Size Doesn't Matter: Cosine-Scored Sparse Autoencoders

Silen Naihin [1] **Lev Stambler** [2]

Abstract

Sparse autoencoders (SAEs) detect features via inner product, so a feature's activation scales with both its directional alignment and the input's norm. Features that fire on token norm therefore claim dictionary slots regardless of content alignment. This matters because sublayer normalization has already discarded the magnitude the score measures, so the encoder detects a quantity the model does not read. We replace the score with a learned blend of cosine similarity and input magnitude, letting the optimizer choose how much norm to use; a per-feature extension lets each feature decide independently. In both regimes, training is free to recover inner product but never does, with no feature ever choosing more than half-magnitude dependence. At matched reconstruction, the cosine encoder learns features that align with human-recognizable concepts far more often than standard, filling dictionary slots that inner product wastes on norm detectors. Loss reweighting that equalizes gradients barely closes the gap, confirming forward-pass score geometry as the lever. The advantage is not universal across tasks or depths, but we believe cosine scoring should be the default for dictionary learning on normalized representations.

1. Introduction

Sparse autoencoders are a standard dictionary-learning tool for mechanistic interpretability (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024; Bussmann et al., 2024; Karvonen et al., 2025). A trained SAE is read as a feature dictionary: if feature i fires on activation x, then x is taken to contain the concept represented by the corresponding decoder direction. The interpretation depends on how the encoder detects those directions. The standard rule is an inner product $\langle w_i, x\rangle$, so a feature fires in proportion to both its directional alignment with x and the magnitude $\|x\|$.

This is the wrong scoring geometry for transformers with pre-sublayer normalization. The inner-product score mixes content alignment with token norm, but normalization strips that norm before the model reads the activation; the encoder detects a quantity the downstream computation ignores. This mismatch compounds under TopK selection (Bussmann et al., 2024): features compete for a limited budget by score, so norm-inflated features crowd out better-aligned ones regardless of content. BatchTopK's batch-wide budget sharpens this, with high-norm tokens claiming slots across the batch, but the crowding is a property of the score, not the selector, and persists when each token is scored independently. Over training, this selection pressure starves content-encoding features: they rarely win TopK slots, receive no learning signal, and never develop. The result is a dictionary dominated by features that fire on magnitude rather than meaning. This is not a minor calibration issue. If dictionary elements fire on norm rather than content, they cannot reliably be used for sparse probing, feature-based steering, or circuit analysis; the practitioner interprets them as model concepts, but the features encode a quantity the model discards. A score aligned with what the model actually reads should depend on direction, not magnitude.

We replace the inner-product score with a cosine score scaled by a learned exponent a on the input norm:

$$s_i(x) \;=\; \|x\|^a \cdot \cos(x, w_i) \;+\; b_{\mathrm{enc},i},$$

which interpolates between cosine ($a = 0$) and inner product ($a = 1$); we use unit-normalized encoder rows so a=0 recovers cosine cleanly (§3).[1] We study two parameter regimes: a single global a shared across features, and a per-feature extension $a_i = a_{\mathrm{base}} + \delta_i$ that lets each feature learn its own norm dependence. The key property is that the optimizer is free to recover inner product (a=1) if magnitude is genuinely useful, but is not forced to use it by default. This lets training discover the right geometry rather than assuming it. In both regimes, training consistently drives a toward zero; no feature ever approaches the inner-product regime, confirming that direction, not magnitude, is the useful signal.

Through experiments on Qwen3-8B and Gemma-2-2B, we

[1]Experiential Labs [2]Tear Labs. Correspondence to: Silen Naihin <silen@experientiallabs.ai>.

Spotlight paper at the ICML 2026 Workshop on Mechanistic Interpretability. Copyright 2026 by the author(s).

[1]Up to a learned global temperature; see §3.

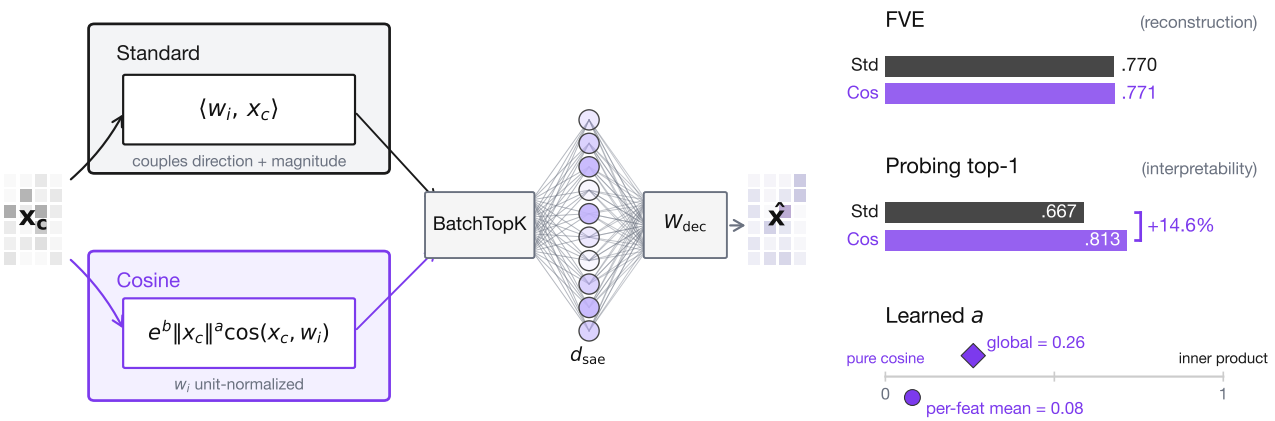

Figure 1. **The cosine encoder: architecture and headline results. Left:** Standard SAE encoder computes $\langle w_i, x_c \rangle$, coupling alignment with norm. Cosine encoder unit-normalizes w_i and replaces the score with $e^b \|x_c\|^a \cos(x_c, w_i) + b_{\text{enc},i}$; a interpolates between pure cosine (a=0) and inner product (a=1). **Right:** Matched reconstruction (FVE ≈ 0.77) with $+14.6\%$ sparse-probing top-1 (mean over three SAE-training seeds). Training drives $a \approx 0.26$, far from the inner-product limit. Qwen3-8B L18, 500M tokens, d_{sae}=65,536.

demonstrate that cosine-scored SAEs match standard BatchTopK on reconstruction fidelity, model-behavior preservation, and per-feature interpretability, while producing dictionaries that align far more often with human-recognizable concepts. A matched-feature decomposition reveals that the majority of the probing gap comes not from improving shared features, but from features the standard encoder never learns at all; its dictionary slots are instead occupied by norm detectors. We isolate the cause to forward-pass TopK selection under norm inflation: loss reweighting that equalizes gradients across norm levels barely closes the gap, confirming that the score geometry itself, not the training signal, is the lever. The advantage replicates across layers and models, though it is not universal; deep LayerNorm layers and sentiment are exceptions where magnitude carries task-relevant signal.

Contributions.

- **Architecture.** We introduce cosine-scored sparse autoencoders, replacing the inner-product encoder score with a learned-exponent cosine score that interpolates between pure cosine (a=0) and inner product (a=1). We study global and per-feature parameterizations and identify the design choices that make each stable (§3).
- **Result.** At matched reconstruction, KL, and per-feature interpretability, the cosine encoder lifts sparse-probing top-1 by $+14.6\%$ on Qwen3-8B. A matched-feature decomposition shows that the majority of the gap comes from features the standard encoder never learns at all, rather than from improving shared features. The result replicates across layers and on Gemma-2-2B (§4.1, §4.3).
- **Mechanism.** The standard encoder's unmatched features fire overwhelmingly on high-norm tokens; they encode magnitude, not content. Forward-pass TopK selection under norm inflation is the primary cause; loss reweighting that equalizes gradients across norm levels barely closes the gap, confirming that the score geometry itself is the lever. The optimizer independently confirms this: training drives a far from the inner-product regime, with no feature ever choosing more than half-magnitude dependence (§4.3, §4.3).

For practitioners, this is a drop-in encoder change that produces more interpretable dictionaries at no reconstruction cost. More broadly, it suggests that scoring geometry is an underexplored design axis for dictionary learning: if inner product is the wrong inductive bias for normalized representations, standard SAE dictionaries trained on modern transformers may systematically undercount content features. We believe cosine scoring should be the default for dictionary learning on normalized representations, and that the failure mode we identify likely affects any inner-product SAE trained on a post-RMSNorm site.

2. Background

A sparse autoencoder (SAE) takes an activation $x \in \mathbb{R}^d$, projects it through a single-layer encoder–decoder pair to a sparse code $z \in \mathbb{R}^{d_{\text{sae}}}$, and reads the decoder rows $\{W_{\text{dec},i}\}$ as a feature dictionary:

$$z = \sigma\big((x - b_{\text{dec}})W_{\text{enc}}^{\top} + b_{\text{enc}}\big) \;\; \hat{x} = zW_{\text{dec}} + b_{\text{dec}}.$$

with reconstruction loss: $\mathcal{L} = \|x - \hat{x}\|^2$. Different SAE variants make σ sparse in different ways, including TopK (Gao et al., 2024), BatchTopK (Bussmann et al., 2024), JumpReLU (Rajamanoharan et al., 2024b), gated activations (Rajamanoharan et al., 2024a), and AbsTopK (Zhu et al., 2025). SAEs tend to assume the linear representa-

