Are Sparse Autoencoder Benchmarks Reliable?

David Chanin [1 2 3]

Abstract

Sparse autoencoders (SAEs) are a core interpretability tool for large language models, and progress on SAE architectures depends on benchmarks that reliably distinguish better SAEs from worse ones. We audit the SAE quality metrics in SAEBench, the de-facto standard SAE evaluation suite, through three complementary lenses: reseed noise on a fixed SAE, ground-truth correlation on synthetic SAEs, and discriminability across training trajectories. We find that two of these metrics, Targeted Probe Perturbation (TPP) and Spurious Correlation Removal (SCR), fail multiple lenses at their canonical settings and should not be used to evaluate SAEs. The other metrics show higher reseed noise and lower discriminability than the field assumes. The sae-probes variant of k-sparse probing is the most reliable metric we tested, but even sae-probes struggles to separate variants of the same SAE architecture. Our results show the field needs better SAE benchmarks.

1. Introduction

Sparse autoencoders (SAEs) decompose a neural network's activation vector into a sparse sum of feature directions and are a core mechanistic interpretability technique for large language models (LLMs) (Cunningham et al., 2024; Bricken et al., 2023). However, recent work has called into question the quality of SAEs compared to supervised baselines (Kantamneni et al., 2025; Wu et al., 2025), highlighting the need for better SAE architectures.

Benchmarking SAEs is harder than most ML evaluation: the "true features" of an LLM are unknown, so we cannot directly tell how well an SAE has learned these "true features". Instead, the field relies on a number of proxy metrics collated in the SAEBench library (Karvonen et al., 2025). Architectural progress comes from small comparisons (a different sparsity penalty, a change to the activation function, an improved auxiliary loss), each staking its case on these proxy metrics, with subsequent work building on whichever variant won. When the proxies rank SAEs reliably, this loop compounds into real progress. When they do not, effort flows into changes that merely look better due to noise or the proxy metric being faulty, and real improvements go undetected. Auditing the proxies is therefore a prerequisite to trusting the architectures they rank.

While directly auditing proxy metrics is difficult due to not knowing the ground truth "true features" of an LLM, we can still define properties that a good proxy metric should satisfy. These include:

1. **Tracks ground-truth.** The score should be monotone in the SAE's ground-truth quality if ground truth is available (e.g. on synthetic models).
2. **Increases throughout training.** A metric whose score systematically declines through training is responding to something other than quality.
3. **Low reseed noise.** The same SAE evaluated on the same data should produce the same score regardless of any random seed used.
4. **Discriminative.** The metric should distinguish runs by an amount substantially larger than its own per-checkpoint jitter.
5. **Internally consistent.** The metric should rank SAEs the same way regardless of nominally-interchangeable hyperparameters.

In this work, we audit six SAEBench evaluations that target overall SAE quality: sparse probing (Gurnee et al., 2023), the sae-probes variant of sparse probing (Kantamneni et al., 2025), targeted probe perturbation (TPP) (Marks et al., 2025), spurious correlation removal (SCR) (Marks et al., 2025), automated interpretation (Bills et al., 2023), and RAVEL (Huang et al., 2024). The five desiderata above cannot all be tested on a single SAE panel: low reseed noise needs the same SAE evaluated many times, ground-truth tracking needs SAEs whose ground-truth quality is known, and discriminability needs SAEs that vary in quality. We therefore approach the audit through three complementary

[1]University College London [2]MATS [3]Decode Research. Correspondence to: David Chanin <david.chanin.22@ucl.ac.uk>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

lenses, each on a different SAE panel and each isolating a different failure mode:

Reseed noise on a real LLM SAE (§3, desideratum 3). We run each SAEBench evaluation five times with different random seeds on a fixed canonical Gemma Scope SAE (Lieberum et al., 2024) (replicated on three other canonical SAEs across model families in Appendix C). Because the SAE is identical across runs, any score spread is pure metric noise; the resulting per-metric CV spans nearly two orders of magnitude and yields a minimum-reliable-difference threshold for single-seed comparisons.

Validity on synthetic SAEs (§5, desideratum 1). Using SynthSAEBench (Chanin & Garriga-Alonso, 2026), a synthetic model whose activations are sparse linear combinations of a known 16k-feature dictionary, we train a panel of SAEs with computable ground-truth quality and check whether each proxy metric's score correlates with that ground truth. This is the only setting in which we can adjudicate desideratum 1 directly; autointerp and RAVEL cannot be tested in this lens because both require natural-language concepts the synthetic dictionary lacks.

Discriminability across training (§4, desiderata 2, 4, and 5). We train two panels of SAEs and snapshot them throughout training: a four-SAE *cross-architecture* panel with deliberately large differences (BatchTopK vs Matryoshka, $k \in \{50, 100\}$), and a four-SAE *Matryoshka* panel that varies a single hyperparameter n across three independent training seeds per variant. The cross-architecture panel asks whether the metric can tell apart very different SAEs; the Matryoshka panel asks the harder, more practically relevant question of whether it can tell apart the small hyperparameter variants practitioners typically compare. The training-trajectory axis lets us check desideratum 2 directly, and the three Matryoshka seeds let us decompose total score variance into between-SAE signal versus training-seed and snapshot noise.

Our findings are as follows. Across the three lenses, TPP and SCR fail multiple desiderata at canonical settings: TPP scores *worse* the more an SAE is trained, and SCR becomes negatively correlated with ground-truth at large top-N. Every other metric is noisier and less discriminative than the field assumes. The sae-probes variant of sparse probing is the most reliable metric we audit; its two main design choices, a much larger dataset suite (113 vs 5) and cross-validated probe regularization, suggest a concrete path forward for the rest of the suite.

Our code[1] and SAE panels[2] are publicly available.

[1]Code: https://github.com/decoderesearch/saebench-reliability-audit

[2]SAE panels: https://huggingface.co/decoderesearch/sae-snapshot-panels

2. Background

Sparse autoencoder A sparse autoencoder maps an activation vector $x \in \mathbb{R}^d$ to a sparse latent vector $z = \sigma(W_{\text{enc}}x + b_{\text{enc}}) \in \mathbb{R}^{d_{\text{sae}}}$ with $d_{\text{sae}} \geq d$, and reconstructs $\hat{x} = W_{\text{dec}}z + b_{\text{dec}}$. SAEs are trained to minimize reconstruction error subject to a sparsity penalty on z, with the goal of recovering interpretable feature directions in the underlying model's activation space (Bricken et al., 2023). By "SAE quality" we mean the degree to which the trained latent directions align with semantically meaningful features.

2.1. SAEBench metrics

SAEBench (Karvonen et al., 2025) measures quality through six proxy metrics[3]. Unless otherwise specified, we leave these at SAEBench default settings. Detailed descriptions of each metric are in Appendix B.

Sparse probing Trains a k-sparse logistic regression probe (Gurnee et al., 2023) on the SAE latents to classify natural-language attributes such as language or profession; higher accuracy at small k means concepts align with small numbers of latents. This benchmark uses 5 probing datasets.

Sae-probes sparse probing Added to SAEBench after the original suite's publication, this is a variation of sparse probing based on Kantamneni et al. (2025). This variant greatly increases the number of datasets used to 113, and uses L1-regularized logistic regression with cross-validation.

Spurious Correlation Removal (SCR) Based on the SHIFT method from Marks et al. (2025), trains intentionally biased probes on co-occurring concepts (e.g. all doctors are male) and measures how much that bias is removed when the SAE latents most associated with the spurious attribute are ablated. This benchmark uses 2 datasets.

Targeted Probe Perturbation (TPP) Also based on Marks et al. (2025), TPP trains a probe on each class within the same category (e.g. different profession), identifies the SAE latents whose ablation most damages the probe, and measures the gap in damage between the intended class probe and the unrelated probes when those top-N latents are zeroed. This benchmark uses 2 datasets.

Autointerp Based on Paulo et al. (2024), Autointerp first randomly selects N latents from the SAE, and uses an LLM to generate labels for those latents. Then, based on that description, a second LLM predicts whether or not that

[3]We do not audit the unlearning metric as it requires chat models while the other metrics use base models. We do not audit absorption and meta structure metrics as these are meant to diagnose a specific failure and are not general quality metrics.

Table 1. Reseed noise on the canonical 65k Gemma Scope SAE (Gemma-2-2b layer 12), five reseeds. The last column is the minimum-reliable-difference threshold $|\Delta|^*$ for a 95% two-tailed comparison of two single-seed scores. Curated subset; the full per-k and per-threshold table is in Appendix C.

Benchmark	Mean	Std	CV	Min reliable $\lvert\Delta\rvert^*$
sae-probes ($k = 5$, accuracy)	0.800	0.002	0.2%	0.008
sae-probes ($k = 1$, accuracy)	0.763	0.001	0.2%	0.004
RAVEL disentangle	0.706	0.001	0.2%	0.004
sparse_probing ($k = 5$, accuracy)	0.852	0.003	0.3%	0.012
autointerp	0.839	0.004	0.5%	0.016
sparse_probing ($k = 1$, accuracy)	0.722	0.009	1.2%	0.035
RAVEL cause	0.676	0.019	2.8%	0.075
TPP top-500	0.345	0.010	2.8%	0.039
SCR top-10	0.174	0.008	4.4%	0.031
SCR top-500	0.351	0.022	6.2%	0.086
TPP top-50	0.094	0.021	23%	0.083

latent will fire on held-out explanations.

RAVEL Based on Huang et al. (2024), RAVEL evaluates disentanglement, causal effect, and isolation of latents corresponding to known relational properties. The metric tries to use SAE latents to causally change predicted attributes for target entities without affecting related entities.

3. Reseed noise on a real LLM SAE

To measure reseed noise we ran each SAEBench evaluation five times on a canonical Gemma Scope JumpReLU SAE (residual stream layer 12 of Gemma-2-2b, width 65k) (Lieberum et al., 2024; Team et al., 2024). Each run uses a different random seed and disables activation caching so that each seed is truly independent.

Per-metric reseed noise. Table 1 summarizes coefficient of variation (CV) on the canonical SAE for a curated subset of metrics. The noise structure is consistent across SAE families and the full per-k and per-threshold breakdown (including the fixed-latent autointerp variant) is in Appendix C.

The spread in noise CV is wide. sae-probes ($k = 5$), sparse probing ($k = 5$), and RAVEL disentangle are at CV $\leq 0.3\%$, low enough that single-seed comparisons resolve score deltas at or above ~ 0.01. SCR has CV 3–6%; TPP has 16–39% CV at small top-N where the score itself is close to zero, so single-seed comparisons read mostly noise. The fifth column $|\Delta|^*$ reports the minimum-reliable difference for a 95% two-tailed comparison of two single-seed scores, computed as $t_{0.025,\,4} \cdot s\sqrt{2} \approx 3.93\, s$ where s is the column-3 std (derivation, multi-seed shrinkage, and cross-SAE caveats in Appendix D). Sparse probing, sae-probes, autointerp, and RAVEL disentangle have thresholds well below 0.04 and are reliable single-seed for almost any plausible comparison; SCR (especially at large top-N) and TPP at moderate top-N require multi-seed averaging.

4. Discriminability across training trajectories

To check whether SAEBench metrics can actually distinguish SAEs, we train two panels with different settings, snapshot them throughout training, and run all SAEBench metrics on each snapshot. The training-trajectory axis additionally surfaces clear failures like metrics that decline as the SAE trains.

Cross-architecture SAE set We train 4 SAEs (BatchTopK $k \in \{50, 100\}$, Matryoshka $k \in \{50, 100\}$) for 1.5B tokens each on Gemma-2-2b layer 12, with LR decay over the final $1/5$, taking 28 checkpoints per SAE. These SAEs are deliberately very different; if SAEBench cannot differentiate them, the metrics have a serious problem.

Matryoshka SAE set We train 4 Matryoshka SAEs (Bussmann et al., 2025) varying in the number of Matryoshka levels per SAE, identified as $n \in \{1, 2, 3, 4\}$, and train each variant from 3 independent random seeds (12 SAE training runs total) so we can separate variant signal from training-seed noise. We train for 300M tokens on Gemma-2-2b layer 12 with LR decay over the final $1/5$, and take 10 post-warmup snapshots per (variant, seed).

4.1. Results

We plot a representative subset of proxy metrics vs training tokens for the cross-architecture SAE set in Figure 1, and for the Matryoshka SAE set in Figure 2. We include the core reconstruction Mean Squared Error (MSE) as a reference. Full results are shown in Appendix T. Further analysis of metric training jitter and discriminability is in Appendix Q.

SCR and TPP decrease during training The most clearly problematic metrics are again SCR and TPP. At top-500, both SCR and TPP actually *decline* during training, implying the clearly incorrect conclusion that an untrained SAE is better than a trained SAE. The decline is not lim-

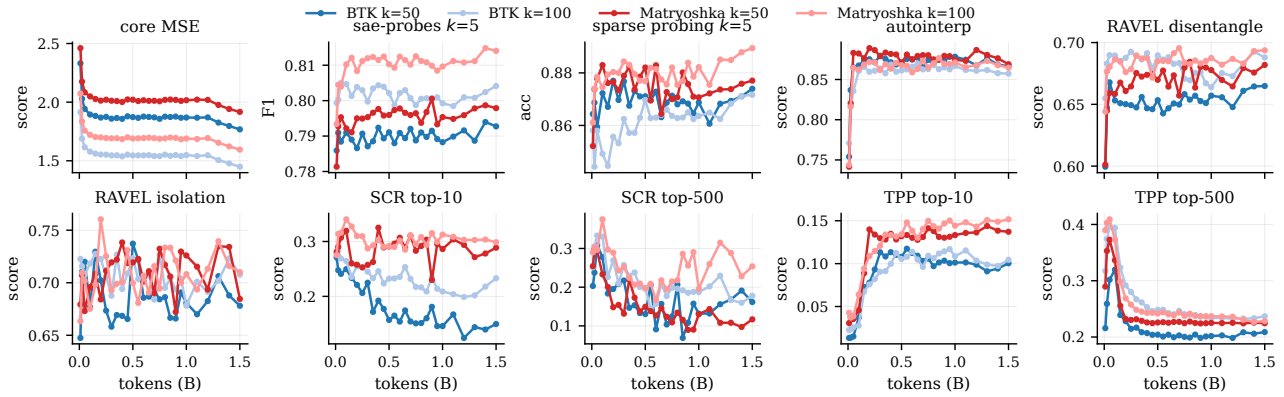

Figure 1. Trajectories of ten representative metrics across the 1.5B-token training run in the cross-architecture SAE set. SCR and TPP top-500 both show that training the SAE makes the metric worse, a clear red-flag for metric quality. Metrics demonstrate varying levels of discriminability, but no metric approaches the clear separation and stability of MSE.

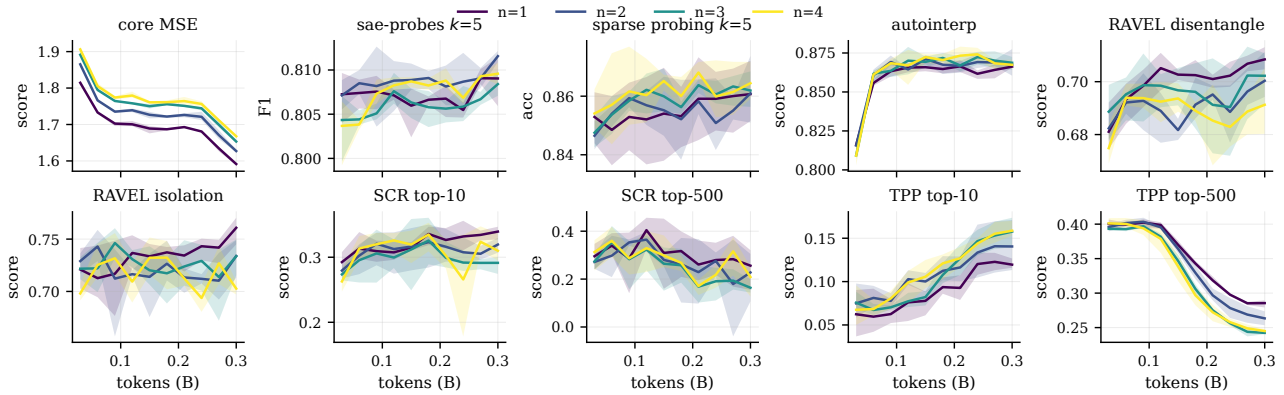

Figure 2. Trajectories of the same ten headline metrics as Figure 1, on the four-SAE Matryoshka set ($n \in \{1, \ldots, 4\}$ inner widths sampled per training step, 300M tokens, 3 training seeds per variant). Solid lines are seed means; shaded bands are min/max envelopes across the 3 seeds. top-500 SCR and TPP show problematic declines over training. No non-core quality metric clearly differentiates these SAEs once seed envelopes are accounted for; multi-seed analysis is in Appendix S.

ited to high top-N: SCR also declines through training at the canonical top-$N = 10$ on 2 of 4 SAEs in the cross-architecture panel (BatchTopK $k = 50$ and $k = 100$, slopes -0.065 and -0.033 score units per billion training tokens; Appendix P, Table 15), so a practitioner running SCR at canonical settings on a standard architecture can observe the metric say their SAE is getting worse as training progresses. In addition, we do not see SCR and TPP ranking SAEs consistently between low and high values of top-N, making it hard to trust these metrics.

