia-160M/410M rows come from a separate cross-scale sweep over $d_{\text{sae}} \in \{512, 1024, 2048, 4096, 8192\}$; they are intended as qualitative scope checks, not a single harmonized benchmark suite.

Layer 2 has the lowest effective rank (280.6, 54.8% of d_{model}) and the highest stability ratio at small d_{sae}. At the largest dictionary tested ($d_{\text{sae}} = 4096$), all layers are near $1.1\times$ random—the same high-overcompleteness pattern as mod-113 at short training.

Table 11: Cross-scale scope checks at short training. Deeper layers with lower effective rank show higher stability ratios. Absolute MMCS remains low (< 0.25) across all conditions at 10 epochs.

Model	Layer	d_{model}	Eff. Rank	Best Ratio	At $d/\text{erank}\approx 5$
Pythia-70M	0	512	391.4	$1.60\times$	$1.18\times$
Pythia-160M	1	768	573.0	$1.84\times$	$1.51\times$
Pythia-160M	10	768	511.8	$1.66\times$	$1.31\times$
Pythia-410M	2	1024	757.2	$2.09\times$	$1.77\times$
Pythia-410M	21	1024	408.9	$2.45\times$	$1.88\times$

I Cross-Scale Replication

J Additional Pythia Held-Out Scope Checks

Table 12 reports additional completed selected-layer TopK audits. These runs test whether the Pythia-410M held-out multi-metric pattern is obviously idiosyncratic to one layer/model. They are not a harmonized scaling law, frontier-scale validation, or evidence of semantic or causal feature identity.

Table 12: Selected-layer Pythia held-out multi-metric scope checks across model sizes. All rows use 5 SAE seeds. Acts is the total cached activation count before the 80/20 split; SAE is d_{sae}/k. Metric cells report absolute value / metric-specific random SAE baseline; ratios are omitted to avoid hiding small absolute baselines. Layers, checkpoints, capacities, and budgets differ, so rows should not be read as monotonic scaling-law points. Activation and firing columns are held-out agreement diagnostics, not causal or semantic validation.

Model	Layer	Epoch	Acts	SAE	Eff. Rank	Held-out EV	Dec. MMCS	Dec. Hung.	Act. Hung.	Fire Hung.
Pythia-70M	4	1000	200K	1024/32	369.4	0.679	0.393/0.151	0.328/0.146	0.481/0.134	0.816/0.067
Pythia-160M	10	1000	200K	1024/32	533.3	0.662	0.354/0.124	0.299/0.120	0.516/0.139	0.814/0.077
Pythia-410M	21	1000	200K	1024/32	438.9	0.947	0.608/0.107	0.556/0.104	0.589/0.166	0.678/0.091
Pythia-1B	8	500	100K	2048/32	717.7	0.985	0.381/0.080	0.320/0.077	0.247/0.168	0.084/0.089
Pythia-1B	12	500	100K	2048/32	943.1	0.957	0.547/0.080	0.470/0.077	0.385/0.177	0.134/0.100
Pythia-1.4B	12	500	100K	2048/32	892.4	0.972	0.421/0.080	0.358/0.077	0.278/0.170	0.095/0.084

K 1-Layer vs. 2-Layer Comparison

We compared SAE stability on 1-layer (erank $= 33.5$) vs. 2-layer (erank $= 80.5$) mod-113 transformers. In the matched sweep, both peak at $d_{\text{sae}} = 64$, then decay toward the random baseline at high d_{sae}. At $d_{\text{sae}} = 1{,}024$, both match the random baseline (MMCS ~ 0.30). The lower effective rank of the 1-layer model produces a slightly higher peak ratio ($1.49\times$ vs. $1.44\times$ random), but this appendix result is weak support for an effective-rank explanation rather than a standalone finding.