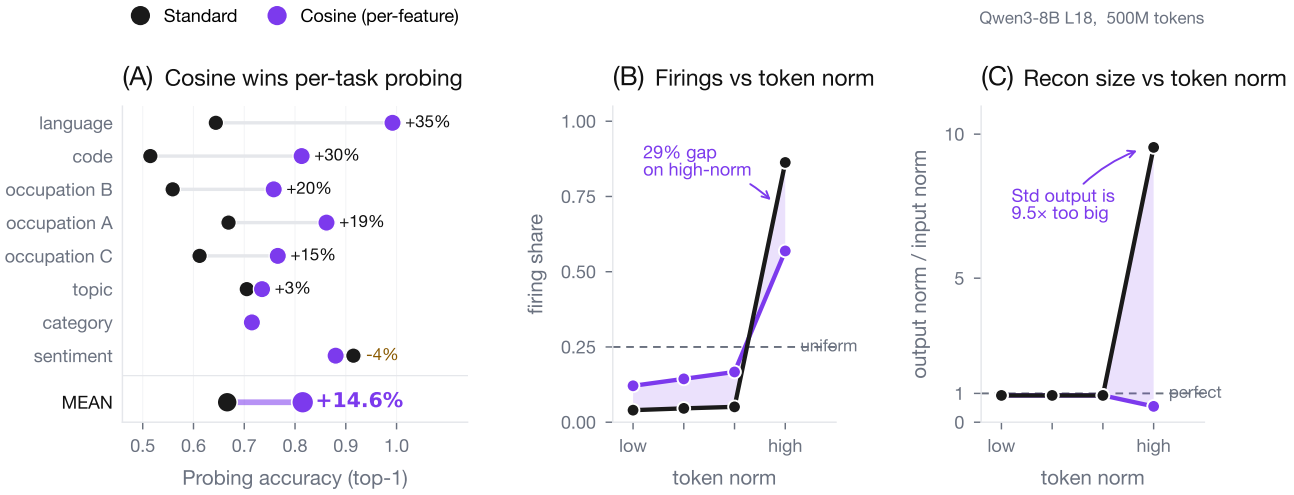

Figure 2. **Cosine-scored SAEs win on probing because standard features fire on token norm. (A)** Result: sparse-probing top-1 across eight tasks (Qwen3-8B L18, 500M tokens, d_{sae}=65,536, matched FVE ≈ 0.77). Per-feature cosine wins on 7/8 tasks; sentiment is the only exception. **(B)** Cause: standard's *unmatched* features (those with no nearest-neighbor counterpart in cosine's dictionary) fire $22\times$ more on the highest-norm token quartile than on the lowest, versus $4.7\times$ for cosine; they encode magnitude, not content. **(C)** Confirmation: on the same high-norm tokens, the standard SAE reconstructs at $9.5\times$ the input norm, while cosine stays close to the input scale ($0.55\times$). Removing the $\|x\|$ factor from the score recovers content-encoding features. Provenance: Appendix D.2.

tion hypothesis: concepts correspond to linear directions in activation space (Park et al., 2024; 2025; Elhage et al., 2022). Recent variants modify dictionary geometry (Korznikov et al., 2025; Bussmann et al., 2025) or loss shape (Nasiri-Sarvi et al., 2026) but leave the encoder score unchanged. A feature's pre-activation is still an inner product $\langle w_i, x\rangle = \|w_i\|\,\|x\|\,\cos(w_i, x)$, so larger-norm tokens raise the score even when directional alignment is unchanged. Following standard practice (Bricken et al., 2023; Gao et al., 2024; Karvonen et al., 2025), decoder rows are constrained to unit norm during training: re-normalized after each optimizer step, with the component of the decoder gradient parallel to each row projected away to account for the Adam-normalization interaction (Gao et al., 2024). The dictionary therefore represents directions, and any per-feature scale is absorbed into the encoder. The encoder reads the input centered on the decoder bias ($x \mapsto x - b_{\text{dec}}$, as in the equation above), tying its zero point to the reconstruction origin (Bricken et al., 2023).

We hook the SAE on the residual stream: the activation x that flows untouched through every transformer block, with each sublayer adding to it. Most modern transformers place an RMSNorm (Zhang & Sennrich, 2019) on the path *into* every sublayer:

$$\text{RMSNorm}(x) \;=\; \sqrt{d}\,\frac{x}{\|x\|} \odot g, \qquad g \in \mathbb{R}^d.$$

Each sublayer therefore reads $x/\|x\|$ up to the per-coordinate gain g, not x itself. The SAE, hooked one step earlier on the residual stream, still sees x in full. Two activations with the same direction and different norms look nearly identical to the model but very different to an inner-product encoder, which scores them in the ratio of their norms. Residual-stream norms are heavy-tailed in practice, driven by rogue dimensions (Timkey & van Schijndel, 2021) and outlier features (Dettmers et al., 2022), so this bias is not hypothetical.

Directly swapping activation directions between random token pairs (no SAE involvement) produces $219\times$ more downstream KL-divergence than swapping their norms at the same layer (Appendix C.1). The model's computation is direction-dominated; the SAE's encoder should be too.

We use BatchTopK throughout (Bussmann et al., 2024), the SAEBench default (Karvonen et al., 2025). BatchTopK imposes a single batch-wide activation budget rather than forcing exactly k features per token, letting the model allocate more features to complex tokens and fewer to simple ones. Given pre-activations $u = \sigma(xW_{\text{enc}}^{\top} + b_{\text{enc}}) \in \mathbb{R}^{N \times d_{\text{sae}}}$ over a batch of N tokens, $z = \text{BatchTopK}(u)$ keeps only the kN largest entries and zeros the rest. This flexibility also means high-norm tokens can claim disproportionately many slots: their inflated pre-activations dominate the batch-wide ranking regardless of directional alignment. Batch-wide competition is one route by which magnitude corrupts selection, but it is not the only one; the inner-product score conflates direction with magnitude through the input norm and the encoder-row norm alike, so the pathology survives even when each token is scored independently. Our cosine score modifies only the encoder pre-activation, not the sparsity mechanism, so it composes with any selector, and we confirm the advantage holds under per-token TopK and Ab-

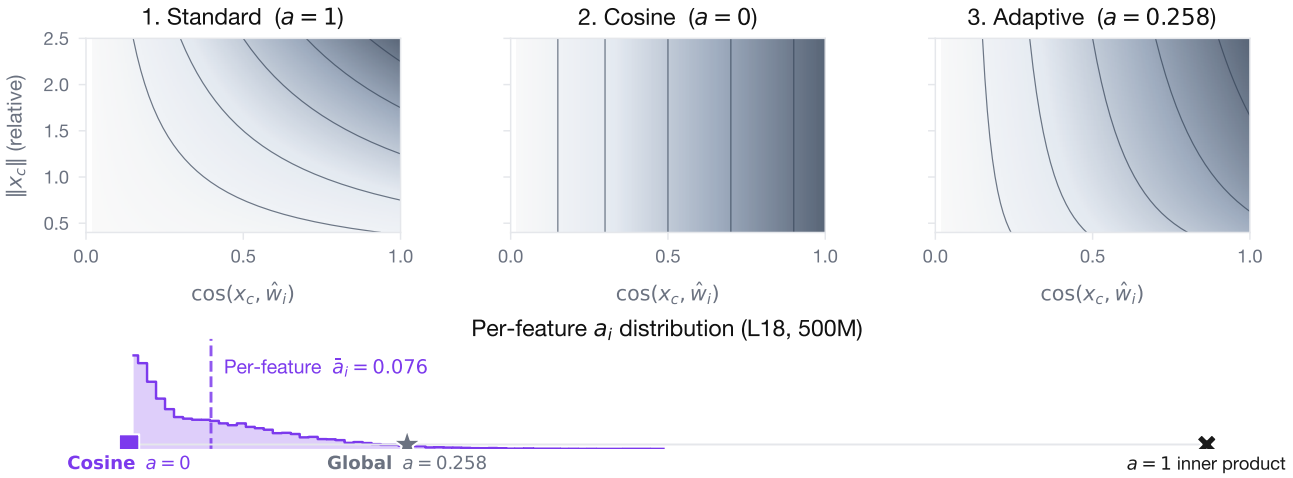

Figure 3. **Score-surface geometry.** Each panel plots the encoder pre-activation $\|x_c\|^a \cos(x_c, w_i)$ over alignment (x-axis) and input norm (y-axis). Black curves join equally-scored pairs. *Left:* a=1 (inner product); hyperbolic curves, high-norm tokens outscore better-aligned low-norm ones. *Center:* a=0 (cosine); vertical curves, norm ignored. *Right:* global learned a≈0.26; mild tilt, close to cosine. Bottom: per-feature a_i distribution at the headline setting.

sTopK as well as BatchTopK (our default; Appendix B.4). Penalty-trained selectors (JumpReLU, gated) are left to future work.

More broadly, modern architectures are adding normalization (QK-norm (Henry et al., 2020; Dehghani et al., 2023), nGPT (Loshchilov et al., 2024)), strengthening the case for direction-aware SAE encoders.

Evaluation metrics. We follow SAEBench (Karvonen et al., 2025) throughout. Reconstruction is reported as fraction of variance explained, $\mathrm{FVE} = 1 - \mathbb{E}\|x - \hat{x}\|^2/\mathrm{Var}(x)$, computed per token and aggregated. Sparse probing fits a linear probe restricted to the k SAE features that best predict a labeled concept and reports the resulting accuracy (top-k) over eight classification datasets; top-1 isolates the single best feature and is the most direct test of whether one dictionary direction matches a human-recognizable concept. Auto-interp asks an LLM to describe a feature from its top-activating contexts and a second LLM to score whether the description predicts the firing pattern; we report the fraction of features judged interpretable (Appendix F).

3. The Cosine Encoder

The fix is straightforward in principle: remove $\|x\|$ from the score. The design question is how much norm information to retain, since the decoder must still reconstruct x at full scale. We define a one-parameter family that lets the optimizer answer this question.

Notation. Let $x_c = x - b_{\mathrm{dec}}$ be the input centered on the decoder bias. We keep encoder rows w_i unit-normalized, recomputing $w_i \leftarrow w_i/\|w_i\|$ on every forward pass with gradients flowing through the normalization.[2] Decoder rows are also unit-normalized, per the held-fixed community recipe. Encoder unit-normalization is what makes the score true cosine. Dropping it (keeping input-side normalization but using raw w_i) yields $\|w_i\| \cdot \cos(x_c, w_i)$, a half-cosine that reintroduces $\|w_i\|$ as a magnitude factor. $\|w_i\|$ then drifts unconstrained (no loss term penalizes it) and the dictionary collapses to 93% dead features at L27/50M.[3]

Score, pipeline, and initialization. We define the pre-activation as

$$s_i(x) \;=\; \underbrace{\exp\big(a \log \|x_c\| + b\big)}_{e^b\,\|x_c\|^a} \cdot \cos(x_c, w_i) \;+\; b_{\mathrm{enc},i},$$

The log-exp parameterization makes the scale explicit: a linear fit $\log(\mathrm{scale}) = a \log \|x_c\| + b$ exponentiates to $\mathrm{scale} = e^b \|x_c\|^a$, where a is the norm-dependence exponent and e^b is the scale at $\|x_c\|$=1 rather than a separately introduced temperature. This form gives well-conditioned gradients in a regardless of $\|x\|$ (Appendix B). It also interpolates cleanly between two familiar cases. When a=0, $\|x_c\|^a$=1 and $s_i = e^b \cos(x_c, w_i) + b_{\mathrm{enc},i}$ is pure cosine at scale e^b. When a=1, $s_i = e^b \langle x_c, w_i \rangle + b_{\mathrm{enc},i}$ is inner product up to a global scale, since w_i is unit-norm. After applying a ReLU, this score is passed to BatchTopK (Bussmann et al., 2024). The decoder remains unchanged: $\hat{x} = zW_{\mathrm{dec}} + b_{\mathrm{dec}}$, with $\mathcal{L} = \|x - \hat{x}\|^2$. We initialize $a = 0$ and $b = \log \sqrt{d_{\mathrm{model}}}$.