No metric reaches the discriminability of MSE The inter-checkpoint jitter for all metrics is still much higher than it is for MSE, indicating there is still large room for improvement in metrics. For the cross-architecture SAE set, sae-probes does a good job of differentiating the different SAEs throughout training. For the Matryoshka SAE set, the four core reconstruction metrics, sae-probes ($k \in \{1, 5\}$), and RAVEL disentangle carry a real but small inter-variant signal (next paragraph); every other non-core metric, including all canonical SCR and TPP thresholds, is dominated by training-seed and snapshot noise rather than by the SAE variant. SAEBench is therefore not yet discerning enough to clearly guide most SAE hyperparameter decisions.

Variance decomposition on the Matryoshka panel The three training seeds per variant let us split each metric's score variance into three sources: the SAE variant we picked, training-seed randomness within a variant, and snapshot-to-snapshot jitter within a single training run. For most audited metrics, training-seed and snapshot variance dominate: across three training seeds and ten post-warmup checkpoints, only seven of 34 metrics carry a between-variant share large enough to confidently distinguish from chance (the four core reconstruction metrics, sae-probes at $k \in \{1, 5\}$, and RAVEL disentangle; details in Appendix S). At SCR top-

$\{5, 10, 50\}$, TPP top-$\{10, 20, 100\}$, sparse probing top-1, sae-probes $k = 16$, and RAVEL cause, each of the three training seeds picks a different best variant among the four. A practitioner who runs these benchmarks once on a panel of similar SAEs reads a winner that is essentially seed-driven.

Cross-metric ranking agreement On both panels, we compute pairwise Spearman ρ of metric rankings across all SNR-informative metrics (SNR ≥ 1, see Appendix Q). When SAEs are very different (cross-architecture panel) metrics broadly agree on the ranking ($\rho = +0.44$); when SAEs are similar (Matryoshka panel) metrics rank essentially independently ($\rho = +0.08$), with stark same-family disagreements such as default sparse probing k=1 vs k=10 negatively correlated at $\rho = -0.4$. Full heatmaps for both panels are in Appendix R, Figure 11.

5. Validity: ground-truth correlation on synthetic SAEs

A metric can have low reseed noise and good discriminability while still measuring the wrong thing: an arbitrary, repeatable function of SAE latents would pass both tests. To check that each metric's score actually reflects SAE quality, we need SAEs whose ground-truth quality is computable, which is only possible on synthetic models. We can then train SAEs on the synthetic model and check whether each proxy metric's score correlates with that ground-truth quality. Sparse probing, TPP, and SCR can be validated this way. Autointerp requires real natural-language text, and RAVEL requires a real LLM to intervene on, so neither can be tested under this lens.

5.1. SynthSAEBench

We use SynthSAEBench (Chanin & Garriga-Alonso, 2026) as a base for our evaluation. SynthSAEBench creates synthetic activations for SAE training with hierarchy, correlation, and superposition, simulating realistic phenomena in an LLM, while providing ground-truth features. The SynthSAEBench-16k model defines 16,384 ground-truth feature directions $d_i \in \mathbb{R}^D$ as random unit vectors. Of these, 10,880 are arranged in 128 disjoint trees of branching factor 4 and depth 3; the remaining 5,504 are non-hierarchical. Within a tree, children are mutually exclusive and a child can fire only if its parent does. Hierarchical roots and non-hierarchical features fire independently with per-feature firing rates.

To draw an activation, a binary firing pattern $z \in \{0, 1\}^{16,384}$ is sampled under these constraints, per-feature magnitudes are drawn from a rectified Gaussian $c_i = z_i \cdot \mathrm{ReLU}(\mu_i + \sigma_i \varepsilon_i)$ with $\varepsilon_i \sim \mathcal{N}(0, 1)$, and the SAE input is $x = D^\top c + b$, where D stacks the unit-norm feature directions and b is a bias. SAEs train on this activation pool and never see D. Feature firing probabilities follow a zipfian distribution, where earlier feature indices fire more frequently than later features.

SynthSAEBench provides two key metrics to score how well a trained SAE recovers the ground-truth features. *Mean Correlation Coefficient (GT-MCC)* matches the SAE's decoder columns w_j to the ground-truth directions on absolute cosine similarity (Hungarian assignment) and reports the mean cosine similarity of matched pairs; this measures how closely the SAE dictionary matches the ground-truth feature directions. *GT-F1* matches each SAE latent to its best-correlated ground-truth feature and reads binary-classification F1 of latent firing against ground-truth feature firing across activations. This metric measures how well the SAE encoder fires the correct features at the correct times.

5.2. Evaluation SAEs

We train a series of SAEs of different architectures with width $d_{\text{sae}} = 4{,}096$ on SynthSAEBench, as recommended by SynthSAEBench (Chanin & Garriga-Alonso, 2026). We assume that the SAEs thus learn roughly the first 4,096 features of SynthSAEBench-16k. We call the first 4,096 SynthSAEBench features *in-sae*, and the remaining 12,288 features *out-of-sae*. This is illustrated in Figure 3.

Synthetic SAEs We train 35 SAEs at $d_{\text{sae}} = 4096$. These SAEs use 5 architectures (standard, JumpReLU, BatchTopK, Matryoshka, Matching Pursuit) each at 7 L0 levels ($\{15, 20, 25, 30, 35, 40, 45\}$). Each SAE is trained on 200M samples following the SynthSAEBench recommended hyperparameters (Chanin & Garriga-Alonso, 2026).

Control SAEs We include 3 degraded variants of the canonical JumpReLU L0=25 SAE: `random_init` and `random_l0_matched` randomize weights, the latter with threshold tuned to L0 ≈ 25; `permuted_decoder` shuffles W_{dec} rows. We also include a *perfect oracle* whose decoder is the first 4096 ground-truth feature directions and encoder returns exact ground-truth activations (GT-MCC = GT-F1 = 1.0 by construction).

5.3. Synthetic task datasets

We construct hierarchy-aligned synthetic analogues of each SAEBench benchmark, partitioned by whether the probing targets consist of feature indices that are in-sae, out-of-sae or a mix. All tasks for a given evaluation share a 60,000-activation pool drawn fresh from the synthetic model per task-generation seed (so the pool, the per-task labels, and the per-task feature selections all re-randomize on reseed); each task then subsets this pool to its own balanced positive/negative classes, with class-balanced positive/negative splits and an 80/20 train/test partition where applicable.

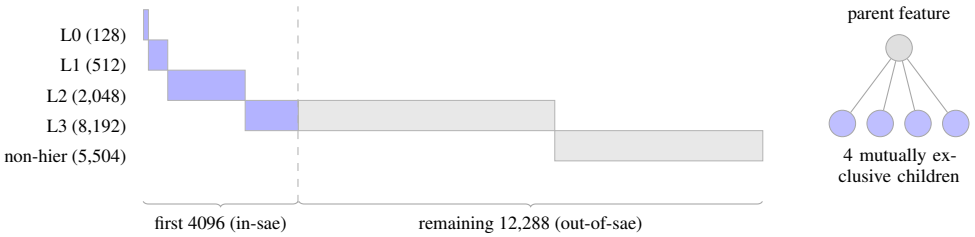

Figure 3. SynthSAEBench-16k contains a 16,384-feature ground-truth dictionary broken into hierarchy levels L0–L3, and non-hierarchical tail. We train SAEs of width 4096, so the first 4096 features of SynthSAEBench-16k are considered in-sae for the SAE (blue). The remaining 12,288 are out-of-sae (gray); the dashed line marks the cutoff, partway through L3.

Table 2. Example tasks from the synthetic-task suite used in §5. Subscripts are ground-truth feature indices; in-sae means index $< 4{,}096$, out-of-sae means index $\geq 4{,}096$.

Task category	Example
Sparse probing (binary classification on activations; train top-k probe on SAE latents)	
single in-sae	is $f_{1,347}$ firing?
single out-of-sae	is $f_{12,492}$ firing?
boolean in-sae	is $f_{17} \wedge f_{1,098}$ firing? or is $f_{12} \vee f_{123}$ firing?
boolean out-of-sae	is $f_{12,340} \vee f_{15,873}$ firing?
boolean mixed	is $f_{2,143} \wedge f_{15,873}$ firing? (one in-sae, one out-of-sae)
TPP (4-sibling group at depth 3; ablate top-N latents most damaging to focal probe)	
in-sae group	focal $f_{2,688}$, siblings $\{f_{2,689}, f_{2,690}, f_{2,691}\}$
out-of-sae group	focal $f_{8,000}$, siblings $\{f_{8,001}, f_{8,002}, f_{8,003}\}$
SCR ((T, S) pair; biased probe on $T \wedge S$, ablate top-N latents associated with S)	
in-sae pair	target $T = f_{317}$, spurious $S = f_{1,422}$ (different L0 subtrees)

Sparse probing We construct 92 binary classification tasks ("is dictionary feature f firing in this activation?"), split by SAE coverage (in-sae / out-of-sae / mixed) and structure (single feature vs 2–3-feature boolean combination); for each task we draw a class-balanced split of up to 1,024 training activations (target 512 positives + 512 negatives) and up to 3,000 test activations (target 1,500 + 1,500), then train a k-sparse logistic regression on SAE latents and read top-k accuracy.

TPP We construct 60 four-sibling groups at depth 3 (analogue of SAEBench's 4-profession bias-in-bios setup), 30 in-sae and 30 out-of-sae; for each group we draw 2,000 samples per sibling class (up to 8,000 activations per group, balanced across the four siblings), train one probe per sibling under an 80/20 train/test split, identify the SAE latents whose ablation most damages the focal-sibling probe, and measure the gap between damage to the focal probe and damage to its three sibling probes.

SCR We construct between 9 and 12 (T, S) pairs per seed (Appendix G) with features T and S from different hierarchy subtrees so they can co-occur. For each pair we draw 1,500 samples per class across the six SCR classes (T, $\neg T$, S, $\neg S$, $T \wedge S$, $\neg T \wedge \neg S$), balanced down to the smallest eligible class and split 80/20 train/test; we train a biased probe only on samples where T and S co-fire, then measure how much that bias is removed when the SAE latents most associated with S are ablated. All SCR pairs are in-sae because SCR requires the (T=1 $\wedge$ S=1) cell of the contingency table to be populated, and out-of-sae features fire too rarely for an out-of-sae control for SCR.

Table 2 gives one concrete example per task category. We include out-of-sae task variants as a control: if a proxy metric scores highly on a feature that the SAE is *not explicitly tracking*, then something has gone awry. Metrics should score out-of-sae task variants much lower than in-sae and mixed variants.

5.4. Results

For each proxy metric task, we calculate the Spearman correlation between the proxy metric and the ground-truth metrics GT-MCC and GT-F1. We consider it a success if the proxy metric tracks either metric, as the ground-truth metrics themselves track different quality aspects of the underlying SAE, and do not always peak at the same L0. We report correlation results for all tasks in Table 3. We show a subset of scatter plots for each task in Figure 4. We additionally run ablation experiments on modified versions of the SynthSAEBench-16k base model in Appendix I, which probes whether the calibration we observe is a property of

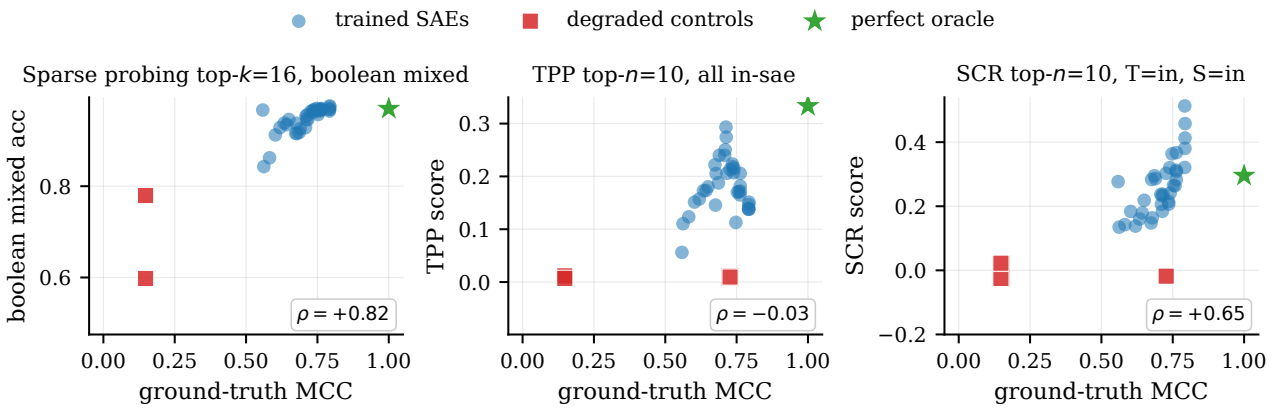

Figure 4. Benchmark score vs GT-MCC at canonical hyperparameters, each point is one SAE. Sparse probing (left) uses the boolean-mixed task category. TPP (middle) and SCR (right) use canonical top-N=10. Sparse probing is well-calibrated with GT-MCC. TPP shows no correlation with GT-MCC. Problematically, SCR gives a lower score to the perfect oracle than to several of the trained SAEs.

Table 3. Spearman ρ between each benchmark's score and ground-truth quality metric. Metrics should correlate with either GT-MCC or GT-F1 for in-sae and mixed tasks, and should not correlate with out-of-sae tasks. Sparse probing is well-calibrated to GT-MCC across k. SCR becomes negatively correlated with GT-MCC at large top-N.

	ρ vs GT-MCC			ρ vs GT-F1		
Sparse probing, top-k =	*1*	*4*	*16*	*1*	*4*	*16*
single in-sae	+0.70	+0.76	+0.49	+0.31	+0.46	−0.22
boolean in-sae	+0.48	+0.68	+0.87	+0.47	+0.37	+0.51
boolean mixed	+0.55	+0.65	+0.82	+0.52	+0.45	+0.46
single out-of-sae (control)	+0.01	+0.23	+0.19	−0.35	−0.58	−0.37
boolean out-of-sae (control)	−0.29	−0.41	−0.22	−0.25	−0.53	−0.53
TPP, top-N =	*10*	*100*	*500*	*10*	*100*	*500*
all in-sae	−0.03	+0.46	+0.60	−0.69	−0.57	−0.49
all out-of-sae (control)	+0.42	+0.75	+0.71	−0.03	+0.36	+0.25
SCR, top-N =	*10*	*100*	*500*	*10*	*100*	*500*
T=in, S=in	+0.65	−0.15	−0.64	+0.41	−0.34	−0.42

the v1 base model or holds across feature distributions.

Sparse probing is reasonably well calibrated to GT-MCC Sparse probing interestingly tracks GT-MCC better than GT-F1, despite being an encoder-only metric. Sparse probing does not penalize latents for firing more than they should since the k-sparse probe can learn its own threshold per firing latent. GT-MCC stays high at high L0, while GT-F1 decays past a clear optimal L0 on most architectures (Appendix J, Figure 5). This is not necessarily a problem, as GT-MCC is a valid metric.

One potential problem with sparse probing is that it saturates well before perfect GT-MCC. At canonical top-$k = 16$, the in-sae single-feature accuracy is above 0.99 for almost every trained SAE in this study, while their GT-MCC values span the much wider range 0.56–0.79. Without spread on the metric side, sparse probing struggles to differentiate between high-quality SAEs.