[2] So cosine is just an inner product on the sphere: with $\|w_i\| = 1$, $\cos(x_c, w_i) = \langle x_c/\|x_c\|, w_i \rangle$, and at a=1 the score below reduces to $\langle x_c, w_i \rangle$ exactly.

[3] The full design-space matrix (these three settings plus per-axis ablations on encoder/decoder normalization and post-decode norm restoration) is in Appendix B.2.

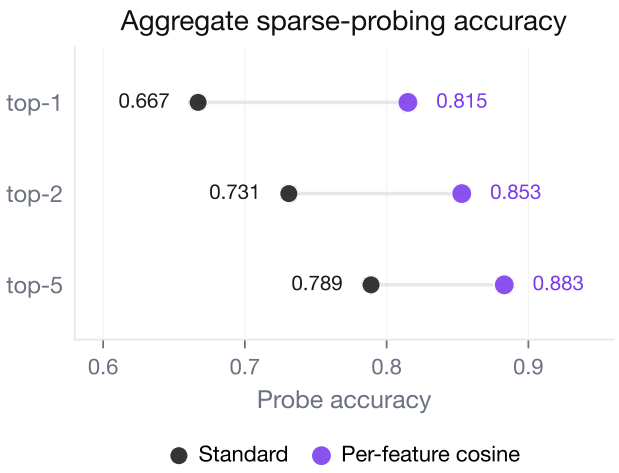

Figure 4. **Aggregate sparse-probing accuracy.** Top-1, top-2, and top-5 probe accuracy across all eight SAEBench datasets. FVE-matched at ≈ 0.77. Black: standard SAE; violet: per-feature cosine encoder. The gap narrows at higher k but remains large ($+9.4\%$ at top-5). Per-dataset breakdown: Fig. 2 Panel A.

The body tracks the two variants used in the headline 500M comparison. A third corner of the family, pinned cosine (a=0, where a is frozen and only b trains), is deferred to App. B.3, alongside the full design-space ablation in App. B.2.

Global learned a. A single scalar $a \in \mathbb{R}$ shared across features, freed at initialization. Pinning a=0 (pure cosine) discards all norm information from the encoder; since the decoder reconstructs x at full scale, the system must recover magnitude from the sparse code alone, and in practice this collapses (Appendix B.2). Freeing a lets the optimizer retain exactly as much magnitude as reconstruction requires.

Per-feature a_i. Different features may want different norm dependence; a position-encoding feature plausibly needs more magnitude than a semantic one. The direct replacement $a \to a_i$, $b \to b_i$ gives $s_i(x) = e^{b_i}\|x_c\|^{a_i}\cos(x_c, w_i) + b_{\mathrm{enc},i}$, adding $\leq 0.1\%$ parameters. Fully free a_i values are unstable at deep layers (cascading to extreme values at L27; Appendix B.2). We instead parameterize $a_i = a_{\mathrm{base}} + \delta_i$ as a shared base plus per-feature offset (both initialized to 0); b_i remains fully free. The shared base anchors the distribution while per-feature deltas allow heterogeneity.

Training recipe. All experiments use the community SAEBench recipe (Karvonen et al., 2025): BatchTopK with $k = 80$, Adam at learning rate $5{\cdot}10^{-5}$, the AuxK auxiliary loss of Gao et al. (2024) to revive dead features, unit-norm decoder rows, and geometric-median initialization for b_{dec}. The encoder is initialized as the transpose of the decoder. The recipe is held fixed across the standard baseline and all cosine variants; only the encoder score changes. Appendix J gives the full hyperparameters and recipe-lineage detail.

4. Experiments

Unless stated otherwise, results use Qwen3-8B at layer 18, 500M FineWeb tokens, $d_{\mathrm{sae}} = 65{,}536$, and the recipe in §3. Mechanism sweeps use the same model at smaller budgets (2–50M tokens). Appendix E gives cross-layer and cross-model checks on Qwen L9/L27, Gemma-2-2B, and three LayerNorm models; Appendix B.2 gives the architecture ablation.

4.1. Reconstruction and sparse probing

At the headline scale, reconstruction is tied between the standard BatchTopK SAE and both cosine variants. Aggregate FVE differs by $\leq 0.4\%$, and the SAEBench substitution metrics (KL-score and CE-score, both bounded $[0, 1]$ with 1 meaning the SAE preserves the model's logits exactly) agree to within 0.001 (0.984–0.985 KL-score, 0.991–0.993 CE-score; Appendix D). The difference is in feature use.

On the eight SAEBench single-feature top-k probing datasets, averaged over three SAE-training seeds at the full 500M scale, the cosine encoder improves top-1 by $+14.1\%$ (global regime) and $+14.6\%$ (per-feature regime): top-1 is 0.667 ± 0.003 (Standard), 0.808 ± 0.006 (global), and 0.813 ± 0.013 (per-feature) (Appendix K). The gap dwarfs the seed-to-seed SD and the $\pm 0.63\%$ 5-seed probe-training noise; we omit these error terms below and report point estimates. Single-feature probing directly measures whether individual dictionary elements correspond to human-recognizable concepts (Karvonen et al., 2025), the core desideratum of interpretability tools. The gap is not a capacity artifact: giving the standard encoder $3\times$ more dictionary slots at matched L0 makes dead rates *worse* (89.4% vs. 77.4%), not better (Figure 25). The advantage is also not specific to BatchTopK's batch-wide competition: at matched L_0 the cosine encoder improves top-1 over inner-product under per-token TopK ($+5$ to $+7\%$) and AbsTopK ($+12$ to $+14\%$) as well, so it reflects inner-product scoring in general rather than one selector (Appendix B.4, Table 6). It is likewise robust to the sparsity budget: sweeping $k \in \{10, 160\}$, the top-1 gain holds at $+8$ to $+11\%$ with no collapse at low k, and per-feature interpretability stays matched throughout (Appendix E.3, Table 14). Per-feature interpretability remains matched: at the 500M headline, an LLM-judged describe-then-predict evaluation on 1000 stratified features per architecture scores 19.2% (per-feature cosine), 21.3% (global), and 20.1% (standard), a 2.1-point band (Appendix F.2; per-architecture feature examples in Appendix F.3).

The gain is uneven across tasks (Fig. 4). Language and code detection account for the largest gaps, but the per-feature advantage remains $+9.1\%$ after removing both. Sentiment is the one reversal: standard is higher by $+3.5\%$, consistent with magnitude carrying task-relevant signal there. We

Table 1. **Matched reconstruction.** Qwen3-8B L18, 500M tokens. Only the score changes. FVE, probing top-1, and learned a are means over three SAE-training seeds (Appendix K); Q4 FVE is the representative seed, which diagnoses the standard baseline's per-quartile failure (§4.2) and is reused for the per-checkpoint analyses in §4.3. Provenance: Appendix D.2.

	Standard	Global a	Per-feature
FVE	0.770	0.769	**0.771**
Probing top-1	$0.667_{\pm 0.003}$	$0.808_{\pm 0.006}$	$\mathbf{0.813}_{\pm 0.013}$
Q4 FVE	-184	$\mathbf{+0.33}$	$\mathbf{+0.25}$
Dead %	0.0	0.0	0.0
Learned a	—	0.258	mean 0.076

return to this case in §5.

The learned norm dependence remains far from the inner-product limit (Fig. 3, Table 1). The global a converges to ≈ 0.26; the per-feature a_i have mean ≈ 0.08, with no $a_i > 0.5$. Both regimes initialize at $a = 0$, so these values are learned rather than imposed. Appendix K reports initialization sweeps and a postnorm-MSE alternative where a becomes negative.

4.2. High-norms break inner-product reconstruction

Aggregate FVE matches, but the standard SAE fails on the high-norm tail predicted by §2. We bin tokens by $\|x_c\|$ into quartiles and write *Q4* for the highest quartile (the 25% of tokens with the largest $\|x_c\|$). Standard and cosine match on Q1–Q3, but the standard encoder over-activates on Q4: for the standard SAE we trained under the community recipe of §3 (the same SAE used in Table 1), Q4 reconstructions have $9.5\times$ the input norm, and Q4 FVE is -184.[4] Both cosine regimes keep Q4 FVE positive (Fig. 6). An independently trained reference SAE shows the same pathology, with Q4 FVE $= -136$ (Karvonen et al., 2025).

This is the failure mode predicted in §2: the inner-product score writes $\|x\|$ into the sparse code, and the decoder turns that scale information into inflated reconstructions, even though downstream RMSNorm has already discarded it.

4.3. Comparing the learned dictionaries

The standard SAE does not merely encode content worse; it fails to learn content features at all, spending most of its capacity on features that detect token magnitude instead. The $+14.6\%$ sparse-probing gap from §4.1 persists at matched aggregate FVE, so the two architectures must learn qualitatively different dictionaries. This section identifies which features each one learns.

[4]The large negative per-quartile number does not contradict aggregate FVE parity: per-quartile FVE normalizes by *within*-quartile variance ($1 - \mathrm{MSE}_q/\mathrm{Var}_q(x)$), and within-Q4 variance is small relative to the global variance that aggregate FVE normalizes against, so Q4's large absolute MSE contributes only modestly to the aggregate numerator.

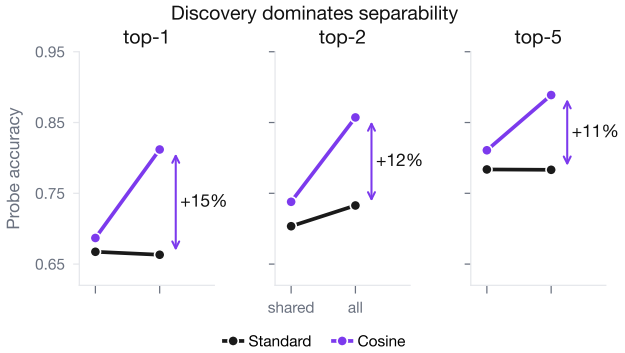

Figure 5. **Discovery dominates separability.** Sparse-probing accuracy when each SAE uses only features shared with the other dictionary ("shared features") versus its full dictionary ("all features"). Standard's flat slope shows its unique features add no probe signal; cosine's steep rise shows its unique features encode interpretable concepts. The gap on the right is the total probing advantage, driven almost entirely by feature discovery rather than cleaner encoding of shared directions.

Dead features do not explain the gap. With the auxiliary loss enabled, both architectures keep $\geq 99\%$ of features alive and match on reconstruction and on LLM-judged per-feature interpretability (within a 2.1-point band at the 500M headline; Appendix F.2). The dead-feature margin closes, but the $+14.6\%$ sparse-probing gap remains.

We pair features across the standard and cosine dictionaries when their per-token activation traces have Pearson correlation ≥ 0.7 *and* their decoder rows have cosine similarity > 0.7: i.e., they activate on the same tokens *and* point in the same direction in residual space. This matches 8,661 features. The remaining 55,789 standard-only features behave as *norm-conditioned features*: they fire on $\|x\|$ rather than on a content direction.

- *Selective.* 86% of their nonzero activations land in Q4 (vs. the 25% a content-blind detector would receive), so they fire on high-norm tokens.
- *Strong.* When they do fire on a Q4 token, their mean activation is $48\times$ the per-token mean across the full token population.
- *Crowding.* BatchTopK on a high-norm token therefore spends most of its sparse-code budget on these features, leaving little capacity for direction-encoding ones (Fig. 6). On tokens where cosine-unique features fire, the standard SAE fires 328 features/token vs. cosine's 122 (Appendix G).

Of the standard SAE's 64,450 alive features, only 13.4% direction-encode under this criterion; the rest primarily encode norm. The between-quartile component drives most of the overall cos>inner rate (80–87%). Within individ-

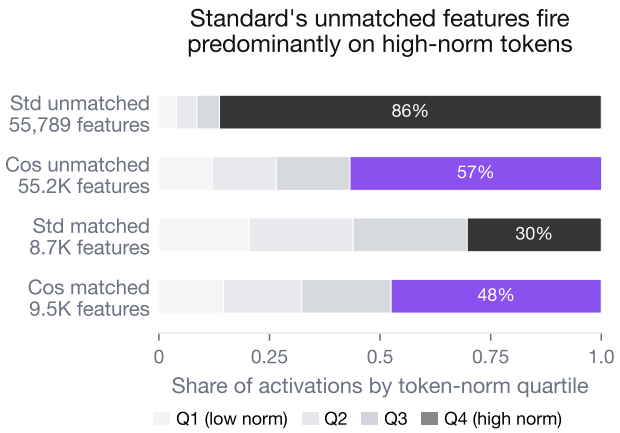

Figure 6. **Norm-conditioned feature allocation.** Standard's unmatched features fire overwhelmingly on Q4; cosine's unmatched features spread across all quartiles. The standard encoder spends most of its capacity on features whose firing is locked to token norm. Per-quartile reconstruction: Appendix G.1.

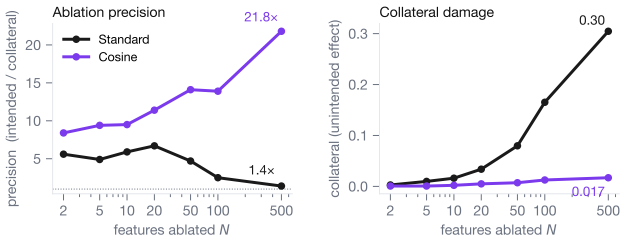

Figure 7. **Cosine features ablate more cleanly.** Ablating the top-N probe-selected features and decomposing the logit change into intended (target concept) and unintended (collateral) effects. **Left:** precision (intended/collateral) rises with N for cosine but collapses toward $1\times$ for standard. **Right:** the cause is collateral, not intended effect; standard's unintended damage grows two orders of magnitude while cosine's stays flat. Qwen3-8B L18, 500M. [exp57b]

ual quartiles the advantage is a more modest 40–70% (Figure 16), consistent with norm variation corrupting TopK selection across the full distribution rather than within each quartile independently. We call these features "norm-conditioned" descriptively: they may encode $\|x\|$-correlated information our probing tasks do not measure (auto-interp parity is consistent with this; Appendix F), but they do not encode the eight concept categories sparse probing targets.

Restricting both dictionaries to the 8,661 matched features drops the top-1 gap (here the representative seed's $+14.9\%$) to $+2.0\%$ (Fig. 5). Reading $+2.0\%$ as the separability contribution and the residual $+12.9\%$ as the discovery contribution: $\sim 87\%$ of the gap comes from features the cosine encoder discovers and standard does not, $\sim 13\%$ from cleaner encoding of shared directions. The unmatched standard features add almost nothing to top-5. The unmatched cosine features add $+7.8\%$, confirming that the discovered features encode content, not just norm-correlated firing. Direct behavioral steering produces matched KL divergence for both architectures (ratio 0.94–$1.00\times$); the quality gap is which features exist, not their individual intervention power (Appendix G). Beyond probing, the cosine dictionary also ablates more cleanly (Fig. 7): removing the top-N probe-selected features, its intended/collateral precision ratio rises with N ($8.4\times \rightarrow 21.8\times$) while standard's collapses ($5.6\times \rightarrow 1.4\times$). The gap is in collateral damage, not intended effect: both architectures remove the target concept comparably, but standard's unintended side effects grow two orders of magnitude with N (to 0.30) while cosine's stay flat (to 0.017), because the norm-conditioned features that fill the standard dictionary are entangled with every concept (full decomposition: Appendix G, Table 18).

Ruling out gradient equalization. A natural alternative is that cosine helps by equalizing gradient magnitudes across norm quartiles (inner-product gradients scale with $\|x_c\|$). The asymmetry is real: the median Q4/Q1 gradient ratio is 1.6–$2.0\times$ for standard vs. 0.8–$1.0\times$ for cosine. But reweighting the standard loss to equalize gradients by construction closes only 1.9–6.8% of the probing gap (Appendix G.3); the forward-pass TopK selection geometry, not backward-pass gradient flow, is the lever. A direct score swap confirms this: reading a trained dictionary with the other score (weights fixed) moves probing to match whichever score is used at read time, while reconstruction stays bound to the training score (Appendix G.2, Table 20).

Four qualitatively different lines of evidence converge on the same root cause: behavioral (sparse probing), structural (norm-detector characterization), interventional (gradient falsification), and architectural (learned a) all point to the inner-product encoder's conflation of direction and magnitude in feature selection. This conflation is a property of the score, not of one selector: it persists under per-token TopK and AbsTopK (Appendix B.4), so batch-wide competition amplifies the effect but is not its source.

4.4. Removing the auxiliary loss for dead features

The headline runs in §4.1 keep the auxiliary dead-feature loss (Gao et al., 2024) enabled in both arms, which equalizes alive-feature counts ($\geq 99\%$) and reconstruction (FVE within 0.4%) so that the surviving $+14.6\%$ sparse-probing gap cannot be attributed to a dead-feature artifact. Disabling the auxiliary loss isolates what the score swap does to the dead-feature distribution on its own. All major public SAE recipes we are aware of (Bricken et al., 2023; Gao et al., 2024; Karvonen et al., 2025) include this loss or an equivalent, so this is a research-only stress test rather than a deployment recipe.

Without the auxiliary loss the cosine advantage survives, but only for the per-feature variant. At Qwen3-8B, 50M tokens,

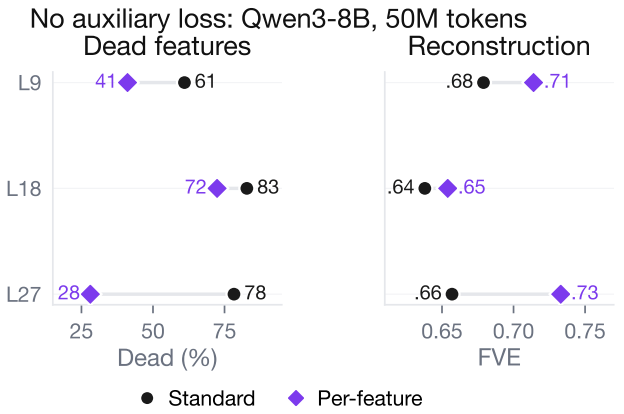

Figure 8. **Without the auxiliary dead-feature loss, per-feature cosine wins at every layer.** Qwen3-8B, 50M tokens, d_{sae}=16,384, $\sqrt{d}$ init. Left: dead-feature %; right: FVE. With the auxiliary loss on (community recipe; headline at d_{sae}=65,536), both architectures hit $\sim 0\%$ dead and FVE parity (§4.1); the gap here is what the auxiliary loss masks. The global-a variant is omitted for readability; see the body.

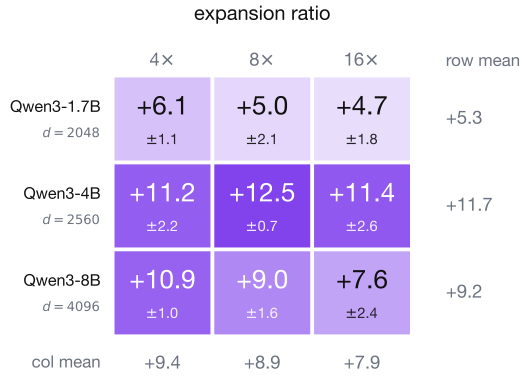

Figure 9. **Model size, not expansion ratio, drives the cosine advantage.** Sparse-probing top-1 gap (cosine − standard, in percentage points; mean ± SD over three SAE-training seeds) across the Qwen3 family. Row means (right) show a $\sim 2\times$ jump from 1.7B to ≥4B; column means (bottom) are flat. All cells use the community SAEBench recipe, 50M tokens, $k = 80$. [exp67]

d_{sae}=16,384 (Fig. 8), per-feature cosine reduces dead features at every layer (L9 61.0 → 41.1%, L18 82.8 → 72.4%, L27 78.3 → 28.1%) and improves FVE at every layer (+3.5/ + 1.6/ + 7.6%). The global-a variant tracks per-feature at L9 and L27 but actually *loses* to Standard at L18 (85.3% dead vs. 82.8%): a single learned scale is a poor compromise at L18's mid-range norm, and the per-feature flexibility is what closes that gap. The pattern reproduces on Gemma-2-2B at 50M tokens: the standard encoder leaves 54–69% of features dead across L7/L13/L19, while cosine stays at 0.0–0.1%, with FVE gains of +3.9 to +6.7% (Table 11; Fig. 21).

Full convergence curves are in Appendix E.