TPP is poorly correlated with ground-truth TPP correlates poorly with GT-MCC at low top-N, with essentially 0 correlation at the canonical top-$N = 10$. Strangely, TPP appears to be better-correlated with out-of-sae features on SynthSAEBench-16k, implying it is tracking some sort of feature mixing between in-sae and out-of-sae features rather than directly measuring SAE quality, but this also varies a lot in our variant model ablations (Appendix I). Either way, the canonical TPP score is not a reliable proxy for SAE quality on this synthetic panel.

SCR is unreliable across multiple axes SCR has three serious problems in this synthetic test. First, even at the canonical top-$N = 10$, SCR ranks the perfect oracle lower than 11 of the 35 trained SAEs tested. Second, when top-N is 50 or higher, the metric becomes *negatively correlated* with all ground-truth metrics ($\rho_{\text{MCC}} = -0.31$ at top-$N = 50$, -0.64 at top-$N = 500$ on the v1 panel; Table 3), meaning better SAEs score lower on SCR than worse SAEs. We

document further SCR failures on variant synthetic models in Appendix I, Table 12.

6. Related work

SAE benchmarking failures. Paulo & Belrose (2025) find that only ∼30% of features are shared across seed-replicated training runs of the same SAE, raising identifiability concerns. Korznikov et al. (2026) report that random-feature baselines match trained SAEs on autointerp, sparse probing, and causal editing; Song et al. (2025) propose a feature-consistency metric (PW-MCC) as a complementary reliability axis. Our audit is complementary to these: rather than introducing a single new diagnostic, we evaluate the existing SAEBench suite along three orthogonal lenses (reseed noise, ground-truth correlation, and discriminability across training).

Cases where supervised baselines beat SAEs. Kantamneni et al. (2025) report that SAE-based sparse probes underperform plain logistic regression across data scarcity, label noise, and covariate shift; Wu et al. (2025) find SAE steering inferior to prompting and difference-in-means representations. DeepMind's recent negative-results report (Smith et al., 2025) echoes both findings on a safety-relevant probing task. These results suggest the existing proxy metrics overstate SAE quality on downstream tasks, motivating an audit of the proxies themselves.

Ground-truth and synthetic SAE evaluation. The toy-model lineage initiated by Elhage et al. (2022) establishes sparse linear combinations of orthogonal feature directions as a controlled testbed for dictionary learning. Karvonen et al. (2024) run an analogous evaluation on Othello and chess board states, and Makelov et al. (2024) construct supervised dictionaries on the IOI task as a comparison ground truth. O'Neill et al. (2025) introduce the optimal-bipartite-matching MCC metric that SynthSAEBench (Chanin & Garriga-Alonso, 2026) adopts. We use SynthSAEBench-16k as our validity panel.

Benchmark auditing in ML more broadly. Liao et al. (2021) catalog 107 surveys of evaluation failures, distinguishing internal validity (noise, leakage, baselines) from external validity (transfer of progress to other tasks); Bowman & Dahl (2021) propose explicit desiderata for NLU benchmarks, the format we follow. Dehghani et al. (2021) show that re-aggregating different subsets of SuperGLUE substantially reorders rankings of competing models. Henderson et al. (2018) and Reimers & Gurevych (2017) study on seed variance in deep RL and NLP. Card et al. (2020) quantifies when test-set sizes are too small for benchmark comparisons to have statistical power.

7. Discussion

Practitioners should avoid SCR and TPP TPP and SCR fail multiple basic sanity checks at their canonical settings on trained SAEs. TPP declines through training at top-$N \geq 50$ across every SAE we tested, and is poorly correlated with ground truth in our synthetic tests. SCR is negatively correlated with ground-truth at top-$N \geq 50$ (§5.4), stays flat or declines through training at canonical top-$N = 10$ on every SAE in our cross-architecture panel (§4), and swings from $\rho = +0.90$ to $\rho = -0.10$ against GT-MCC across reasonable variations of the same synthetic base model (Appendix I). We recommend avoiding SCR and TPP benchmarks.

SAE metrics struggle with noise and discriminability Many metrics have inter-checkpoint jitter comparable to their inter-variant signal, making them hard to use as a clear target for the field to hill-climb on. The most promising directions for future work are increasing the number and variety of datasets used by each metric, improving the reliability of internal probing steps, and replacing random latent sampling with reproducible selections that transfer across SAE training runs.

Limitations of our audit Even for metrics that pass our checks, we cannot say with certainty that a proxy metric is "correct" on LLMs. It is possible for biases or dataset limitations to remain that we cannot measure without ground-truth on real LLMs. Limitations are discussed further in Appendix A.

The field needs better SAE benchmarks Without clear and reliable benchmarks, the field will struggle to improve SAEs. Progress on SAE architectures requires stacking small improvements that the current suite cannot reliably detect. We believe SAE benchmarking is solvable, and we hope this audit catalyzes further work to extend and improve the existing suite.

Acknowledgements

David Chanin was supported thanks to EPSRC EP/S021566/1 and the Machine Learning Alignment and Theory Scholars (MATS) program. Compute for training and evaluating LLM SAEs was provided thanks to Modal.

References

Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J., and Saunders, W. Language models can explain neurons in language models. 2023, 2023. URL https://openaipublic

.blob.core.windows.net/neuron-explainer/paper/index.html.

Bowman, S. R. and Dahl, G. What will it take to fix benchmarking in natural language understanding? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4843–4855, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.385. URL https://aclanthology.org/2021.naacl-main.385/.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Bussmann, B., Leask, P., and Nanda, N. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Bussmann, B., Nabeshima, N., Karvonen, A., and Nanda, N. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

Card, D., Henderson, P., Khandelwal, U., Jia, R., Mahowald, K., and Jurafsky, D. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Chanin, D. and Garriga-Alonso, A. Synthsaebench: Evaluating sparse autoencoders on scalable realistic synthetic data. *arXiv preprint arXiv:2602.14687*, 2026.

Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., Golechha, S., and Bloom, J. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *Advances in Neural Information Processing Systems*, 38: 82318–82355, 2026.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Dehghani, M., Tay, Y., Gritsenko, A. A., Zhao, Z., Houlsby, N., Diaz, F., Metzler, D., and Vinyals, O. The benchmark lottery. *arXiv preprint arXiv:2107.07002*, 2021.

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

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. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gurnee, W., Nanda, N., Pauly, M., Harvey, K., Troitskii, D., and Bertsimas, D. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=JYs1R9IMJr.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. doi: 10.1609/aaai.v32i1.11694. URL https://ojs.aaai.org/index.php/AAAI/article/view/11694.

Huang, J., Wu, Z., Potts, C., Geva, M., and Geiger, A. Ravel: Evaluating interpretability methods on disentangling language model representations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8669–8687, 2024.

Kantamneni, S., Engels, J., Rajamanoharan, S., Tegmark, M., and Nanda, N. Are sparse autoencoders useful? a case study in sparse probing. *arXiv preprint arXiv:2502.16681*, 2025.

Karvonen, A., Wright, B., Rager, C., Angell, R., Brinkmann, J., Smith, L., Mayrink Verdun, C., Bau, D., and Marks, S. Measuring progress in dictionary learning for language model interpretability with board game models. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 83091–83118. Curran Associates, Inc., 2024. doi: 10.52202/079017-2644. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9736acf007760cc2b47948ae3cf06274-Paper-Conference.pdf.

Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J., Chanin, D., Lau, Y.-T., Farrell, E., McDougall, C., Ayonrinde, K., Till, D., Wearden, M., Conmy, A., Marks, S., and Nanda, N. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability, 2025. URL https://arxiv.org/abs/2503.09532.

Korznikov, A., Galichin, A., Dontsov, A., Rogov, O., Oseledets, I., and Tutubalina, E. Sanity checks for sparse autoencoders: Do saes beat random baselines? *arXiv preprint arXiv:2602.14111*, 2026.

Liao, T., Taori, R., Raji, I. D., and Schmidt, L. Are we learning yet? a meta review of evaluation failures across machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=mPducS1MsEK.

Lieberum, T., Rajamanoharan, S., Conmy, A., Smith, L., Sonnerat, N., Varma, V., Kramár, J., Dragan, A., Shah, R., and Nanda, N. Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2, August 2024.

Makelov, A., Lange, G., and Nanda, N. Towards principled evaluations of sparse autoencoders for interpretability and control. *arXiv preprint arXiv:2405.08366*, 2024.

Marks, S., Rager, C., Michaud, E. J., Belinkov, Y., Bau, D., and Mueller, A. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

McDougall, C., Conmy, A., Kramár, J., Lieberum, T., Rajamanoharan, S., and Nanda, N. Gemma scope 2: Technical paper. Technical report, Google DeepMind, 2025. URL 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.

O'Neill, C., Gumran, A., and Klindt, D. Compute optimal inference and provable amortisation gap in sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=8forr1FkvC.

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

Paulo, G., Mallen, A., Juang, C., and Belrose, N. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024.

Reimers, N. and Gurevych, I. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Smith, L., Rajamanoharan, S., Conmy, A., McDougall, C., Kramár, J., Lieberum, T., Shah, R., and Nanda, N. Negative results for sparse autoencoders on downstream tasks and deprioritising SAE research (mechanistic interpretability team progress update). Google DeepMind Safety Research, Medium, 2025. URL https://deepmindsafetyresearch.medium.com/negative-results-for-sparse-autoencoders-on-downstream-tasks-and-deprioritising-sae-research-6cadcfc125b9.

Song, X., Muhamed, A., Zheng, Y., Kong, L., Tang, Z., Diab, M. T., Smith, V., and Zhang, K. Position: Mechanistic interpretability should prioritize feature consistency in SAEs. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=d9ACURK6bI.

Team, G., 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. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Wu, Z., Arora, A., Geiger, A., Wang, Z., Huang, J., Jurafsky, D., Manning, C. D., and Potts, C. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025.

A. Limitations

The synthetic-SAE correlation analysis (§5) on SynthSAEBench-16k uses three task-generation seeds; the SAE-variation models are evaluated at a single seed each, so the cross-base-model fragility of SCR canonical we report combines genuine cross-model variation with single-seed task-sampling noise. The cross-architecture training-trajectory analysis (§4) uses a single training seed per variant, so per-step jitter on that panel is a lower bound on the metric's pure noise (it includes some actual SAE-quality drift across snapshots); the sampled-Matryoshka panel uses three training seeds per variant and is decomposed in Appendix S. Three seeds is the minimum that supports a variance decomposition, so individual share values there are imprecise even though the qualitative ordering is robust. Our discriminability (§4) and validity (§5) analyses use a single underlying language model (Gemma-2-2b); discriminability and validity may behave differently on other model families, though the four-SAE cross-model reseed-noise comparison in Appendix C suggests the noise patterns themselves are largely model-agnostic. The four-SAE sampled-Matryoshka panel covers 300M tokens of training, so trajectories may not have converged. SynthSAEBench-16k is one synthetic model architecture, so generalizing the calibration findings to other ground-truth setups is open work.

We do not evaluate the following SAEBench metrics: *absorption* (Chanin et al., 2026) and *meta_structure* diagnose specific SAE failure modes (feature absorption and latent-space meta-structure) rather than scoring quality, and *unlearning* targets instruct-tuned-model SAEs while every other SAEBench evaluation and the SAE training literature target base models. Several audited evaluations originated as standalone benchmarks (Gurnee et al., 2023; Bills et al., 2023; Kantamneni et al., 2025; Marks et al., 2025; Huang et al., 2024); we audit their SAEBench-as-configured versions only. We also consider AxBench (Wu et al., 2025) as out of scope, as it is a general steering benchmark rather than being SAE-specific.

B. SAEBench metric details

This appendix gives the full procedure for each of the six SAEBench quality metrics audited in the paper. The condensed descriptions in §2 are sufficient for following the audit findings; this appendix is the reference for readers who want the implementation-level setup, hyperparameters, and dataset choices.

Sparse probing (Gurnee et al., 2023) measures whether SAE latents disentangle natural-language concepts into a small number of latents. For each binary probing task (e.g. "is this text in French" or "is this profession a doctor"), SAEBench encodes the text into SAE latents, identifies the top-k latents most informative about the binary label using per-latent statistics, and trains a logistic-regression probe restricted to those k latents. The headline metric is test accuracy at $k \in \{1, 2, 5\}$; higher accuracy at small k indicates the relevant concept is concentrated in a small number of latents. The default benchmark uses 5 probing datasets covering language identification, profession (bias-in-bios), and topic classification.

Sae-probes sparse probing (Kantamneni et al., 2025) is a variant of sparse probing that runs the same conceptual evaluation on a much larger and more diverse task suite. Sae-probes uses 113 binary classification tasks (vs sparse probing's 5), trains an L1-regularized logistic-regression probe with cross-validation over the L1 strength, and reports test AUC, F1, and accuracy at multiple values of k. The cross-validated regularization makes sae-probes results less sensitive to the specific probe-training hyperparameters, and the larger task suite reduces the dataset-sampling noise of the headline number.

Spurious Correlation Removal (SCR) (Marks et al., 2025) tests whether SAE latents can be used to remove a spurious correlation from a probe without retraining. The setup picks a target attribute T (e.g. profession) and a spurious attribute S (e.g. gender) and trains a logistic-regression probe to predict T on a confounded subset of the data containing only $T \wedge S$ and $\neg T \wedge \neg S$ examples, so the probe can solve the training task by reading either attribute. SCR then identifies the top-N SAE latents most associated with S (by largest mean-activation difference between $S=1$ and $S=0$ examples), ablates them, and reports how much of the accuracy gap between this biased probe and an oracle probe trained on un-confounded data is closed on a held-out balanced test set where T and S are independent. A higher SCR score means the SAE successfully isolated the spurious attribute into a small set of removable latents. The benchmark uses 2 datasets, and the canonical top-N is 10; SAEBench's

default sweep reports top-$N \in \{2, 5, 10, 20, 50, 100, 500\}$. The internal probe is trained with ℓ_1 penalty 10^{-3} at batch size 16; both are load-bearing for the metric value (Appendix H).

Targeted Probe Perturbation (TPP) (Marks et al., 2025) measures whether the SAE causally separates concepts within a single category. The default setup uses bias-in-bios restricted to a 4-class group of professions. For each focal class c in the group, SAEBench trains four binary probes (one per class) on the SAE latents, then identifies the top-N latents whose ablation most damages the focal probe's accuracy, zeroes those N latents, and measures the accuracy drop on the focal probe versus the average accuracy drop on the three non-focal probes. The TPP score is the gap (focal drop minus non-focal drop), averaged over choice of focal class. A high TPP means each class's representation can be isolated to a small set of latents that does not affect the other classes' probes. The benchmark uses 2 datasets and reports across $N \in \{2, 5, 10, 20, 50, 100, 500\}$. As with SCR, the internal probe-training ℓ_1 penalty and batch size matter for the metric value.

Autointerp (Bills et al., 2023; Paulo et al., 2024) measures how well SAE latents can be described in natural language and how well those descriptions predict latent firing. SAEBench's workflow first selects N latents from the SAE (uniformly at random by default), collects the top-activating natural-language examples for each, asks an LLM (the explainer) to generate a natural-language description of what each latent fires on, and then asks a second LLM (the scorer) to predict, given the description, which held-out examples should activate the latent. The autointerp score is the agreement between scorer predictions and ground-truth firing patterns. The default uses $N = 1000$ randomly-selected latents and a total token budget of 2M. We additionally evaluate a fixed-latent variant where the same 1000 latents are used across reseeds (selected via seed 42), so reseed noise is isolated to the LLM-judge variance and excludes latent-sampling noise.

RAVEL (Huang et al., 2024) measures whether SAE latents can be used to causally edit one attribute of an entity without affecting other attributes. The setup uses entity-attribute datasets (e.g. cities with attributes country, language, continent), and for each (target attribute, distractor attribute) pair the RAVEL pipeline finds SAE latents whose ablation changes the model's prediction of the target attribute (the *cause* score) without changing its prediction of the distractor attribute (the *disentangle* score); the *isolation* score combines the two. SAEBench reports all three sub-scores; in the body we focus on *disentangle*, which has the lowest reseed noise of the three. RAVEL's default datasets are Gemma-2-2b specific (entity sets and prompt templates are hard-coded for that base model), which is why we restrict the RAVEL evaluation to Gemma Scope SAEs in Appendix C.

C. Per-SAE reseed noise tables

Tables 4, 5, 6, and 7 give the full per-benchmark breakdown of reseed CV on a four-SAE panel spanning three model families: the canonical 65k and 16k Gemma Scope SAEs (residual stream layer 12 of Gemma-2-2b), the canonical gpt2-small 32k SAE (gpt2-small layer 6 `hook_resid_post`), and the canonical Llama Scope 8x SAE (Llama-3.1-8B layer 16 residual, width 32k). All four use five reseeds at seeds $\{42, 123, 456, 789, 2024\}$ at canonical hyperparameters. Same protocol as §3. Table 1 in §3 is the curated subset of Table 4. The qualitative pattern is the same across all four SAEs: sae-probes, sparse probing top-k, RAVEL disentangle, and autointerp are sub-1% CV; TPP and SCR at large top-N are mid-single-digit CV (with the notable exception of SCR top-500 on gpt2-small at 27%); TPP at small top-N has very high relative noise because the score itself is close to zero. RAVEL is restricted to the two Gemma SAEs because the SAEBench RAVEL wrapper hard-codes datasets for Gemma-2-2b only.

D. Derivation of single-seed comparison thresholds

The fifth column of Table 1 reports a per-metric threshold $|\Delta|^*$ for a 95% two-tailed comparison of two single-seed scores. We derive it here.

The test. If a practitioner evaluates two SAEs at one seed each and compares their scores s_A and s_B, the difference $\Delta = s_A - s_B$ has variance $2\sigma^2$ under independent reseeds, where σ is the metric's per-seed standard deviation. We do not know σ exactly: the column-3 std s in Table 1 is estimated from only $n = 5$ canonical-SAE reseeds, so the right test is a two-sample t-test rather than a z-test. Under H_0 "the two SAEs have equal true score" the statistic $\Delta/(s\sqrt{2})$ follows a t-distribution with $n - 1 = 4$ degrees of freedom; the 95% two-tailed threshold is therefore

$$|\Delta|^* \;=\; t_{0.025,\,4} \cdot s\sqrt{2} \;\approx\; 3.93\, s.$$

The equivalent multiplier under known σ would be $1.96\sqrt{2} \approx 2.77$; the inflation factor $t_{0.025,4}/z_{0.025} \approx 1.41$ reflects the chi-squared uncertainty in s given only 5 reseeds (the 95% CI on σ from $n = 5$ samples is roughly $0.6s$ to $2.9s$). Concretely: a practitioner who runs sae-probes on two SAEs and sees a 0.005 gap should not conclude one is better, because the threshold is 0.008; on SCR top-500 the same logic requires $|\Delta| > 0.09$ before a single-seed comparison is informative.

Table 4. Full reseed noise breakdown on the canonical 65k Gemma Scope SAE, five reseeds.

Benchmark	Mean	Std	CV
sae-probes ($k = 1$, accuracy)	0.763	0.001	0.2%
sae-probes ($k = 2$, accuracy)	0.780	0.001	0.1%
sae-probes ($k = 5$, accuracy)	0.800	0.002	0.2%
sparse_probing ($k = 1$, accuracy)	0.722	0.009	1.2%
sparse_probing ($k = 2$, accuracy)	0.773	0.005	0.6%
sparse_probing ($k = 5$, accuracy)	0.852	0.003	0.3%
RAVEL cause	0.676	0.019	2.8%
RAVEL disentangle	0.706	0.001	0.2%
RAVEL isolation	0.736	0.018	2.5%
autointerp random-latent ($n_{\text{latents}} = 1000$)	0.839	0.004	0.5%
autointerp fixed-latent ($n_{\text{latents}} = 1000$)	0.846	0.002	0.2%
SCR top-2	0.048	0.003	6.4%
SCR top-5	0.114	0.005	4.4%
SCR top-10	0.174	0.008	4.4%
SCR top-20	0.263	0.008	3.2%
SCR top-50	0.349	0.016	4.7%
SCR top-100	0.364	0.016	4.5%
SCR top-500	0.351	0.022	6.2%
TPP top-2	0.005	0.001	19%
TPP top-5	0.010	0.002	16%
TPP top-10	0.025	0.010	39%
TPP top-20	0.045	0.016	36%
TPP top-50	0.094	0.021	23%
TPP top-100	0.157	0.022	14%
TPP top-500	0.345	0.010	2.8%

Multi-seed extension. Averaging S independent reseeds of each SAE shrinks the per-mean variance from σ^2 to σ^2/S, so $\mathrm{Var}(\Delta) = 2\sigma^2/S$ and the threshold scales as $|\Delta|^*_S = t_{0.025,\,2S-2} \cdot s\sqrt{2/S}$, where the larger pooled $df = 2S - 2$ also shrinks the multiplier toward the z limit. For $S = 3$ (e.g. the Matryoshka-panel design of §4) the threshold is roughly $1.6\,s$; with $S = 5$ it is roughly $1.0\,s$.

Cross-SAE applicability. Column $|\Delta|^*$ is derived from reseed noise on the canonical 65k Gemma Scope SAE and applies most directly to comparisons involving that SAE. Reseed std varies up to $\sim$2$\times$ across the four canonical SAEs in our cross-model panel (Tables 4–7) — e.g. SCR top-10 std is 0.005 on gpt2-small but 0.009 on the 16k Gemma Scope SAE, $\sim$1.8$\times$ apart. We recommend treating Table 1's column 5 as a useful single-figure rule of thumb but, when a more careful threshold is needed for a specific SAE, recomputing $|\Delta|^*$ from that SAE's own per-metric std using the formula above. The full per-SAE std breakdowns are in Appendix C.

E. Diagnostic-SAE $\times$ six-benchmark validation

We applied the two real-LLM diagnostic SAEs introduced in §5 (`permuted_decoder` and `random_l0_matched`) to all six SAEBench evaluations on the canonical Gemma-Scope JumpReLU SAE (Gemma-2-2b layer 12 residual, width 16k, mean L0 124.7); both were rebuilt for this SAE. Each diagnostic was run at 3 reseeds per cell with activation caching disabled, matching the reseed-noise protocol of §3. `random_l0_matched` achieved L0 124.5 via a uniform threshold tuned on a 20k-token Gemma-2-2b L12 calibration set.

Reduced evaluation configuration. For compute reasons the diagnostic-SAE evaluation that produced Table 8 used a smaller subset of the SAEBench default configuration: sparse probing on 3 of the 8 default datasets (`LabHC/bias_in_bios_class_set1`, `canrager/amazon_reviews_mcauley_1and5`, `fancyzhx/ag_news`); SCR on the single `LabHC/bias_in_bios_class_set1` dataset (4 of the canonical 8 sub-tasks); and autointerp at $N = 200$ randomly-sampled latents instead of the canonical $N = 1000$. All three rows of Table 8 use this same reduced configuration, so the diagnostic comparisons (canonical vs each degraded variant) are internally valid. As a consequence of the reduced configuration, absolute scores in Table 8 differ from those in Appendix C (5 datasets, $N = 1000$) and Appendix F (canonical configuration); for example, canonical SCR top-10 is 0.208 here versus 0.287 in the canonical-config tables. The qualitative diagnostic conclusions reported below depend on relative comparisons within the table and are unaffected by the configuration choice.

Table 5. Full reseed noise breakdown on the canonical 16k Gemma Scope SAE, five reseeds.

Benchmark	Mean	Std	CV
sae-probes ($k = 1$, accuracy)	0.763	0.001	0.1%
sae-probes ($k = 2$, accuracy)	0.782	0.001	0.1%
sae-probes ($k = 5$, accuracy)	0.806	0.001	0.2%
sparse_probing ($k = 1$, accuracy)	0.756	0.004	0.5%
sparse_probing ($k = 2$, accuracy)	0.804	0.003	0.3%
sparse_probing ($k = 5$, accuracy)	0.871	0.004	0.4%
RAVEL cause	0.655	0.008	1.2%
RAVEL disentangle	0.696	0.003	0.4%
RAVEL isolation	0.736	0.011	1.4%
autointerp random-latent ($n_{\text{latents}} = 1000$)	0.820	0.003	0.4%
autointerp fixed-latent ($n_{\text{latents}} = 1000$)	0.829	0.003	0.4%
SCR top-2	0.132	0.004	3.0%
SCR top-5	0.215	0.008	3.6%
SCR top-10	0.287	0.009	3.0%
SCR top-20	0.363	0.011	3.1%
SCR top-50	0.410	0.022	5.3%
SCR top-100	0.347	0.021	6.2%
SCR top-500	0.302	0.037	12%
TPP top-2	0.009	0.001	14%
TPP top-5	0.022	0.009	42%
TPP top-10	0.037	0.015	39%
TPP top-20	0.069	0.021	31%
TPP top-50	0.138	0.025	18%
TPP top-100	0.220	0.024	11%
TPP top-500	0.400	0.006	1.6%

Pass/fail summary. `permuted_decoder` **fails** sparse_probing, sae-probes, and autointerp (each within reseed noise of canonical, indicating these benchmarks are encoder-only by construction); **passes** TPP, SCR, and RAVEL (each crashes well outside reseed noise). `random_l0_matched` **passes** all six benchmarks (each crashes well below canonical), but the no-information floor is non-trivially high on sparse_probing (0.744) and sae-probes (0.745), so a casual benchmark consumer might mistake an L0-matched random projection for a weak but meaningful SAE.

Implementation notes. SAEBench's RAVEL hooks assume `Gemma2DecoderLayer.forward` returns a `(hidden_states,)` tuple; in transformers 5.6.2 it returns `hidden_states` directly, breaking the hook code with `IndexError: too many indices for tensor of dimension 2`. Our runner monkey-patches the hook boundary; without this, all RAVEL runs fail. Worth flagging as a SAEBench compatibility bug. The `random_l0_matched` diagnostic uses a uniform threshold across all 16,384 latents (single value), whereas canonical Gemma-Scope JumpReLU uses a learned per-feature threshold (range 2.59–16.76, mean 3.13). The deviation is intentional (the random encoder has no learned feature structure that per-feature thresholds could exploit) but is worth flagging.

F. Sanity check against public SAEBench baselines

To check that our SAEBench pipeline matches the official one, we fetched the per-SAE result JSONs published by the SAEBench authors on Hugging Face (`adamkarvonen/sae_bench_results`) for the canonical 16k Gemma Scope SAE (Gemma-2-2b layer 12, also used in Appendix C) and compared them to the means of our reseeded runs. Table 9 shows the side-by-side. Every metric matches the HF baseline to within ± 0.012 except SCR top-500 (-0.042), well within our own multi-seed standard deviation in each case.

G. SCR task construction

The SCR task generator iterates over four (T-structure, S-structure) combinations: T as a single in-sae feature or as the OR of two in-sae features, crossed with S in the same two structures. For each combination, the generator draws random in-sae feature pairs and keeps up to 3 that pass two filters: (i) T and S come from different L0 root subtrees of the SynthSAEBench-16k hierarchy (so neither sibling mutual exclusion nor parent-child gating prevents independent co-firing), and (ii) all four cells of the (T, S) firing contingency table have at least 100 samples on a 60,000-activation pool drawn for the seed. The maximum is therefore $4 \times 3 = 12$ tasks per seed; the actual count varies

Table 6. Full reseed noise breakdown on the canonical gpt2-small 32k SAE (gpt2-small layer 6 `hook_resid_post`), five reseeds. RAVEL is omitted because the SAEBench RAVEL wrapper has no gpt2 dataset.

Benchmark	Mean	Std	CV
sae-probes ($k = 1$, accuracy)	0.690	0.001	0.2%
sae-probes ($k = 2$, accuracy)	0.704	0.001	0.1%
sae-probes ($k = 5$, accuracy)	0.724	0.001	0.2%
sparse_probing ($k = 1$, accuracy)	0.700	0.006	0.9%
sparse_probing ($k = 2$, accuracy)	0.776	0.003	0.4%
sparse_probing ($k = 5$, accuracy)	0.822	0.003	0.4%
autointerp random-latent ($n_{\text{latents}} = 1000$)	0.868	0.003	0.4%
autointerp fixed-latent ($n_{\text{latents}} = 1000$)	0.867	0.002	0.3%
SCR top-2	0.064	0.004	6.8%
SCR top-5	0.092	0.005	5.8%
SCR top-10	0.128	0.005	4.2%
SCR top-20	0.181	0.004	2.2%
SCR top-50	0.231	0.008	3.3%
SCR top-100	0.250	0.008	3.3%
SCR top-500	0.222	0.060	27%
TPP top-2	0.005	0.002	29%
TPP top-5	0.012	0.003	25%
TPP top-10	0.021	0.003	14%
TPP top-20	0.039	0.001	3.5%
TPP top-50	0.088	0.006	7.3%
TPP top-100	0.147	0.009	6.0%
TPP top-500	0.337	0.009	2.6%

because random sampling occasionally fails to produce 3 viable pairs in 15 attempts (most often for the (OR, OR) combination, which has the strictest co-firing constraint). On our three task-generation seeds we obtained 9, 11, and 12 tasks respectively (Table 10).

We also attempted out-of-sae versions of these combinations (T or S as the OR of two out-of-sae features), but the (T=1 $\wedge$ S=1) cell never reached the 100-sample minimum: out-of-sae features fire too rarely on this model, and OR'ing two of them does not lift the joint rate enough. All retained SCR tasks are therefore in-sae.

H. SCR batch-size sweep on the v1 panel

SAEBench flags `probe_train_batch_size` as load-bearing for SCR. Table 11 reports Spearman ρ between SCR canonical (top-$n = 10$) and GT-MCC on the 39-SAE v1 panel under a single task-generation seed (1234), as we sweep the batch size over $\{8, 16, 32, 64\}$ with all other hyperparameters at canonical defaults. The canonical batch size 16 yields the highest ρ against both ground-truth metrics; deviating to batch 64 degrades the signal substantially. Against GT-F1 the picture is similar in shape, with batch 16 also the best. The canonical ρ here differs from Table 3 ($+0.79$ vs $+0.65$) because this sweep uses a single task-generation seed while Table 3 reports the three-seed mean.

I. Cross-base-model variation results

To probe whether the correlations reported in §5.4 reflect properties of SynthSAEBench-16k v1 specifically rather than the benchmarks themselves, we re-run the synthetic-task pipeline on three base-model variations published alongside SynthSAEBench-16k:

- `rel-p-0.5`: per-feature firing probabilities are halved relative to v1, so the activation pool is sparser overall.
- `rel-p-1.5`: per-feature firing probabilities are scaled up by $1.5\times$, so features fire more often.
- `std-2.5`: the rectified-Gaussian magnitude distribution uses a $2.5\times$ larger standard deviation, so on-firing magnitudes vary more widely.

For each variation we train a fresh 5-SAE log-uniform-L0 panel from scratch on the variation's activations, then evaluate at the same canonical hyperparameters used in §5.4. Each variation panel has 9 SAEs (5 trained + 3 degraded controls + 1 oracle) at a single task seed, so within-trained ρ is sensitive to small-panel noise; rankings are stable within each variation, however, and the disagreement between rows reflects population-level ρ on different feature distributions.

Table 12 gives the headline canonical numbers; Table 13 extends the comparison across the in / out / mixed task categories. Two findings emerge. First, SCR canonical's within-trained ρ swings from $+0.90$ on `std-2.5` to -0.10

Table 7. Full reseed noise breakdown on the canonical Llama Scope 8x SAE (Llama-3.1-8B layer 16 residual, width 32k), five reseeds. RAVEL is omitted because the SAEBench RAVEL wrapper has no Llama dataset.

Benchmark	Mean	Std	CV
sae-probes ($k = 1$, accuracy)	0.742	0.003	0.4%
sae-probes ($k = 2$, accuracy)	0.758	0.001	0.1%
sae-probes ($k = 5$, accuracy)	0.785	0.003	0.3%
sparse_probing ($k = 1$, accuracy)	0.782	0.003	0.4%
sparse_probing ($k = 2$, accuracy)	0.825	0.004	0.5%
sparse_probing ($k = 5$, accuracy)	0.869	0.004	0.4%
autointerp random-latent ($n_{\text{latents}} = 1000$)	0.860	0.004	0.4%
autointerp fixed-latent ($n_{\text{latents}} = 1000$)	0.858	0.003	0.3%
SCR top-2	0.045	0.005	11%
SCR top-5	0.079	0.007	9.0%
SCR top-10	0.121	0.006	4.6%
SCR top-20	0.163	0.005	3.2%
SCR top-50	0.182	0.005	2.8%
SCR top-100	0.227	0.007	3.1%
SCR top-500	0.241	0.007	2.8%
TPP top-2	0.008	0.002	20%
TPP top-5	0.013	0.002	14%
TPP top-10	0.017	0.003	18%
TPP top-20	0.026	0.004	14%
TPP top-50	0.045	0.003	5.7%
TPP top-100	0.062	0.003	4.3%
TPP top-500	0.127	0.004	2.9%

Table 8. Diagnostic SAE scores across the six SAEBench benchmarks under the reduced configuration described above, all on the canonical Gemma-Scope SAE. Headline metric per benchmark: `sae_top_5_test_accuracy`, `mean_k5_acc`, `tpp_threshold_10_total_metric`, `scr_metric_threshold_10`, `autointerp_score`, `disentangle_score`.