4.5. Cross-model behavior

Our results persist on Gemma-2-2B (a different RMSNorm family, $d_{\text{sae}} = 9{,}216$) but do not persist as strongly on non-RMSNorm models. On Gemma-2-2B, the fixed-SAE top-1 gap is +3.4±0.3%, smaller than on Qwen but same-signed. To disentangle whether model size or dictionary size drives this, we trained a three-seed 3×3 sweep over model size (Qwen3-1.7B/4B/8B) and expansion ratio (4×/8×/16×) at the same recipe (Figure 9). The gap tracks model dimension, not dictionary size: row-mean gaps are +5.3/+11.7/+9.2% across d_{model}, large at every size but not strictly monotonic, while column means across expansion ratio are flat (+9.4/+8.9/+7.9%). Gemma's smaller gap therefore reflects its small d_{model}=2304, not its smaller dictionary. On three LayerNorm models (Pythia-2.8B/6.9B, Falcon-7B), cosine loses its advantage at deep layers (Appendix E.4). A plausible explanation is that LayerNorm's mean subtraction preserves magnitude-correlated structure that RMSNorm erases: the residual stream's per-coordinate mean carries norm information into the sublayer, so inner-product scoring is partially correct at depth in LayerNorm models. We lack a direct intervention test and flag this as an open question.

The cosine encoder trained on 50M tokens exceeds the standard's 500M sparse-probing top-1 at L9 (+17.3%) and L27 (+8.7%), a 10× token-budget saving (Appendix E). Norm-conditioned feature allocation and discovery dominance both reproduce across these settings, with the effect magnitude tracking model dimension rather than dictionary size (Appendix E).

5. Limitations and Future Works

Open mechanism gaps and scope bounds. The mechanism is well isolated but not exhaustively: §4.3 rules out gradient flow as the load-bearing lever, and a direct score swap in both directions (Appendix G.2) shows reconstruction is training-bound while probing follows the read-time score; an initialization sweep over a remains future work. The decoder is also not norm-invariant: under norm-noise training (Uniform$(0.5, 2.0)$), cosine FVE drops 4.5–5.8% vs. 1.8–3.4% for standard, so a deployment norm shift may erode the gain (Appendix H). The gain is also bounded in scope: cosine loses to inner product at deep LayerNorm layers (Pythia-2.8B drops from 100% at L8 to 40% at L24; Appendix E.4), sentiment is the one task reversal (standard +3.5% top-1), and the 500M headline, though reproduced across three SAE-training seeds (Appendix K), is dictionary-level rather than feature-paired (Leask et al., 2025). The advantage holds across three selectors (BatchTopK, per-token TopK, AbsTopK; Appendix B.4); penalty-trained selectors (JumpReLU, gated), scale (> 8B), instruction-tuned/RLHF

checkpoints, circuit-level interventions, and competing geometries (QK-norm, nGPT) are deferred to Appendix H.

Future works. Several recipe choices deserve a closer look. The first is decoder unit-norm: it is load-bearing for reconstruction in some variants (Table 2), but the constraint is not mirrored in the model's normalized readout, and exploring softer regularizers (norm penalties, scheduled annealing, or absorbing $\|w_i\|$ into a per-feature bias) would tell us whether unit-norm is truly essential or just convenient. A second direction is to fold the per-coordinate RMSNorm gain $g \in \mathbb{R}^d$ into the score itself: the current cosine score $\cos(x_c, w_i)$ ignores g, but downstream sublayers read $g \odot x_c$ (§2), so a variant like $\cos(g \odot x_c, w_i)$ would align the encoder strictly with what the model actually consumes. Finally, the auxiliary dead-feature loss may be unnecessary under cosine: even without it, cosine retains $+6.3\%$ FVE and $2.39\times$ more alive features than standard at L27 (Appendix E, Table 10), so a cleaner, auxiliary-free recipe looks within reach.

6. Discussion and Conclusion

For SAEs trained on normalized residual streams, raw inner-product scoring is the wrong default; the mismatch is starkest under RMSNorm, which erases global scale entirely. It mixes content alignment with token norm, while the downstream transformer path removes most of that global scale before reading the residual stream. Replacing the score with a learned norm-scaled cosine keeps reconstruction matched, but changes which features the dictionary spends capacity on.

At matched FVE, KL/CE substitution scores, and per-feature LLM-judged interpretability, our cosine SAE recovers substantially better single-feature coverage of probing targets (§4.3, §4.1).

The evidence points to the forward-pass TopK selection geometry as the primary mechanism. Equalizing gradient magnitudes closes only 1.9–6.8% of the gap (§4.3): the inner-product score still inflates all pre-activations on high-norm tokens, BatchTopK still selects the same norm-conditioned features, and the same firing pattern reappears. This mechanism matters because pre-sublayer normalization has already discarded or attenuated token-level magnitude from the downstream computation; the norm-conditioned features that dominate the standard dictionary encode information the model does not fully read. Whether the gain is best understood as TopK selection geometry alone or as a combination of selection geometry and RMS-geometric alignment remains open; both point to the same practical recommendation: practitioners training SAEs on normalized models should use cosine scoring as the default encoder, with the strongest gains expected under RMSNorm.

We recommend the per-feature base+delta variant ($a_i = a_{\text{base}} + \delta_i$) as the default recipe: it gives the best probing gain ($+14.6\%$), the base+delta anchoring keeps it stable where free per-feature exponents collapse at deep layers (Appendix B.2.2), and its $1 + 2\,d_{\text{sae}}$ extra parameters are $< 0.1\%$ overhead. This is the recipe used for the headline runs; full training details and provenance are in Appendix J.

Concretely: unit-normalize encoder rows on each forward pass, replace the score with $e^b \|x_c\|^a \cos(x_c, w_i) + b_{\text{enc},i}$, and initialize $a{=}0$, $b{=}\log\sqrt{d_{\text{model}}}$. The global variant adds two scalar parameters; the per-feature variant parameterizes $a_i = a_{\text{base}} + \delta_i$ with per-feature b_i, adding $1 + 2\,d_{\text{sae}}$ parameters ($< 0.1\%$ overhead).

Impact Statement

This paper advances mechanistic interpretability and dictionary learning. We do not identify additional societal consequences beyond those generally associated with interpretability research.

References

Arora, A., Wu, Z., Steinhardt, J., and Schwettmann, S. Language model circuits are sparse in the neuron basis, 2026.

Basu, S. et al. Interpretability without actionability, 2026.

Borobia, A. et al. How pruning reshapes features, 2026.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N. L., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. Anthropic technical report; https://transformer-circuits.pub/2023/monosemantic-features, 2023.

Bussmann, B., Leask, P., and Nanda, N. BatchTopK sparse autoencoders, 2024.

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

Chanin, D. and Garriga-Alonso, A. SynthSAEBench: Evaluating sparse autoencoders on scalable realistic synthetic data, 2026.

Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., and Bloom, J. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Oral presentation.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models, 2023.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A., Caron, M., Geirhos, R., Alabdulmohsin, I., Jenatton, R., Beyer, L., Tschannen, M., Arnab, A., Wang, X., Riquelme, C., Minderer, M., Puigcerver, J., Evci, U., Kumar, M., van Steenkiste, S., Elsayed, G. F., Mahendran, A., Yu, F., Oliver, A., Huot, F., Bastings, J., Collier, M. P., Gritsenko, A., Birodkar, V., Vasconcelos, C., Tay, Y., Mensink, T., Kolesnikov, A., Pavetic, F., Tran, D., Kipf, T., Lucic, M., Zhai, X., Keysers, D., Harmsen, J., and Houlsby, N. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition. Anthropic technical report, 2022.

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

Friedman, D., Lampinen, A., Dixon, L., Chen, D., and Ghandeharioun, A. Interpretability illusions in the generalization of simplified models. In *International Conference on Machine Learning (ICML)*, 2024.

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

Grindrod, J. Sparse auto-encoders and holism about large language models, 2026.

Gulko, A., Peng, Y., and Kumar, S. CE-Bench: Towards a reliable contrastive evaluation benchmark of interpretability of sparse autoencoders, 2025. BlackboxNLP Workshop @ EMNLP.

Henry, A., Dachapally, P. R., Pawar, S., and Chen, Y. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP*, 2020.

Jedryszek, J. and Crook, O. Stable and steerable sparse autoencoders with weight regularization, 2026.

Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J., Chanin, D., Lau, Y.-T., Farrell, E., Conmy, A., McDougall, C., Lo Piano, F., Templeton, A., Marks, S., Wright, B., Bricken, T., Conerly, T., Smith, L., and Nanda, N. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *International Conference on Machine Learning (ICML)*, 2025.

Kempf, L. et al. Simple LLM baselines are competitive for model diffing, 2026.

Koromilas, P. et al. PolySAE: Modeling feature interactions in sparse autoencoders via polynomial decoding, 2026.

Korznikov, V., Belrose, N., and Sharkey, L. OrtSAE: Orthogonal sparse autoencoders uncover atomic features, 2025.

Leask, P., Bussmann, B., et al. Sparse autoencoders do not find canonical units of analysis. In *International Conference on Learning Representations (ICLR)*, 2025.

Liu, Z. Provably extracting the features from a general superposition, 2025.

Loshchilov, I., Hsieh, C.-P., Sun, S., and Ginsburg, B. nGPT: Normalized transformer with representation learning on the hypersphere, 2024.

Luo, H. et al. From atoms to trees: Building a structured feature forest with hierarchical sparse autoencoders, 2026.

Ma, L. et al. Falsifying sparse autoencoder reasoning features in language models, 2026.

Miller, A., Draye, Y., and Schölkopf, B. Identifying intervenable and interpretable features via orthogonality regularization, 2026.

Minder, J., Tao, F., Costa, J. B., Hannemann, T., Geiger, A., and Ravfogel, S. Overcoming sparsity artifacts in crosscoders to interpret chat-tuning, 2025.

Narayanaswamy, S. et al. Improving robustness in sparse autoencoders via masked regularization, 2026.

Nasiri-Sarvi, A. et al. MonoLoss: A training objective for interpretable monosemantic representations, 2026.

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

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

Prieto, L. et al. From data statistics to feature geometry: How correlations shape superposition, 2026.

Qwen Team. Qwen-Scope: Sparse autoencoders for Qwen3.5 models. Hugging Face collection; https://huggingface.co/collections/Qwen/qwen-scope, 2026. Accessed 2026-05-03.

Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramár, J., Shah, R., and Nanda, N. Improving dictionary learning with gated sparse autoencoders, 2024a.

Rajamanoharan, S., Lieberum, T., Sonnerat, N., Conmy, A., Varma, V., Kramár, J., and Nanda, N. Jumping ahead: Improving reconstruction fidelity with JumpReLU sparse autoencoders, 2024b.

Saraswatula, S. S. and Klindt, D. A. Data whitening improves sparse autoencoder learning, 2025.

Saurez, A., Lee, Y., and Har, D. Why linear interpretability works: Invariant subspaces as a result of architectural constraints, 2026.

Shafran, A. et al. From directions to regions: Decomposing activations in language models via local geometry, 2026.

Tian, C., Tian, K., and Hu, N. Measuring sparse autoencoder feature sensitivity, 2025.

Tigges, C., Hollinsworth, O. J., Geiger, A., and Nanda, N. Linear representations of sentiment in large language models, 2023.

Timkey, W. and van Schijndel, M. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Wang, J. et al. Dimensional collapse in transformer attention outputs: A challenge for sparse dictionary learning, 2025.

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, 2025.

Yang, W. et al. Step-level sparse autoencoder for reasoning process interpretation, 2026.

Zhang, B. and Sennrich, R. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Zhu, X., Khalili, M. M., and Zhu, Z. AbsTopK: Rethinking sparse autoencoders for bidirectional features, 2025.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., Goel, S., Li, N., Byun, M. J., Wang, Z., Mallen, A., Basart, S., Koyejo, S., Song, D., Fredrikson, M., Kolter, J. Z., and Hendrycks, D. Representation engineering: A top-down approach to AI transparency, 2023.

A. Extended Related Work

This appendix supplements the body Related Work (§2).

Residual-stream geometry. Whitening preprocessing (Saraswatula & Klindt, 2025) and dimensional collapse in attention outputs (Wang et al., 2025); representation engineering (Zou et al., 2023), sentiment-direction work (Tigges et al., 2023), and invariant subspaces forced by linear interfaces (Saurez et al., 2026).

Adjacent objective-level work. Latent-scaling crosscoders (Minder et al., 2025) and MonoLoss (vision SAEs) (Nasiri-Sarvi et al., 2026).

Alternative SAE benchmarks. CE-Bench (Gulko et al., 2025); SynthSAEBench (Chanin & Garriga-Alonso, 2026).

Concurrent failure modes. Feature absorption (Chanin et al., 2025) and proposed fixes via masked regularization (Narayanaswamy et al., 2026), orthogonality (Miller et al., 2026), hierarchical decoders (Luo et al., 2026), weight regularization (Jedryszek & Crook, 2026); polynomial decoders (Koromilas et al., 2026), correlation-shaped superposition (Prieto et al., 2026), provable feature extraction (Liu, 2025); sensitivity gaps (Tian et al., 2025), superficial reasoning features (Ma et al., 2026), pruning robustness (Borobia et al., 2026); multi-dimensional features (Engels et al., 2025), region-based decompositions (Shafran et al., 2026).

SAE limitations and baselines. Prompting outperforms SAE steering (Wu et al., 2025); raw MLP neurons match SAE sparsity for circuit discovery (Arora et al., 2026); LLM baselines match SAE-based model diffing (Kempf et al., 2026); SAE features fail to translate into actionable corrections (Basu et al., 2026); interpretability illusions OOD (Friedman et al., 2024). SAEs remain in use for polysemantic neurons (Cunningham et al., 2023; Grindrod, 2026); reasoning-process SAEs (Yang et al., 2026).

Position relative to prior SAE variants. Only the encoder score changes; the sparsity mechanism, decoder, and training objective are held fixed. The intervention is compatible with concurrent improvements (Matryoshka, JumpReLU, AbsTopK, masked regularization). Released SAE suites for the Qwen family (Qwen Team, 2026) are available for large Qwen-family runs.

B. Cosine SAE Architecture

This appendix provides the formal variant definitions for reproducibility (§B.1), documents the design-space exploration that led to the final architecture (§B.2), and reports what the optimizer learns when given freedom (§B.3).

B.1. Variant Definitions

Notation: $x \in \mathbb{R}^d$ residual-stream activation; $x_c = x - b_{\text{dec}}$; $W_{\text{enc}}, W_{\text{dec}} \in \mathbb{R}^{d_{\text{sae}} \times d}$; $\text{normalize}(\cdot)$ is ℓ_2-normalization along the last axis. All variants share BatchTopK ($k = 80$), optimizer, training data, token budget, and the recipe of §J.

Standard BatchTopK SAE (baseline).

$$
\begin{aligned}
z &= \text{BatchTopK}\big(\text{ReLU}(x_c W_{\text{enc}}^\top + b_{\text{enc}})\big), \\
\hat{x} &= z W_{\text{dec}} + b_{\text{dec}}, \qquad \mathcal{L} = \|x - \hat{x}\|^2.
\end{aligned}
$$

Encoder rows unconstrained: $s_i(x) = \|W_{\text{enc},i}\| \, \|x_c\| \, \cos(W_{\text{enc},i}, x_c)$.

Adaptive Cosine SAE (global a).

$$
\begin{aligned}
x_{\text{unit}} &= \text{normalize}(x_c), \\
W_{\text{unit}} &= \text{normalize}(W_{\text{enc}}, \text{dim}{=}{-1}), \\
\cos(x_c, w_i) &= (x_{\text{unit}} W_{\text{unit}}^\top)_i, \\
\gamma(x) &= \exp\big(a \log \|x_c\| + b\big), \quad a, b \in \mathbb{R}, \\
s_i(x) &= \gamma(x) \, \cos(x_c, w_i) + b_{\text{enc},i}, \\
z &= \text{BatchTopK}\big(\text{ReLU}(s)\big), \\
\hat{x} &= z W_{\text{dec}} + b_{\text{dec}}, \quad \mathcal{L} = \|x - \hat{x}\|^2.
\end{aligned}
$$

At $a = 0$, $s_i = e^b \cos(x_c, w_i)$; at $a = 1$, $s_i = e^b \langle W_{\text{unit},i}, x_c \rangle$. Init: $a = 0$, $b = \log \sqrt{d_{\text{model}}}$. Added params: 2 scalars.

Per-Feature Adaptive Cosine SAE (base+delta).

$$\begin{aligned} a_i &= a_{\text{base}} + \delta_i, \quad a_{\text{base}}, \{\delta_i\} \in \mathbb{R}^{1+d_{\text{sae}}}, \\ \gamma_i(x) &= \exp\big(a_i \log \|x_c\| + b_i\big), \\ s_i(x) &= \gamma_i(x)\ \cos(x_c, w_i) + b_{\text{enc},i}. \end{aligned}$$

Encode/decode/loss as Adaptive Cosine. Init: $a_{\text{base}} = 0$, $\delta_i = 0$, $b_i = \log \sqrt{d_{\text{model}}}$. Added params: $2d_{\text{sae}} + 1$ scalars ($\leq 0.1\%$). The base+delta parameterization prevents per-feature instability at deep layers (§B.2.2).

Magnitude-Bypass SAE.

$$\begin{aligned} x_{\text{unit}} &= x_c / \|x_c\|, \quad W_{\text{unit}} = \text{normalize}(W_{\text{enc}}, \text{dim}{=}{-1}), \\ z &= \text{BatchTopK}\big(\text{ReLU}(x_{\text{unit}} W_{\text{unit}}^\top)\big), \\ W_{\text{dec}}^u &= \text{normalize}(W_{\text{dec}}, \text{dim}{=}{-1}), \\ x_{\text{raw}} &= z W_{\text{dec}}^u, \\ \hat{x} &= \|x_c\| \cdot \frac{x_{\text{raw}}}{\|x_{\text{raw}}\|} + b_{\text{dec}}, \quad \mathcal{L} = \|x - \hat{x}\|^2. \end{aligned}$$

Encoder activations bounded in $[0, 1]$; magnitude $\|x_c\|$ enters only the decoder rescale. The sparse code carries no magnitude information.

Gradient asymmetry. For an active feature i:

$$\frac{\partial s_i^{\text{std}}}{\partial W_{\text{enc},i}} = x_c \quad \Rightarrow \quad \left\| \frac{\partial s_i^{\text{std}}}{\partial W_{\text{enc},i}} \right\| = \|x_c\|.$$

For all cosine variants, s_i depends on x_c through $x_c / \|x_c\|$ and a scalar, so the encoder-gradient norm is bounded independently of input magnitude.

Held fixed across variants. BatchTopK $k = 80$; geometric-median init of b_{dec}; decoder columns Kaiming + unit-norm; encoder $= W_{\text{dec}}^\top$ ($0.1 \cdot W_{\text{dec}}^\top$ for Standard); Adam, learning rate, schedule; token budget; d_{sae}. All † and unmarked rows use the AuxK auxiliary loss of Gao et al. (2024); the ‡ rows (from an early 5M-token sweep) predate its adoption but exp46 confirms aux-k is a no-op for cosine-encoder architectures at that budget (§B.2.3).

B.2. Design-Space Exploration

The final architecture emerged from a series of failures. Each design choice prevents a specific collapse mode. Table 2 summarizes the full ablation; the subsections below explain the rationale.

B.2.1. Why the scale parameter is necessary

The most natural first attempt is pure cosine scoring: normalize both the input and the encoder rows, so $s_i = \cos(x_c, w_i) + b_{\text{enc},i}$. This removes all norm dependence from feature selection. It also fails catastrophically: FVE $= 0.009$ with 97% dead features (Table 2, row "naive cosine"). [exp10]

The failure is a reconstruction-scale mismatch. Cosine scores are bounded in $[-1, 1]$; after BatchTopK and ReLU, the sparse code z has entries in $[0, 1]$. The decoder must reconstruct x at its full scale ($\|x_c\| \sim 64$–400 depending on layer), but zW_{dec} produces vectors of order $\|z\|_1 \cdot 1 \approx k = 80$. At shallow layers where $\|x_c\| \approx 64$ this nearly works; at L27 where $\|x_c\| \approx 407$ the decoder cannot bridge the gap. Features receive near-zero gradient because reconstruction error is dominated by the global scale mismatch rather than directional errors, and 97% die within the first few thousand steps.