SAE / Benchmark	sparse_probing	sae-probes	TPP@10	SCR@10	autointerp	RAVEL disent.
canonical	0.858 ± 0.004	0.881 ± 0.000	0.046 ± 0.029	0.208 ± 0.025	0.790 ± 0.010	0.697 ± 0.004
`permuted_decoder`	0.857 ± 0.005	0.881 ± 0.000	0.004 ± 0.008	-0.002 ± 0.039	0.790 ± 0.005	0.290 ± 0.002
`random_10_matched`	0.744 ± 0.008	0.745 ± 0.003	0.002 ± 0.004	-0.002 ± 0.001	0.694 ± 0.012	0.291 ± 0.002

on `rel-p-1.5`, far beyond what task-sampling noise on a single underlying ρ can explain, and the -0.10 value confirms that SCR can read negatively correlated with quality on a perfectly reasonable variant of the same base model. Second, sparse probing's calibration on out-of-sae features is itself firing-rate-dependent: single out-of-sae ρ is $+0.13$ on `rel-p-0.5` (preserved), $+0.23$ on v1, and $+0.67$–$+0.73$ on the higher-firing-rate `rel-p-1.5` and `std-2.5` variations. When out-of-sae features fire often enough, hidden-space proxies for them become accessible to a probe via correlated in-sae latents, so the agnosticism we report on v1 is a property of v1's firing-rate regime rather than a guarantee. The $+0.00$ within-trained value for sparse probing on `rel-p-0.5` in Table 12 is dominated by a non-monotone GT-MCC-vs-L0 curve on that variation, not a calibration failure.

J. Ground-truth F1 and MCC vs L0 per SAE architecture

Figure 5 plots GT-F1 and GT-MCC as a function of measured L0 for each of the five SAE architectures in the v1 panel (Standard, JumpReLU, BatchTopK, Matryoshka, Matching Pursuit), at the seven trained L0 targets ($\{15, 20, 25, 30, 35, 40, 45\}$). GT-F1 declines monotonically with L0 on Standard, JumpReLU, BatchTopK, and Matching Pursuit, peaks around L0$\approx$ 25–30 on Matryoshka, and is everywhere noticeably below 1 even on the best-performing variants. GT-MCC, in contrast, either rises gently with L0 or plateaus, and stays relatively high (above 0.55) across the full L0 range. This is a property of the two metrics rather than of the SAEs: GT-F1 penalizes over-firing latents (any latent that fires when its target ground-truth feature is off lowers precision), so increasing L0 past the underlying feature firing rate degrades F1; GT-MCC tolerates over-firing as long as the encoder still discriminates target-feature presence above noise, so it is much less sensitive to the L0-target axis. Because the proxy benchmarks studied in this paper consume only the encoder's output,

Table 9. Side-by-side comparison of our reseeded means against the public SAEBench result JSONs for the canonical 16k Gemma Scope SAE. Sanity check that our pipeline reproduces the original work; differences are within reseed noise.

Benchmark	Ours (mean)	HF baseline	Ours − baseline
sparse_probing top-1 acc	0.756	0.760	−0.003
sparse_probing top-5 acc	0.871	0.878	−0.007
TPP top-2	0.009	0.009	+0.001
TPP top-10	0.037	0.025	+0.012
TPP top-50	0.138	0.140	−0.002
TPP top-500	0.400	0.394	+0.005
SCR top-10	0.287	0.284	+0.004
SCR top-500	0.302	0.344	−0.042
autointerp score	0.820	0.828	−0.008

Table 10. Number of SCR tasks kept per seed, broken down by the structures of T and S. Each (T-structure, S-structure) combination targets 3 tasks but is capped at the number of random feature pairs (up to 15 draws) that satisfy both the different-root constraint and the ≥ 100 samples per contingency cell minimum.

seed	(single, single)	(single, or)	(or, single)	(or, or)	total
1234	3	3	3	0	9
2222	3	3	2	3	11
3333	3	3	3	3	12

sparse probing's better correlation with GT-MCC than GT-F1 (Table 3) is consistent with the proxy and ground-truth metric agreeing on what they tolerate.

K. Hyperparameter sweep for sparse probing, TPP, and SCR

We calculate Spearman ρ between benchmark score and GT-MCC across the v1 synthetic panel as a function of the benchmark's primary hyperparameter (top-k for sparse probing, top-n for TPP and SCR), and show results in Figure 6.

L. Per-task / per-hyperparameter validity scatter grids

Figure 4 in §5 shows benchmark score vs GT-MCC at canonical hyperparameters for one task category each. For completeness, this appendix gives the full (category $\times$ hparam) grid of scatter panels for sparse probing (Figure 7), TPP (Figure 8), and SCR (Figure 9). Single task-generation seed (1234); each panel reports its multi-seed mean Spearman ρ where multi-seed data exists, and its single-seed Spearman otherwise. Canonical-hparam columns (top-$k = 16$ for sparse probing, top-$n = 10$ for TPP, and top-$n \in \{10, 50, 500\}$ for SCR) use the full 35-trained-SAE panel; the other columns use the 15-trained-SAE sub-panel.

M. Practitioner summary across the suite

We provide a Practitioner summary across the SAEBench metrics audited in this study in Table 14.

N. SAE training setup for the discriminability panels

The two panels of trained SAEs analyzed in §4 (cross-architecture and log-uniform-prefix Matryoshka) share most training hyperparameters and differ only in architecture-specific configuration. We document them here to support reproducibility and to explain the log-uniform-prefix Matryoshka variant, which is non-standard.

Common settings. All SAEs are trained on the residual stream after layer 12 of `google/gemma-2-2b` (`blocks.12.hook_resid_post`), $d_{\text{in}} = 2304$, on The Pile (Gao et al., 2020) with context length 1024. Every SAE has $d_{\text{sae}} = 32{,}768$ ($16 \times d_{\text{in}}$). We use Adam with $\beta_1 = 0.9$ (SAELens defaults: $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay 0), peak learning rate 3×10^{-4}, no warm-up, and a linear LR decay over the final fifth of training. Training batch size is 4,096 tokens. Activations are accessed through `transformer_lens` with `exclude_special_tokens=True` and `prepend_bos=True`. Mixed-precision (bf16 autocast) is enabled for both the language model and the SAE. The cross-architecture panel uses seed 0 for every variant; the sampled-Matryoshka panel uses seeds $\{0, 1, 2\}$ for every variant (Appendix S).

Cross-architecture panel (1.5B tokens). Four SAEs trained for 1.5×10^9 tokens each (366,210 steps; LR decays over the final 3×10^8 tokens). The four variants differ only in the sparsifying activation:

- **BatchTopK** ($k \in \{50, 100\}$), one decoder; the global

Table 11. SCR canonical (top-$n = 10$, `probe_l1_penalty`=10^{-3}) Spearman correlation with GT-MCC and GT-F1 across the 39-SAE v1 panel as a function of `probe_train_batch_size`. Single task-generation seed (1234). The canonical batch size 16 maximizes both signals.

`probe_train_batch_size`	ρ vs GT-MCC	ρ vs GT-F1
8	+0.72	+0.65
16 (canonical)	**+0.79**	**+0.70**
32	+0.70	+0.61
64	+0.54	+0.49

Table 12. Cross-base-model behavior of three benchmarks at canonical hyperparameters with GT-MCC. Each cell is full-panel ρ / within-trained-SAE ρ. SCR canonical's within-trained ρ swings from $+0.90$ to -0.10 across base models, beyond what task-sampling noise on a single underlying ρ can explain. The $+0.00$ on `rel-p-0.5` for sparse probing is dominated by a non-monotone GT-MCC-vs-L0 curve on that variation rather than a calibration failure.

Benchmark	v1 (3 seeds)	`std-2.5` (1 seed)	`rel-p-0.5` (1 seed)	`rel-p-1.5` (1 seed)
Sparse probing single in-sae (top-16)	$+0.58/+0.49$	$+0.98/+0.90$	$+0.68/+0.00$	$+1.00/+1.00$
TPP all out-of-sae (top-10)	$+0.38/+0.42$	$-0.03/+0.90$	$+0.17/+0.60$	$+0.18/+1.00$
SCR canonical (top-10)	$\mathbf{+0.64/+0.65}$	$\mathbf{+0.64/+0.90}$	$\mathbf{+0.55/+0.30}$	$\mathbf{+0.13/-0.10}$

top-$k \cdot B$ activations across the batch are kept active per step (Bussmann et al., 2024).

- **Matryoshka BatchTopK** ($k \in \{50, 100\}$) with three nested inner widths {2,048, 8,192, 32,768} (i.e. $d_{\text{sae}}/16$, $d_{\text{sae}}/4$, d_{sae}). At each step the loss is the sum of BatchTopK reconstruction losses over the three prefixes (and the auxiliary dead-feature term), so each prefix learns to be a self-contained dictionary (Bussmann et al., 2025). Inner-width selection is fixed across training steps.

We snapshot 28 times per SAE (token counts $0, 10\text{M}, 25\text{M}, 50\text{M}, 100\text{M}, 150\text{M}, \ldots, 1.0\text{B}, \ldots, 1.5\text{B}$). The token-0 (random-init) snapshot is dropped from all training-trajectory analyses.

Log-uniform-prefix Matryoshka panel (300M tokens). Four Matryoshka BatchTopK SAEs ($k = 100$) per training seed, trained for 3×10^8 tokens each (73,242 steps; LR decays over the final 6×10^7 tokens). Variants differ only in $n \in \{1, 2, 3, 4\}$, the number of inner-width prefixes sampled per training step. We train each variant from three independent random seeds (0, 1, 2), giving $4 \times 3 = 12$ SAE training runs in total (Appendix S). Each (variant, seed) takes 11 snapshots at $\{0, 30\text{M}, 60\text{M}, \ldots, 300\text{M}\}$ tokens.

The non-standard part is the inner-width selection rule. Rather than a fixed prefix list, at each training step we draw n widths from a log-uniform distribution and use them as that step's matryoshka prefixes (always including the full d_{sae} as the outermost prefix):

$$\log(w_i) \sim \text{Uniform}(\log w_{\min},\ \log w_{\max})$$

Where $\quad w_{\min} = 64, \quad w_{\max} = d_{\text{sae}} = 32{,}768.$

The n samples are deduplicated, sorted, and clamped to $[w_{\min}, w_{\max} - 1]$, and the Matryoshka loss is computed against the resulting prefixes for that step only. Different steps see different prefix sets, so the SAE is trained as a Matryoshka over the entire log-spaced range $[64, 32{,}768]$ rather than over a small fixed prefix list. A higher n trades wall-clock speed for variance reduction in the per-step prefix coverage. To our knowledge the same log-uniform prefix-sampling scheme is used by Gemma Scope 2's Matryoshka SAEs (McDougall et al., 2025); we adopt it here to mirror that setup.

The intuition behind log-uniform sampling is that "how good is this SAE at a given prefix width w?" is most naturally a question about $\log w$ rather than w: doubling w from 64 to 128 changes capacity as much as doubling from 16,384 to 32,768. Log-uniform sampling spreads training signal evenly across that scale rather than concentrating it on widths near d_{sae}, which a uniform-in-w rule would do.

O. Compute resources

All experiments were run on a single NVIDIA H100 (80GB) GPU at a time; we did not parallelize a single experiment across multiple GPUs. Wall-clock numbers below are approximate, since (a) several experiments were re-run for the L0 fill-in and the reseed sweep, and (b) we share H100s with other users.

LLM SAE training (Section 4).

- *Cross-architecture panel*: each of the 4 SAEs (1.5B tokens) takes ∼12 hours on an H100 ⇒ ∼48 H100-hours for the panel (single training seed).

Table 13. Full-panel Spearman ρ between each benchmark category and GT-MCC across the four base-model variations, at canonical hyperparameters (top-$k = 16$ for sparse probing; top-$n = 10$ for TPP and SCR). Sparse probing single in-sae stays positive across all four variations, but single out-of-sae ρ rises sharply on the higher-firing-rate variations (`rel-p-1.5`, `std-2.5`), indicating that sparse probing's silence on out-of-sae features depends on the base model's firing-rate regime. SCR canonical is positive on three variations and weakly positive on `rel-p-1.5`, but its within-trained ρ on `rel-p-1.5` (Table 12) is the headline failure.

Benchmark category	v1	`std-2.5`	`rel-p-0.5`	`rel-p-1.5`
Sparse probing single in-sae	+0.58	+0.98	+0.68	+1.00
Sparse probing single out-of-sae	+0.23	+0.67	+0.13	+0.73
TPP all in-sae	+0.18	+0.75	+0.65	+0.77
TPP all out-of-sae	+0.38	−0.03	+0.17	+0.18
SCR canonical (T=in, S=in)	+0.64	+0.64	+0.55	+0.13

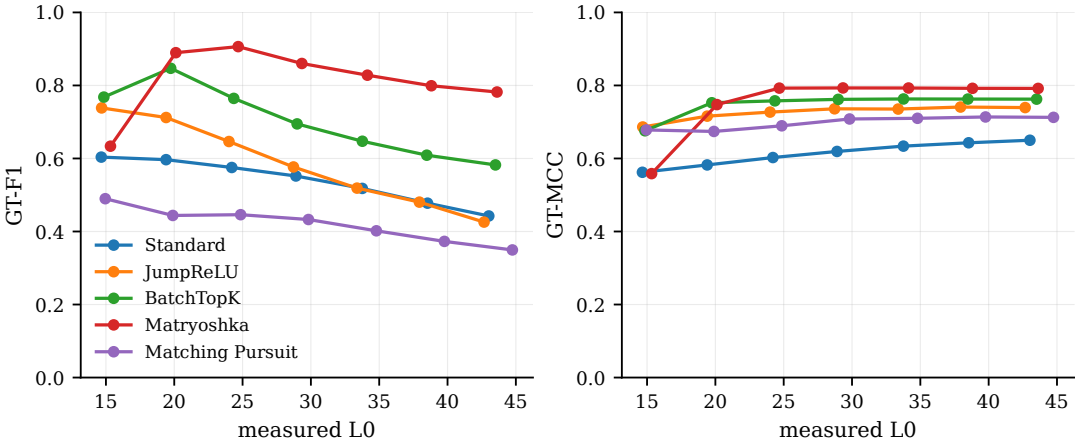

Figure 5. GT-F1 (left) and GT-MCC (right) vs measured L0 for each SAE architecture in the v1 panel. Each point is one trained SAE (mean across task-generation seeds). GT-F1 has a clear optimum or monotone decay; GT-MCC stays high or rises gently across the same L0 range.

- *Sampled-Matryoshka panel*: each of the $4 \times 3 = 12$ SAEs (4 variants $\times$ 3 training seeds, 300M tokens each) takes ∼3 hours on an H100 ⇒ ∼36 H100-hours for the panel.

Synthetic-data SAE training (Section 5). The 35 SAEs trained on SynthSAEBench-16k (5 architectures $\times$ 7 L0 levels, 200M synthetic samples each, $d_{\text{sae}} = 4096$) are much faster than the LLM SAEs because there is no language-model forward pass. Each takes roughly 20–30 minutes on an H100; the full panel is well under 20 H100-hours.

SAEBench evaluations (Sections 3, 4). Per-SAE wall clock varies widely across evaluations: sparse probing, TPP, SCR, and RAVEL are each 20–40 minutes on an H100; autointerp is mostly bound by OpenAI API latency rather than H100 time; core and sae-probes are 5–15 minutes once the LM activations are cached. The reseed-noise study (§3) covers 4 canonical SAEs $\times$ 5 reseeds $\times$ 6 evaluations and totals ∼40 H100-hours plus ∼$30 of OpenAI API spend on autointerp. The snapshot evaluations dominate total compute: 28 snapshots per cross-architecture SAE and 11 snapshots per sampled-Matryoshka SAE, $\times$ 6 evaluations across the $4 + 12 = 16$ trained SAEs (4 cross-architecture and 12 sampled-Matryoshka after multi-seed expansion), comes to ∼500–650 H100-hours.

Synthetic-validity evaluations (Section 5). Each of the 40 SAEs in the synthetic panel (across the v1 and three SAE-variation panels) is encoded once on a 60,000-activation pool and then run through 92 sparse-probing tasks, 60 TPP sibling groups, and up to 12 SCR pairs per task-generation seed. Per SAE this is well under an hour on an H100; across 3 task seeds for v1 and a single seed for each variation, the panel costs ∼60 H100-hours total.

Preliminary and exploratory experiments. The numbers above are for the experiments reported in this paper. The full project also included exploratory runs (alternative L0 schedules, alternative reseed-noise SAEs, batch-size and probe-regularizer sweeps, hyperparameter ablations) that we did not include because they did not change the qualitative findings. We estimate the total project compute at roughly 2–3$\times$ the above, dominated again by SAEBench snapshot evaluations on intermediate SAE checkpoints.

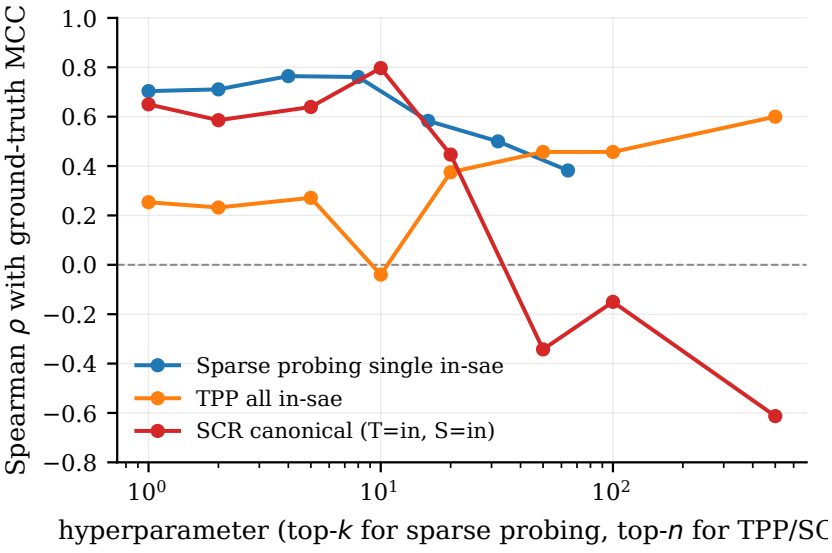

Figure 6. Spearman ρ between benchmark score and GT-MCC across the v1 synthetic panel as a function of the benchmark's primary hyperparameter (top-k for sparse probing, top-n for TPP and SCR), single seed (1234). Canonical-hparam points (top-$k = 16$ for sparse probing, top-$n \in \{10, 50, 500\}$ for SCR, top-$n = 10$ for TPP) use the full 35-trained-SAE panel; non-canonical points use the 15-trained-SAE sub-panel from the original L0 set $\{15, 25, 45\}$. Sparse probing stays positive across the full range of k. SCR's correlation flips from positive at top-$n \leq 20$ to strongly negative at top-$n = 500$ ($\rho = -0.61$ on the 35-SAE panel). TPP's correlation in-sae stays in the -0.04 to $+0.60$ range and dips to near zero at canonical top-$n = 10$.

P. Probe-ablation training-trajectory slopes

Table 15 reports linear-regression slopes of TPP and SCR scores against training tokens (in score units per billion tokens, after a 100M-token warmup) for every threshold across the four-SAE training panel. Positive values indicate improvement through training; negative values indicate decline. TPP shows a clean cross-over between top-20 and top-50 (positive slopes for all four variants below the cross-over, negative for all four above). SCR slopes are negative across most thresholds for both BTK variants and at top-$N \geq 50$ for the Matryoshka variants.

Q. SNR ranking of SAEBench metrics on the training panels

Definition. We summarize each metric by an ANOVA-style signal-to-noise ratio: the standard deviation across variants of each variant's mean score after a 100M-token warm-up (between-variant variation), divided by the pooled standard deviation of detrended within-variant residuals (within-variant noise, with a degree-2 polynomial fit removing monotone trajectory motion). $\mathrm{SNR} = 1$ means a metric's spread across variants matches its own per-checkpoint jitter; $\mathrm{SNR} \geq 2$ means a metric reads a comfortably stable inter-variant signal.

Results. Figure 10 shows the SNR ranking for every SAEBench metric on the four-SAE architecture panel. Core reconstruction metrics dominate (MSE and explained variance both at SNR 14.2, KL preservation 10.4, CE preservation 8.4). Among non-core metrics, sae-probes is the most discriminative (SNR 5.4–6.6 across $k = 1 \ldots 16$). Default SAEBench sparse probing is roughly two-thirds as discriminative (top-1 SNR 4.2, degrading to 2.8 at top-10). SCR and TPP read modest signals at moderate thresholds (SCR top-$\{2, 5, 10, 20\}$: 3.5–3.9; TPP top-$\{50, 100\}$: 3.0–3.9), while several published metrics have SNR ≤ 2: RAVEL cause/isolation (1.0–0.7), autointerp (1.8), TPP top-$\{5, 10, 500\}$ (1.1–1.6), SCR top-$\{50, 100, 500\}$ (1.5). For these latter metrics inter-variant spread is comparable to per-checkpoint jitter, so picking a winner between SAEs in this panel from one checkpoint is roughly a coin flip.

We do not show an analogous SNR ranking for the sampled-Matryoshka panel: with three training seeds available there, the variance decomposition in Appendix S (Figure 12) cleanly separates between-variant signal from training-seed and snapshot noise, which a single-number SNR summary cannot. The qualitative ordering on the Matryoshka panel is broadly consistent with the architectures panel (core reconstruction at the top; sae-probes and RAVEL disentangle next; SCR and TPP at the bottom), though variance shares are much smaller because the four Matryoshka SAEs are closer in absolute score than the four arch-panel SAEs. We do still compute the same SNR statistic on the Matryoshka panel and use it as the SNR-≥ 1 filter for the cross-metric ranking heatmap (Appendix R, Figure 11); we just do not

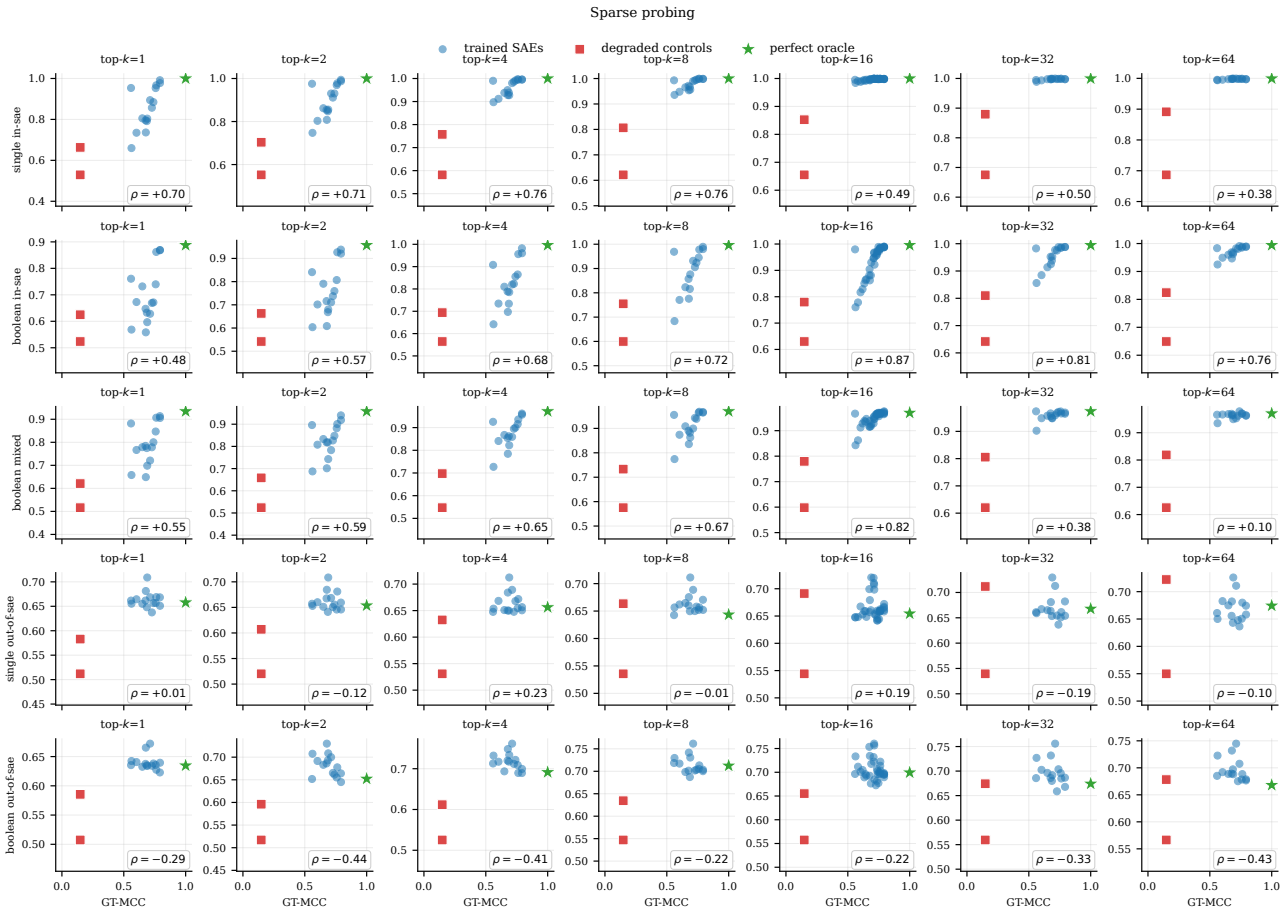

Figure 7. Sparse probing: GT-MCC vs benchmark score across all task categories (rows) and top-k values (columns). Decoder-only degraded controls (`permuted_decoder`) are excluded because sparse probing is encoder-only and would score them identically to the canonical SAE. The boolean-mixed row and the $k = 16$ column correspond to the panel shown in Figure 4.

plot the resulting ordering separately because Figure 12 subsumes it.

R. Cross-metric ranking agreement (contradiction matrices)

For each panel we compute pairwise Spearman ρ of metric rankings using each metric's mean trajectory value per variant (post-warmup), oriented so higher = better SAE. We restrict to SNR-informative metrics (SNR ≥ 1, see Appendix Q). Figure 11 shows a representative subset of nine metrics on each panel; the title of each subplot reports the mean off-diagonal ρ across the full SNR-informative metric set on that panel.

The cross-architecture panel (left) shows broad agreement (mean $\rho = +0.44$ across 30 informative metrics): when SAEs are very different from each other, most SAEBench metrics rank them similarly. The Matryoshka panel (right) shows essentially no agreement (mean $\rho = +0.08$ across 17 informative metrics) and several stark contradictions, including default sparse probing k=1 vs k=10 negatively correlated at $\rho = -0.4$ (a single metric family disagreeing with itself), and SCR top-100 disagreeing with core MSE, RAVEL disentangle, and SCR top-$\{2, 10, 500\}$ at $\rho = -0.2$ to -0.4. The two panels combined: SAEBench reliably ranks SAEs only when they are obviously different.

S. Multi-seed variance decomposition on the Matryoshka panel

This appendix gives the multi-seed analysis of the four-variant sampled-Matryoshka panel from §4. Each variant ($n \in \{1, 2, 3, 4\}$) was trained from three independent random seeds (0, 1, 2), all other settings as in Appendix N. After dropping the token-0 random-init snapshot and applying a 60M-token warm-up, each (variant, seed) trajectory contributes 9 snapshot scores per metric, for a total of $4 \times 3 \times 9 = 108$ post-warmup observations per metric. Throughout this appendix "audited metrics" refers to the 34

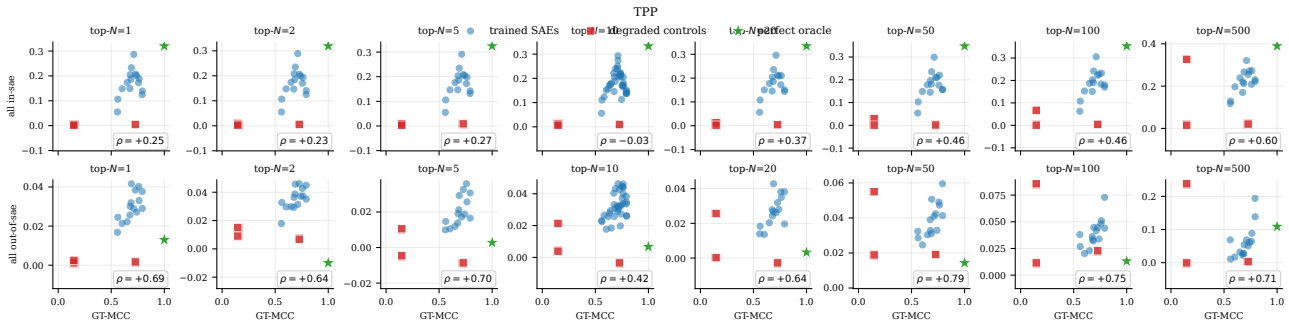

Figure 8. TPP: GT-MCC vs benchmark score across all-in-sae and all-out-of-sae sibling groups (rows) and top-n values (columns). The all-in-sae row at $n = 10$ corresponds to the panel shown in Figure 4; out-of-sae correlations are higher than in-sae correlations across the entire n range, the "leak" discussed in §5.4.

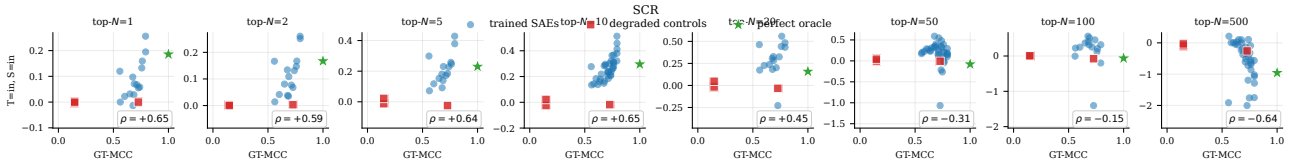

Figure 9. SCR (T=in, S=in): GT-MCC vs benchmark score across top-n values. The $n = 10$ panel corresponds to Figure 4; correlations swing from positive at small n to strongly negative by $n = 500$.

SAEBench scores we report on this panel.

S.1. Variance decomposition

Definitions. For each metric, let $x_{v,s,t}$ denote the post-warmup score at variant v, seed s, snapshot t. We compute three variance components,

$$V_{\text{variant}} = \mathrm{Var}_v\Big(\mathrm{mean}_{s,t}\, x_{v,s,t}\Big),$$
$$V_{\text{seed}-\text{variant}} = \mathrm{mean}_v\Big(\mathrm{Var}_s\big(\mathrm{mean}_t\, x_{v,s,t}\big)\Big),$$
$$V_{\text{snap}} = \mathrm{mean}_{v,s}\Big(\mathrm{Var}_t\, x_{v,s,t}\Big),$$

i.e. the variance of variant means (signal), the average across variants of variance across seeds within a variant (training-seed noise), and the average within-trajectory variance (snapshot wiggle). We report each as a share of the sum, e.g. $\mathrm{share}_{\text{variant}} = V_{\text{variant}}/(V_{\text{variant}} + V_{\text{seed}-\text{variant}} + V_{\text{snap}})$. This share has a similar functional form to a one-way random-effects intraclass correlation coefficient but is not equivalent; we report it as a variance share.

Null calibration. Under i.i.d. standard-normal noise on the same panel design (4 variants $\times$ 3 seeds $\times$ 9 snapshots), $\mathrm{share}_{\text{variant}}$ is bounded above zero by chance alone. A 5,000-trial Monte-Carlo simulation gives $\Pr(\mathrm{share}_{\text{variant}} > 0.20) < 10^{-3}$, $\Pr(\mathrm{share}_{\text{variant}} > 0.10) \approx 0.02$, and $\Pr(\mathrm{share}_{\text{variant}} > 0.05) \approx 0.21$. We treat $\mathrm{share}_{\text{variant}} > 0.20$ as reliable evidence of inter-variant signal, $[0.10, 0.20]$ as weak evidence, and < 0.10 as consistent with no detectable signal on this panel.