Adding a learned global intercept b (so the scale is e^b, a single scalar) partially recovers: FVE rises to 0.449 but 76% of features still die (row "global b, no a"). The intercept provides a fixed boost but cannot adapt to the token-level norm variation that the decoder needs. Freeing the exponent a (so scale $= e^{a \log \|x_c\| + b} = e^b \|x_c\|^a$) lets the encoder pass exactly as much per-token magnitude as reconstruction requires. The optimizer converges to $a \approx 0.26$ globally and $\bar{a}_i \approx 0.08$ per-feature; 0% dead, FVE $= 0.77$. [exp12] [exp42c]

Table 2. Design-space ablation. Headline setting: Qwen3-8B L18, 500M tokens; † from L27/50M (same recipe); ‡ from L27/5M (early sweep; no aux-k, but exp46 confirms aux-k is a no-op at this budget). *Cos>inner* is the per-feature win rate (%) on the diagnostic of §C. [exp40/42c/44] [exp46] [exp48]

Variant	Design knobs						Metrics				
	Input norm	Unit enc.	Unit dec.	Restore $\|x\|$	Scale	Enc. bias	FVE	Dead%	Probe top-1	Cos>inner	Learned a
inner-product baseline	—	—	✓	—	none	✓	0.770	0.0	0.667	62	—
naive cosine (×)	✓	✓	✓	—	none	✓	$0.009^{\ddagger}$	$97.0^{\ddagger}$	—	—	—
global b, no a	✓	✓	✓	—	global b	✓	$0.449^{\ddagger}$	$76^{\ddagger}$	—	—	—
global a encoder	✓	✓	✓	—	global a, b	✓	0.769	0.0	0.800	63	0.258
per-feature a_i (no anchor)	✓	✓	✓	—	per-feat a_i, b_i	✓	$0.721^{\dagger}$	$83.4^{\dagger}$	—	$71^{\dagger}$	0.022
per-feature base+delta	✓	✓	✓	—	base+δ_i	✓	**0.771**	0.0	**0.815**	62	0.076
no enc. norm, per-feat b_i	✓	—	✓	—	per-feat b_i	✓	$0.770^{\dagger}$	$0.4^{\dagger}$	—	$61^{\dagger}$	$0.201^{\dagger}$
no enc. norm, global a, b (×)	✓	—	✓	—	global a, b	✓	$0.614^{\dagger}$	$93.2^{\dagger}$	—	—	$0.620^{\dagger}$
magnitude-bypass	✓	✓	✓	✓	none	—	0.767	4.3	0.783	69	—
encoder rows free	✓	—	✓	✓	none	—	$0.754^{\dagger}$	$0.3^{\dagger}$	—	$67^{\dagger}$	—
decoder rows free	✓	✓	—	✓	none	—	$0.550^{\ddagger}$	$0.0^{\ddagger}$	—	—	—
both encoder/decoder free	✓	—	—	✓	none	—	$0.554^{\ddagger}$	$0.0^{\ddagger}$	—	—	—
w/o norm restore (×)	✓	✓	—	—	none	—	$0.082^{\ddagger}$	$88.5^{\ddagger}$	—	—	—
input norm only, no scale (×)	✓	—	✓	—	none	✓	$0.297^{\dagger}$	$93.5^{\dagger}$	—	$58^{\dagger}$	—
no enc. norm, base+δ_i	✓	—	—	✓	base+δ_i	✓	$0.740^{\dagger}$	$20.0^{\dagger}$	—	$61^{\dagger}$	$-0.59^{\dagger}$

B.2.2. Per-feature instability and the base+delta fix

Allowing each feature its own exponent a_i is appealing: a positional feature may need more norm information than a semantic one. The direct parameterization (a_i free, initialized at 0) works at shallow layers and short budgets (L9/5M: 0% dead, matching the global variant). At L27 with 50M tokens, it collapses: 83.4% dead (54,657 of 65,536 features). This cascade occurs even with encoder unit-normalization active and is specific to the per-feature degree of freedom in the exponent, not to missing normalization constraints. [exp47]

The failure mechanism is a winner-take-all cascade. At L27, $\|x_c\| \approx 407$ while the initial scale $e^b = \sqrt{d} = 64$, a $6.4\times$ mismatch. Some features randomly receive slightly more gradient early in training (due to initialization variance in W_{enc} directions), fire more often, and grow their a_i to capture more of the high-norm tokens. As their a_i increases, their pre-activations on high-norm tokens inflate further, raising the batch-wide TopK threshold. Features that started with slightly less gradient now fall below the threshold, receive zero gradient, and die. The cascade is self-reinforcing: at step 5,500, approximately 43,000 features die simultaneously in a single 500-step window. By step 10,000 the dictionary has 83% dead features and cannot recover.

The base+delta parameterization $a_i = a_{\text{base}} + \delta_i$ prevents this cascade. The shared a_{base} moves all features together during early training, ensuring the global scale tracks $\|x_c\|$ before any per-feature divergence can occur. Once a_{base} has converged (typically by step 5,000), the per-feature δ_i offsets begin differentiating. The result: 0.4% dead at L27/50M (vs. 83.4% unconstrained), with FVE = 0.772 vs. 0.721. [exp47]

At the headline setting (L18, 500M tokens), base+delta achieves the best sparse probing in the entire table (0.815) at 0% dead and matched FVE (0.771).

B.2.3. Stabilization mechanisms: encoder normalization and norm restoration

Once input normalization strips $\|x\|$ from the encoder's view, something must prevent norm-dominance from re-entering through other paths. Two mechanisms independently solve this problem: (1) encoder unit-normalization, which prevents encoder-row magnitudes from drifting and recapitulating norm dominance on the weight side; and (2) post-decode norm restoration, which rescales the reconstruction to match $\|x_c\|$ externally. Either one suffices; removing both is catastrophic.

Encoder normalization. Unit-normalizing encoder rows ($w_i \leftarrow w_i/\|w_i\|$ on each forward pass) ensures the cosine score is a true cosine similarity. Dropping this constraint while keeping input normalization (row "input norm only, no scale") gives FVE = 0.297 at 93.5% dead. The failure mode: without $\|w_i\| = 1$, encoder row norms drift freely. Rows that happen to grow large dominate the TopK competition (their pre-activations scale with $\|w_i\|$), starving smaller rows of gradient. This recapitulates the same norm-dominance pathology we set out to fix, but now on the encoder side rather than the input side. The collapse is layer-independent: free-encoder adaptive cosine dies at 90.8% (L9), 94.6% (L18), and 93.2% (L27), all at 50M tokens. [exp23] [exp45]

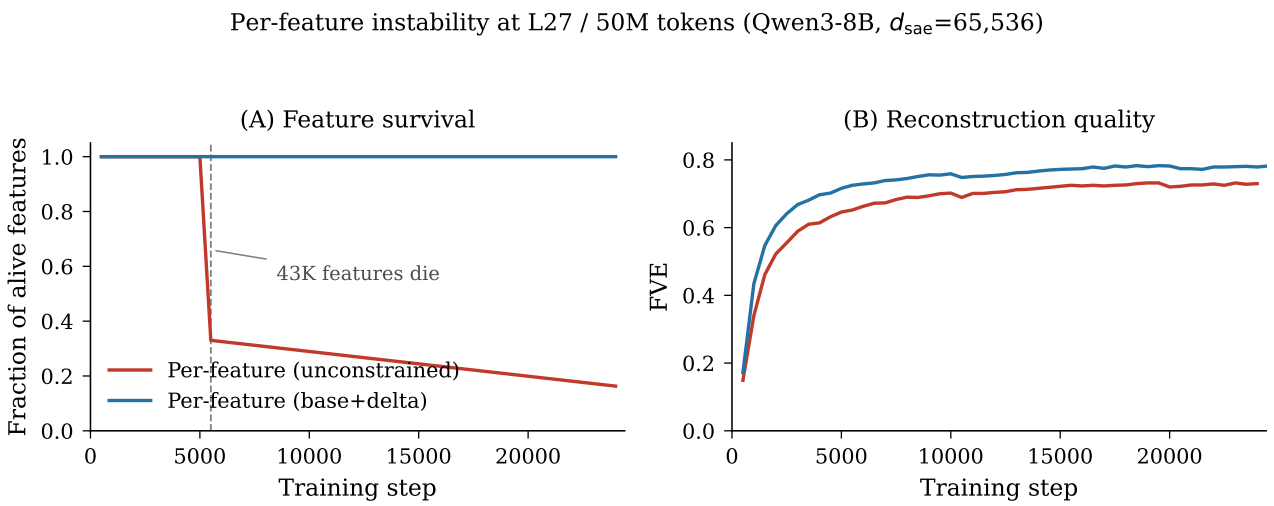

Figure 10. **Winner-take-all cascade in unconstrained per-feature a_i.** Qwen3-8B L27, 50M tokens. **(A)** Unconstrained per-feature (red) loses 67% of features in a single 500-step window at step 5,500; base+delta (blue) retains all features throughout. **(B)** Base+delta also achieves higher final FVE (0.77 vs. 0.72) because surviving features encode useful content rather than fighting for norm-dominated TopK slots. [exp47]

Releasing encoder normalization while keeping post-decode norm restoration and a global learned a, b (row "no enc. norm, global a, b") also collapses: 93.2% dead, with a running away to 0.62 (toward inner product). The restoration step helps the decoder but cannot prevent the encoder-side norm drift that kills feature selection. The one exception: per-feature b_i with free encoder rows achieves 0.4% dead (row "no enc. norm, per-feat b_i"). Here each feature's individual scale e^{b_i} can compensate for encoder-row magnitude variation, effectively absorbing $\|w_i\|$ into b_i. This works but at a cost: FVE matches the baseline (0.770); sparse-probing performance is untested for this variant. [exp46]

Norm restoration. Within the magnitude-bypass family (no learned a, sparse codes bounded in $[0, 1]$), norm restoration is the load-bearing stabilizer. A factorial ablation at L27/5M shows: restoration-on variants all land within 0.8% FVE (0.550–0.558) at 0% dead regardless of encoder/decoder normalization; restoration-off variants collapse to $\sim$ 89% dead with FVE 0.08–0.14 (Figure 11). Encoder and decoder unit-norm constraints are decorative when restoration is active. [exp46]

Learned exponent as a third path. With a learned exponent a (the adaptive and per-feature variants), neither encoder normalization nor norm restoration is strictly necessary. The per-feature base+delta variant achieves 0% dead and FVE 0.771 with encoder normalization but without restoration (Table 2, row 6). Conversely, base+delta without encoder normalization but with restoration achieves FVE $= 0.740$ at 20% dead (row "no enc. norm, base+δ_i"); functional but degraded. The learned exponent provides partial self-stabilization: pre-activations scale as $\|x_c\|^{a_i}$, so the sparse code carries per-token magnitude directly and the decoder can reconstruct at the correct scale without an external rescale.

The base+delta parameterization is uniquely robust because the shared a_{base} anchors all features to a common scale (preventing the encoder-side divergence that kills free-encoder variants) while the per-feature δ_i offsets allow heterogeneity without competitive pressure. This is why it survives with only one stabilizer active. Other parameterizations (global a, unconstrained per-feature) require encoder normalization.

Summary and open questions. In practice, our recommended variants use encoder normalization (which is cheap and guarantees true cosine geometry). The adaptive/per-feature family does not use norm restoration; the magnitude-bypass family relies on it. We have not ablated all pairwise combinations at the headline setting. Specifically, (1) per-feature base+delta with both encoder normalization and norm restoration active, and (2) per-feature base+delta without either, remain untested at 500M tokens. Whether the 20% dead rate of the no-enc-norm variant (exp48, L27/50M) improves with longer training or worsens is unknown. These are natural directions for future exploration.