Results. Table 16 reports the variance-share decomposition for the 34 audited metrics, sorted by $\mathrm{share}_{\text{variant}}$. Figure 12 shows the same data as a stacked bar. Seven metrics clear the 0.20 bar: the four core reconstruction metrics (shares 0.39–0.78), sae-probes at $k \in \{1, 5\}$, and RAVEL disentangle. Twelve more metrics fall in $[0.10, 0.20]$. The remaining fifteen metrics fall below 0.10, including the three SCR thresholds at ≥ 50, all four TPP thresholds at ≥ 20, sparse probing top-$\{1, 5\}$, sparse probing overall, RAVEL cause, and TPP top-5. The canonical SCR top-10 and TPP top-10 thresholds sit at the boundary (0.09 and 0.10 respectively), with most of their score variance attributable to seed and snapshot noise rather than to the SAE variant.

S.2. Per-seed winner stability

For each metric we computed the post-warmup mean per (variant, seed) and picked the variant with the best score (per the metric's natural "higher = better" or "lower = better" orientation) within each seed. Table 17 summarises how many distinct variants are crowned across the three seeds.

Three-distinct winners. Nine audited metrics produce a different best variant on each of the three seeds: RAVEL cause, sae-probes $k = 16$, SCR top-5, SCR top-10, SCR top-50, sparse probing top-1, TPP top-10, TPP top-20, and TPP top-100. The canonical-batch SCR top-10 and TPP top-$\{10, 20\}$ are in this group, so a single-seed report at these

Table 14. Practitioner summary across the SAEBench metrics audited in this study. Reseed CV is from the canonical 65k Gemma Scope SAE (Table 1); see Appendix C for the 16k SAE. Validity ρ is the multi-seed mean Spearman correlation with GT-MCC across the 35-trained-SAE v1 panel (degraded controls and the oracle excluded). SNR is the ANOVA-style between/within ratio on the 4-SAE architecture training panel (see Appendix Q, Figure 10); ranges span hyperparameter sweeps within the metric.

Metric	Reseed CV (§3)	Validity ρ (§5)	SNR (§4)	Direction with training
Report by default.				
core (MSE / EV / KL / CE)	–	–	8–14	monotone improving
sae-probes (k=1…16)	0.2%	high	5–7	monotone improving
sparse probing (top-k)	0.3–1.2%	$+0.49$ to $+0.87$	1.8–4.2	monotone improving
RAVEL disentangle	0.2%	–	2.6	monotone improving
autointerp (n_{lat}=1k)	0.5%	–	1.8	monotone improving
Avoid: sign-flips, low SNR, or declines with training.				
SCR	3–7%	-0.64 to $+0.65$	1.5–3.9	declines on 3/4 variants
TPP	3–39%	≈ 0	1.1–3.9	improves low N, declines high N
RAVEL cause / isolation	2–3%	–	0.7–1.0	coin-flip on snapshots

Table 15. Slopes of TPP and SCR scores against training tokens (score per B tokens after a 100M warmup) on the four-SAE training panel. TPP shows a clean cross-over between top-20 and top-50: positive (improving) for all four variants at top-$N \leq 10$ and at top-$N = 20$ for three of four, then negative (declining) for all four at top-$N \geq 50$. SCR is negative across most thresholds for the BTK variants and at top-$N \geq 50$ for the Matryoshka variants.

TPP top-N	2	5	10	20	50	100	500
BTK k=50	$+0.039$	$+0.057$	$+0.010$	-0.017	-0.048	-0.046	-0.030
BTK k=100	$+0.016$	$+0.048$	$+0.028$	$+0.014$	-0.029	-0.056	-0.062
Matryoshka k=50	$+0.037$	$+0.025$	$+0.025$	$+0.002$	-0.021	-0.028	-0.026
Matryoshka k=100	$+0.019$	$+0.054$	$+0.039$	$+0.007$	-0.021	-0.029	-0.046
SCR top-N	2	5	10	20	50	100	500
BTK k=50	-0.040	-0.028	-0.065	-0.065	-0.077	-0.083	-0.038
BTK k=100	-0.019	-0.030	-0.033	-0.070	-0.066	-0.103	-0.060
Matryoshka k=50	-0.009	$+0.022$	$+0.005$	$+0.008$	-0.025	-0.046	-0.066
Matryoshka k=100	$+0.003$	-0.003	-0.006	$+0.009$	-0.016	$+0.047$	-0.004

canonical thresholds amounts to picking among three plausible winners. This count itself is not statistically beyond a uniform-random-winner null (Pr($\geq$ 3 distinct) ≈ 0.38 for 4 variants and 3 seeds), so we read it as a concrete illustration of the low-$\text{share}_{\text{variant}}$ result, not as independent evidence.

Single-winner metrics. Nine audited metrics agree on a single best variant across all three seeds: the four core reconstruction metrics (all pick $n=1$), core KL divergence ($n = 1$), RAVEL isolation ($n = 1$), TPP top-500 ($n = 1$, where the score declines through training so "best" is the least-degraded variant), sparse probing overall ($n=1$, the metric is independent of top-k selection), and sae-probes $k=5$ ($n=2$). The remaining sixteen audited metrics pick two distinct winners across the three seeds.

S.3. Cross-seed ranking agreement

For each metric and each pair of seeds we computed the Spearman ρ between the two seeds' rankings of the four variants by post-warmup mean, then averaged across the three pairs. Table 18 reports per-metric mean ρ, sorted descending.

The four core reconstruction metrics all reach $\rho = 1.0$ (perfect cross-seed agreement on the variant order $n=1 > n=2 > n=3 > n=4$, in MSE-direction). Core KL and TPP top-500 reach $\rho = 0.87$, also a strong rank-stability signal. Below $\rho \approx 0.5$ the picture decays into the regime where small perturbations to the per-variant means flip the ranking, and below $\rho \approx 0$ the cross-seed ranking is no better than chance.

The lower end of Table 18 shows several metrics with negative mean ρ, including SCR canonical (top-10, $\rho = -0.20$) and TPP canonical (top-10 $\rho = -0.20$, top-20 $\rho = -0.33$). With only 4 variants, only 11 distinct ρ values are possible per pair, and a uniform-random-rankings null has Pr(mean $\rho < 0) \approx 0.55$ and Pr(mean $\rho < -0.33) \approx 0.27$ across 3 seed pairs. We therefore do not interpret negative mean ρ as anti-correlation; the informative reading is that these metrics produce no detectable consistent ranking across seeds, while the metrics at the top of the table do produce a stable ranking.

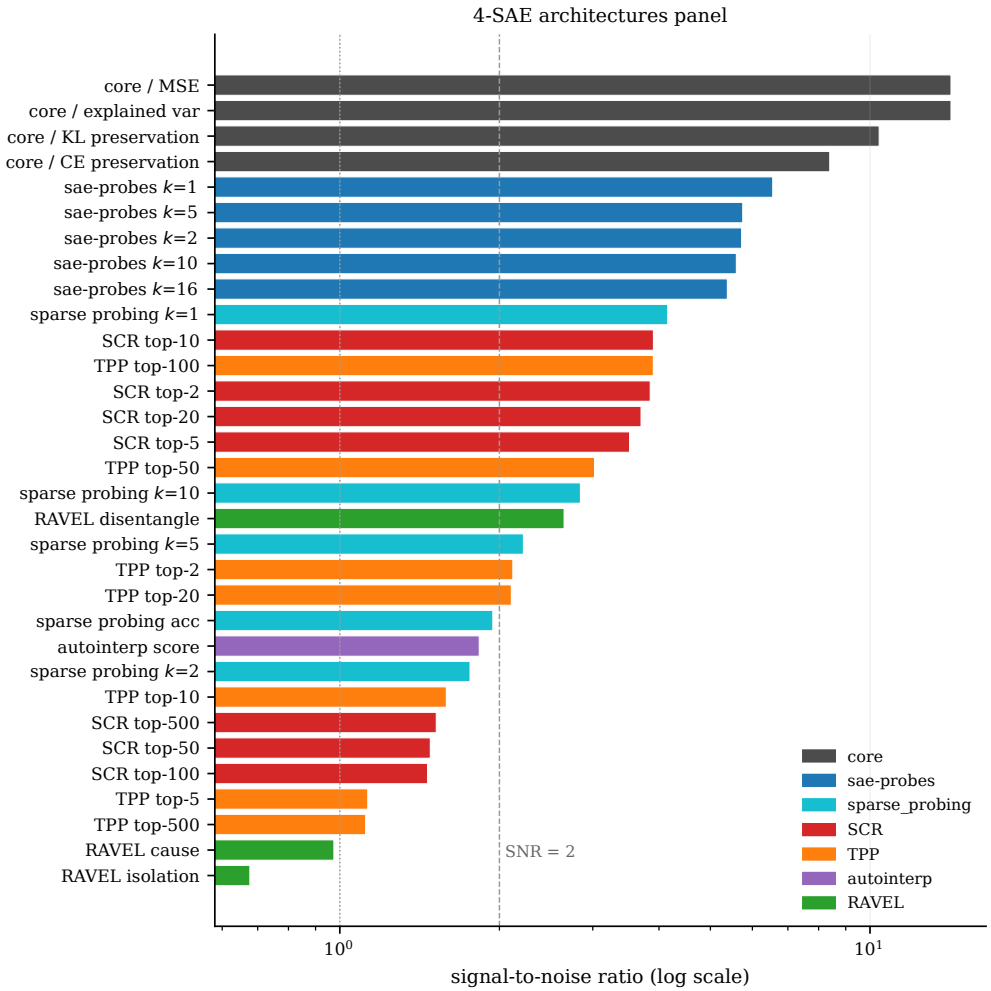

Figure 10. Signal-to-noise ratio of every SAEBench metric on the four-SAE architecture snapshot panel, ordered by SNR (log scale). Reference dashed line at SNR $= 2$. The Matryoshka-panel analogue is the variance decomposition in Figure 12.

S.4. Caveats

Three seeds is the minimum that supports a variance decomposition; the between-seed-within-variant variance is itself estimated from only 3 within-variant samples per metric, so individual share values are imprecise. The qualitative ordering (which metrics clear $\text{share}_{\text{variant}} > 0.20$) is robust under bootstrap; specific share values should not be over-interpreted, especially in the $[0.05, 0.20]$ band.

The within-trajectory variance term V_{snap} includes any monotone trajectory motion within the post-warmup window, so for metrics that improve through training (e.g. core MSE) some of V_{snap} is real quality drift rather than jitter, biasing $\text{share}_{\text{variant}}$ down for those metrics. Even with this bias, all four core reconstruction metrics and three encoder-only metrics (sae-probes $k \in \{1, 5\}$, RAVEL disentangle) clear the 0.20 bar, so the qualitative ordering at the top of the table is robust.

We did not collect multi-seed data for the cross-architecture panel, since we train for so much longer (1.5B tokens vs 300M tokens) on that panel making running multiple seeds too expensive. That panel's variants differ much more than the Matryoshka panel's, so per-variant signal is large enough that a single-seed analysis broadly suffices, but a multi-seed cross-architecture replication remains useful future work.

T. Full-metric training trajectories

Figure 1 in the main text plots ten representative metrics across training. For completeness, this appendix gives the trajectories of every SAEBench metric across the same training runs, both for the four-SAE architecture sweep (BatchTopK $k \in \{50, 100\}$, Matryoshka $k \in \{50, 100\}$; Figure 13) and for the four-SAE sampled-Matryoshka hyperparameter sweep ($n \in \{1, 2, 3, 4\}$, varying the number of inner widths sampled per training step; Figure 14). Both

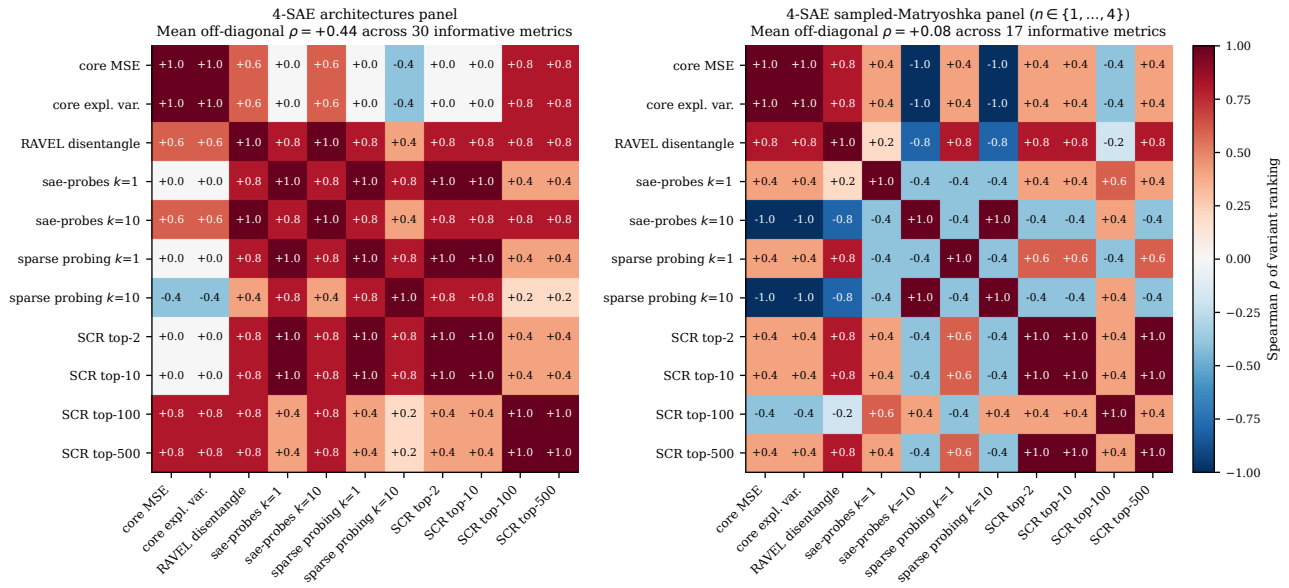

Figure 11. Pairwise Spearman ρ between SAEBench metric rankings on the four-SAE architectures panel (left) and the four-SAE sampled-Matryoshka panel (right; $n \in \{1, 2, 3, 4\}$). Each cell is the rank correlation between two metrics' final-snapshot rankings of the four SAEs in that panel, oriented so positive ρ means agreement on which SAE is better. Subtitle reports the mean off-diagonal ρ across all SNR-informative metrics (SNR ≥ 1); the displayed subset of nine metrics is representative. The Matryoshka panel uses seed 0; the multi-seed analysis is in Appendix S.

panels share the same metric layout: core reconstruction (row 1), encoder-only quality (sae-probes, autointerp, and RAVEL disentangle on row 2), default sparse probing and the remaining RAVEL metrics (row 3), TPP across all thresholds (row 4), and SCR across all thresholds (row 5).

The four-SAE arch panel (Figure 13) shows that core, sae-probes, autointerp, RAVEL disentangle, and default sparse probing all rise monotonically through training for every variant; TPP improves at low thresholds (top-$N \leq 20$) and declines at high thresholds (top-$N \geq 50$) for every variant, with the cross-over visible by inspection; SCR declines at most thresholds for both BTK variants and at ≥ 50 for the Matryoshka variants. RAVEL cause and isolation are nearly flat across training. The four-SAE sampled-Matryoshka panel (Figure 14) compresses to 300M tokens and is noisier, but the same TPP cross-over and SCR-at-large-N decline reproduce.

U. GT-F1 correlations

In §5 we mainly focus on correlations against GT-MCC because most metrics seem better correlated to GT-MCC rather than GT-F1, and we only consider it necessary for a proxy metric to correlate to either GT-MCC or GT-F1. Table 19 reports the analogous correlations against GT-F1 for the headline benchmarks across the v1 panel; both full-panel and within-trained-SAE values are shown. Finding a proxy metric that correlates with GT-F1 would be a good direction for future SAE benchmarking work.

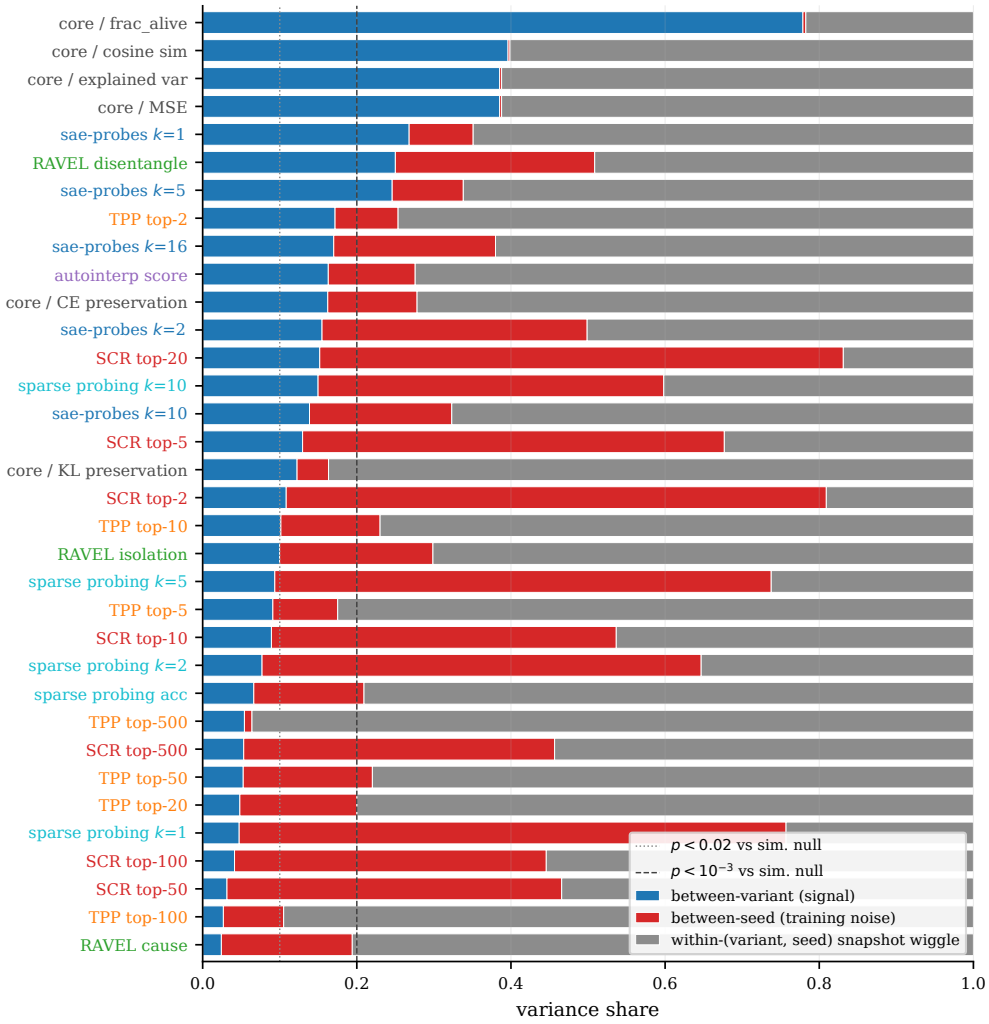

Figure 12. Per-metric variance decomposition on the four-variant Matryoshka panel (3 seeds $\times$ 9 post-warmup snapshots per (variant, seed)), sorted by between-variant variance share. The four core reconstruction metrics, sae-probes $k \in \{1, 5\}$, and RAVEL disentangle are the only metrics where the SAE variant accounts for more than 0.20 of total variance (dashed line: $p < 10^{-3}$ vs the simulated null; dotted line: $p < 0.02$). Y-axis labels are coloured by metric family (gray = core, blue = sae-probes, cyan = sparse probing, red = SCR, orange = TPP, purple = autointerp, green = RAVEL).

Table 16. Variance-share decomposition for each of the 34 audited SAEBench metrics on the Matryoshka panel, sorted by between-variant share (signal). The right-hand columns are the share of total variance attributable to between-training-seed-within-variant noise and within-(variant, seed) between-snapshot wiggle. The horizontal rules separate metrics with reliable inter-variant signal ($\mathrm{share}_{\text{variant}} > 0.20$, $p < 10^{-3}$ vs the simulated null), weak evidence ($[0.10, 0.20]$, $p < 0.02$), and no detectable signal (< 0.10).

Metric	$\mathrm{share}_{\text{variant}}$	$\mathrm{share}_{\text{seed}}$	$\mathrm{share}_{\text{snap}}$
core frac_alive	0.78	0.00	0.22
core cossim	0.40	0.00	0.60
core explained_variance	0.39	0.00	0.61
core MSE	0.39	0.00	0.61
sae-probes k=1	0.27	0.08	0.65
RAVEL disentangle	0.25	0.26	0.49
sae-probes k=5	0.25	0.09	0.66
TPP top-2	0.17	0.08	0.75
sae-probes k=16	0.17	0.21	0.62
autointerp	0.16	0.11	0.72
core CE loss	0.16	0.12	0.72
sae-probes k=2	0.15	0.34	0.50
SCR top-20	0.15	0.68	0.17
sparse probing top-10	0.15	0.45	0.40
sae-probes k=10	0.14	0.18	0.68
SCR top-5	0.13	0.55	0.32
core KL divergence	0.12	0.04	0.84
SCR top-2	0.11	0.70	0.19
TPP top-10	0.10	0.13	0.77
RAVEL isolation	0.10	0.20	0.70
sparse probing top-5	0.09	0.64	0.26
TPP top-5	0.09	0.08	0.82
SCR top-10	0.09	0.45	0.46
sparse probing top-2	0.08	0.57	0.35
sparse probing overall	0.07	0.14	0.79
TPP top-500	0.05	0.01	0.94
SCR top-500	0.05	0.40	0.54
TPP top-50	0.05	0.17	0.78
TPP top-20	0.05	0.15	0.80
sparse probing top-1	0.05	0.71	0.24
SCR top-100	0.04	0.40	0.55
SCR top-50	0.03	0.43	0.53
TPP top-100	0.03	0.08	0.89
RAVEL cause	0.02	0.17	0.81

Table 17. Number of distinct variants crowned across 3 training seeds for the 34 audited SAEBench metrics on the Matryoshka panel, with the canonical headline-suite metrics named individually. The "three distinct" column lists every metric in that bucket; the "two distinct" and "one consistent" columns name only canonical or otherwise headline metrics.

# distinct winners across 3 seeds	# metrics	metrics (selection)
1 (consistent across all seeds)	9	core MSE, cossim, EV, frac_alive, KL, CE; RAVEL isolation; sae-probes $k = 5$; TPP top-500
2 (different winner from one seed)	16	autointerp; RAVEL disentangle; SCR top-$\{2, 20, 100, 500\}$; TPP top-$\{2, 5, 50\}$; sae-probes $k \in \{1, 2, 10\}$; sparse probing top-$\{2, 5, 10\}$
3 (different winner each seed)	9	RAVEL cause; sae-probes $k = 16$; SCR top-$\{5, 10, 50\}$; sparse probing top-1; TPP top-$\{10, 20, 100\}$

Table 18. Mean Spearman ρ between the variant rankings produced by each pair of training seeds, averaged across the three seed pairs, for each of the 34 audited SAEBench metrics on the Matryoshka panel. With 4 variants the Spearman lattice has 11 points per pair and 27 distinct mean values, so small differences in mean ρ are quantised. We read the upper end (stable cross-seed ranking) as informative and the lower end (no detectable consistent ranking) as consistent with low $\mathrm{share}_{\text{variant}}$, not as active anti-correlation.

Metric	mean ρ across seeds
core MSE	$+1.00$
core cossim	$+1.00$
core explained_variance	$+1.00$
core frac_alive	$+1.00$
core KL divergence	$+0.87$
TPP top-500	$+0.87$
core CE loss	$+0.73$
TPP top-2	$+0.73$
autointerp	$+0.67$
RAVEL isolation	$+0.67$
sae-probes k=1	$+0.67$
sae-probes k=5	$+0.67$
sparse probing overall	$+0.47$
sae-probes k=16	$+0.40$
RAVEL disentangle	$+0.20$
TPP top-5	$+0.20$
sparse probing top-10	$+0.13$
sparse probing top-5	$+0.07$
sae-probes k=10	$+0.07$
TPP top-100	-0.07
sae-probes k=2	-0.07
TPP top-50	-0.07
sparse probing top-2	-0.13
SCR top-100	-0.13
SCR top-2	-0.20
SCR top-5	-0.20
RAVEL cause	-0.20
SCR top-10	-0.20
TPP top-10	-0.20
sparse probing top-1	-0.33
SCR top-500	-0.33
TPP top-20	-0.33
SCR top-20	-0.33
SCR top-50	-0.40

Table 19. Spearman ρ of each benchmark with GT-MCC and GT-F1 on the v1 panel, multi-seed mean across 3 task-generation seeds at canonical hyperparameters.

Benchmark	ρ vs GT-MCC (full / within-trained)	ρ vs GT-F1 (full / within-trained)
Sparse probing single in-sae (top-16)	$+0.58/+0.49$	$+0.02/-0.22$
Sparse probing boolean in-sae (top-16)	$+0.90/+0.87$	$+0.60/+0.51$
Sparse probing single out-of-sae (top-16)	$+0.23/+0.19$	$-0.17/-0.37$
Sparse probing boolean out-of-sae (top-16)	$-0.04/-0.22$	$-0.21/-0.53$
TPP all in-sae (top-10)	$+0.18/-0.03$	$-0.22/-0.69$
TPP all out-of-sae (top-10)	$+0.38/+0.42$	$+0.11/-0.03$
SCR canonical (top-10)	$+0.64/+0.65$	$+0.51/+0.41$

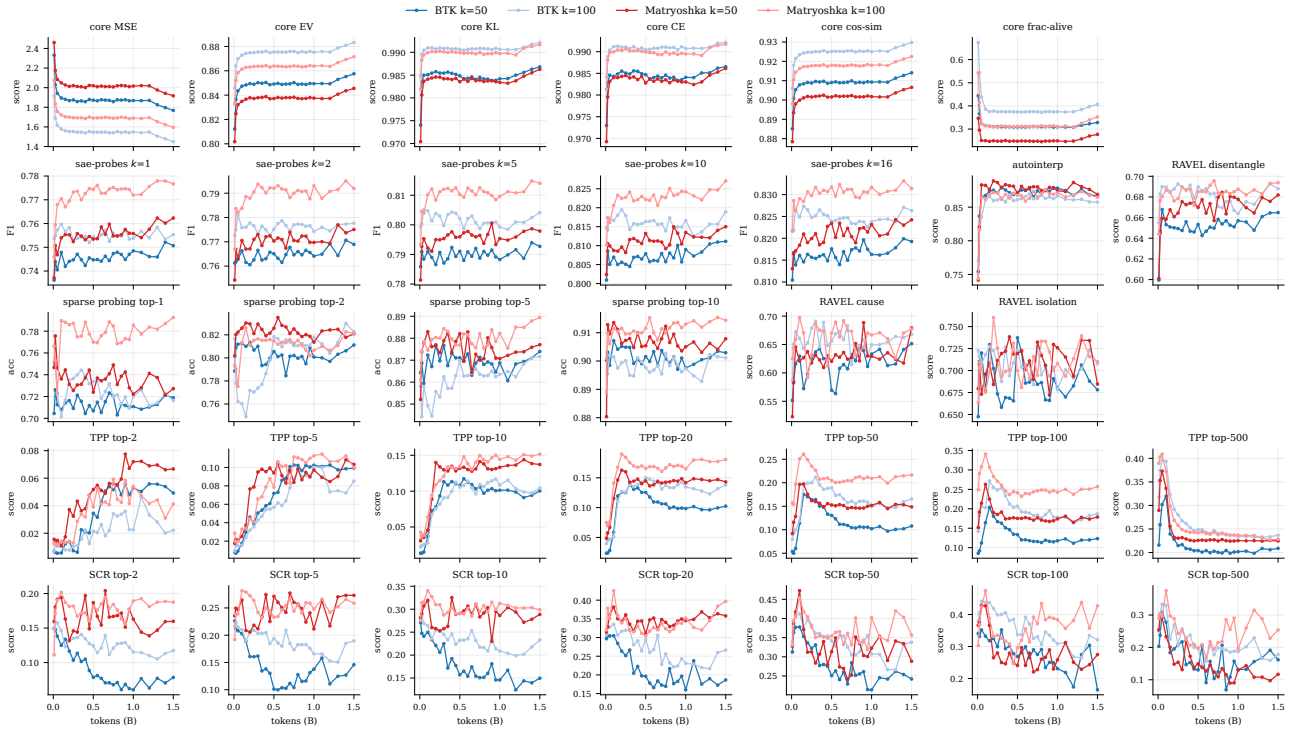

Figure 13. Trajectories of every SAEBench metric across the four-SAE architecture training panel (BTK $k \in \{50, 100\}$, Matryoshka $k \in \{50, 100\}$, 1.5B tokens). Rows from top to bottom: core reconstruction; encoder-only quality (sae-probes, autointerp, RAVEL disentangle); default sparse probing and remaining RAVEL metrics; TPP top-$\{2, 5, 10, 20, 50, 100, 500\}$; SCR top-$\{2, 5, 10, 20, 50, 100, 500\}$.

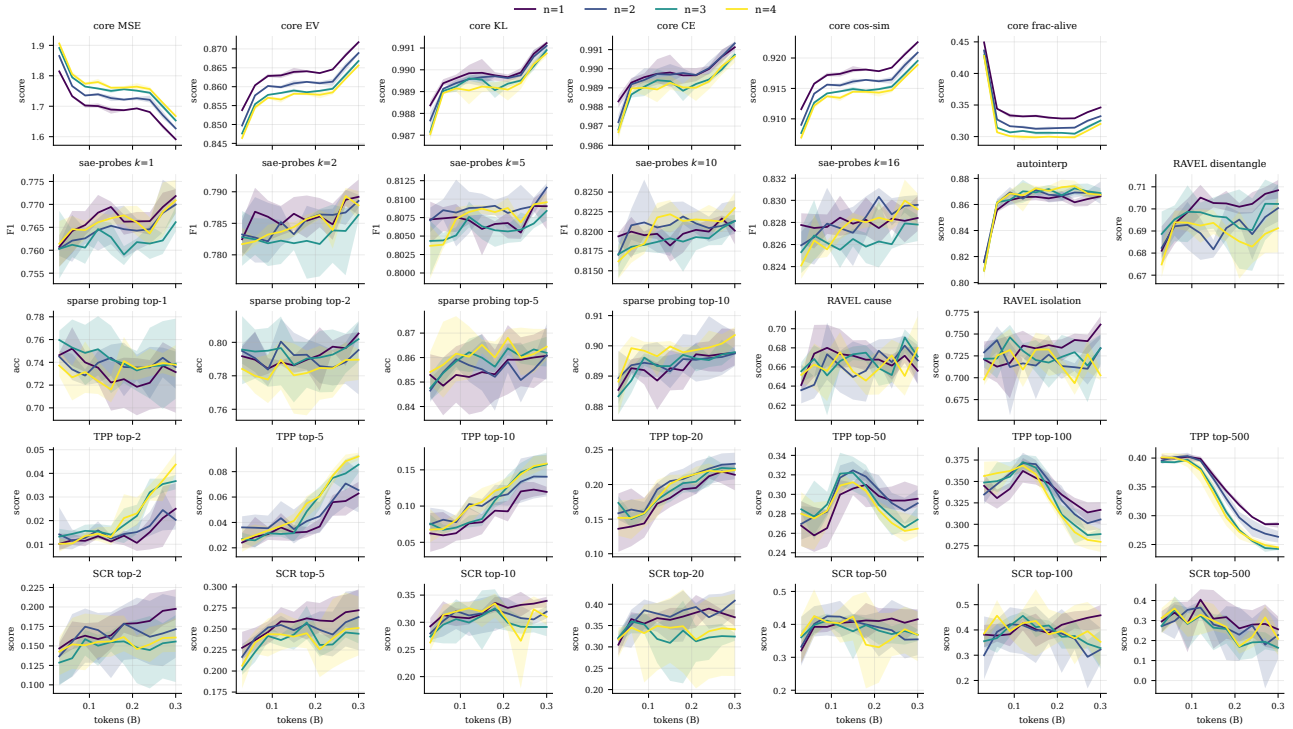

Figure 14. Trajectories of every SAEBench metric across the four-SAE sampled-Matryoshka training panel ($n \in \{1, 2, 3, 4\}$ inner widths sampled per training step, 300M tokens, 3 training seeds per variant). Solid lines are seed means; shaded bands are min/max envelopes across the 3 seeds. Layout identical to Figure 13.