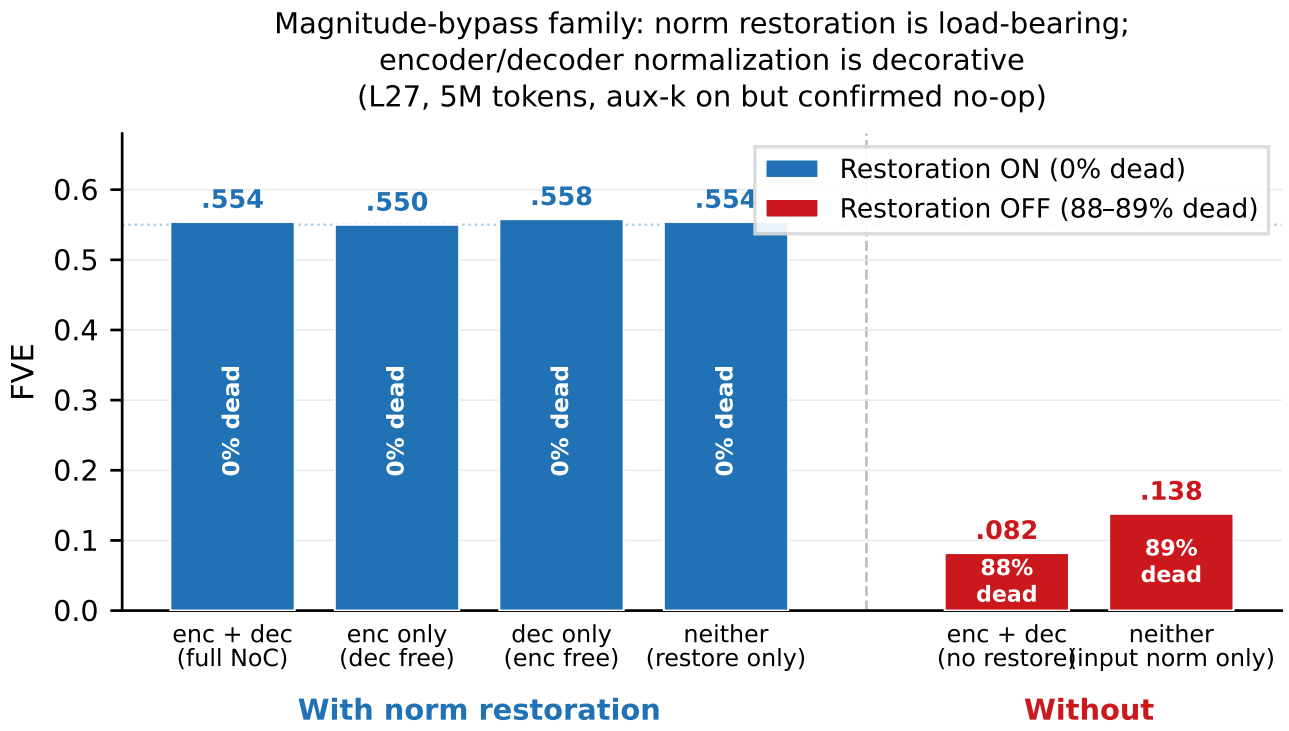

Figure 11. **Magnitude-bypass family: norm restoration is load-bearing; encoder/decoder normalization is decorative.** L27, 5M tokens, aux-k on (confirmed no-op). All four restoration-on variants (blue) achieve FVE ≈ 0.55 at 0% dead regardless of enc/dec norm. Restoration-off variants (red) collapse to $\geq 88\%$ dead. For the adaptive family, encoder normalization plays the equivalent stabilizing role (see text above). [exp46]

B.2.4. Initialization sensitivity

The scale parameter b is initialized to $\log \sqrt{d_{\text{model}}}$, which sets the initial pre-activation magnitude to $\sqrt{d} \cdot \cos(\theta) \approx 64$ for Qwen3-8B. This works when residual-stream norms are of similar order or larger (Qwen L9: $\|x\| \approx 58$, L27: ≈ 407). It fails catastrophically when norms are much smaller. [exp42]

On Mistral-7B, activation norms at L8 are only 6.3 ($10\times$ smaller than $\sqrt{d} = 64$). The $\sqrt{d}$ initialization overshoots: pre-activations are an order of magnitude too large, features saturate and die immediately (100% dead at L8, 97.6% at L16). No gradient flows from dead features, so the optimizer cannot recover. This is the opposite failure mode from Qwen L27, where $\sqrt{d}$ *undershoots* ($\|x\| = 407$ vs. $\sqrt{d} = 64$) but features stay alive and the optimizer climbs toward the correct scale. [exp25]

The asymmetry: undershoot is recoverable (gradients still flow), overshoot is not (dead features produce no gradient). We resolve this with a norm-adaptive initialization rule: if the mean centered activation norm $\text{mean}\,\|x_{\text{train}} - b_{\text{dec}}\|$ differs from $\sqrt{d_{\text{model}}}$ by more than $2\times$, use $b = \log(\text{mean}\,\|x\|)$ instead. With this fix, Mistral achieves positive cosine advantage at all layers ($+1.9\%$ FVE at L8, $+3.0\%$ at L24, 55% fewer dead features). [exp42]

A subtlety at longer budgets: on Qwen L27 at 50M tokens, norm-adaptive init hurts relative to $\sqrt{d}$ because it removes the gradient pressure that teaches a to compensate for scale (the optimizer "needs to struggle" with a wrong b to discover that magnitude matters). The $\sqrt{d}$ default works when norms exceed the init; norm-adaptive is necessary only in the overshoot regime. [exp34] [exp41]

B.2.5. Group-size experiments

Rather than one global a or 65,536 free a_i values, an intermediate option shares each a across a group of G features. We tested $G \in \{1, 4, 16, 64, 256, 9216\}$ at L13 on Gemma-2-2B/50M tokens. [exp37]

$G = 4$ (4 groups of 2,304 features) achieves the best SAEBench composite in this sweep: KL-score 0.9785, CE-score

Table 3. Initialization sensitivity across models. The direction of the mismatch determines recoverability. [exp42] [exp42c]

Model	Layer	$\|x\|$	Init	Tokens	Outcome
Qwen	L27	407	$\sqrt{d}$ (undershoot)	500M	best
Qwen	L27	407	norm-adaptive	50M	hurts (a stuck)
Mistral	L8	6.3	$\sqrt{d}$ (overshoot)	5M	100% dead
Mistral	all	varied	norm-adaptive	50M	cosine wins

0.9786, RAVEL 0.619, outperforming both the fully global setting ($G = 1$: KL 0.9767, RAVEL 0.618) and the fully free setting ($G = 9216$: KL 0.9783, RAVEL 0.609). Dead-feature rates decrease monotonically with group size (41% at $G = 1$ to 20% at $G = 9216$), but downstream task performance is non-monotone.

We note two caveats. First, this experiment uses Gemma-2-2B at 50M tokens with a smaller dictionary ($d_{\text{sae}} = 9{,}216$); results may not transfer directly to the headline Qwen3-8B/65,536 setting. Second, differences between adjacent group sizes are small (within 0.002 on KL-score), so the $G = 4$ optimum is suggestive rather than definitive. The broader conclusion is more robust: some per-feature heterogeneity in norm dependence improves downstream metrics relative to a single global a, while fully unconstrained per-feature freedom can hurt.

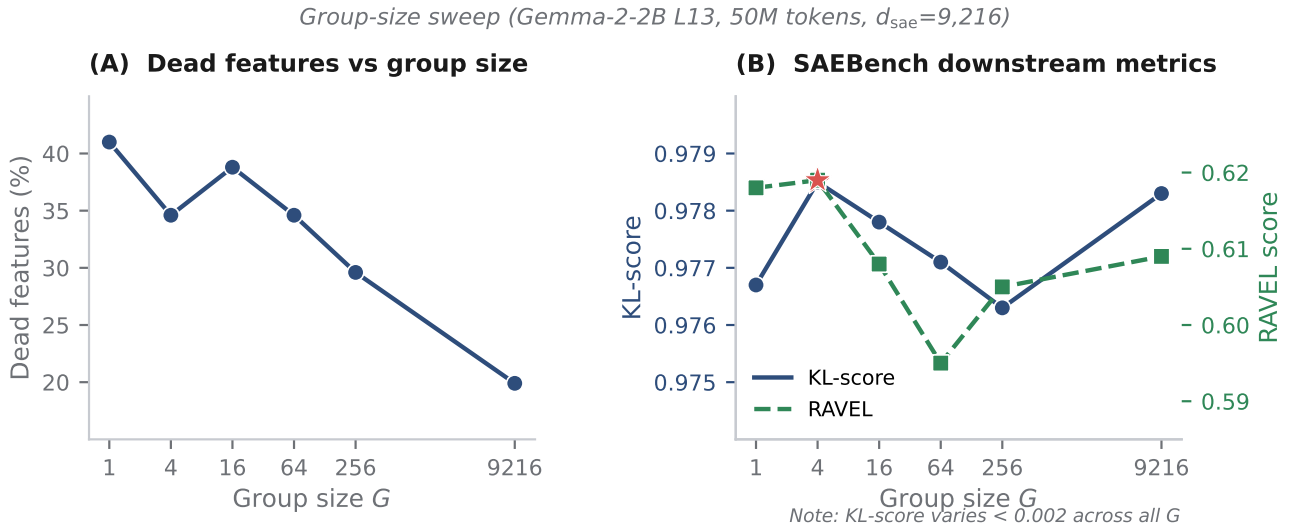

Figure 12. **Group-size sweep.** Gemma-2-2B L13, 50M tokens, $d_{\text{sae}} = 9{,}216$. **(A)** Dead-feature rate decreases monotonically with group size. **(B)** KL-score and RAVEL peak at $G = 4$ (star), though differences are small (< 0.002 KL-score between adjacent sizes). The result is suggestive of an intermediate optimum but has not been replicated at the headline scale. [exp37]

In practice, we use per-feature base+delta rather than group-sharing because: (a) it achieves 0% dead at the headline setting; (b) the shared base provides a similar anchoring effect to small group sizes; (c) it adds negligible parameter overhead ($2d_{\text{sae}}$ scalars). A systematic comparison of group-sharing vs. base+delta at the headline scale (Qwen3-8B, 500M tokens, $d_{\text{sae}} = 65{,}536$) remains a natural future direction.

B.2.6. Summary of design lessons

1. Pure cosine (no scale) fails because the decoder cannot bridge the norm gap between bounded sparse codes and real activation magnitudes (§B.2.1).
2. The learned exponent a lets the encoder pass exactly as much magnitude as reconstruction requires; the optimizer consistently chooses $a \ll 1$ (§B.3).
3. Per-feature a_i must be anchored (base+delta) to prevent winner-take-all cascades at deep layers where $\|x_c\| \gg \sqrt{d}$ (§B.2.2).
4. Either encoder unit-normalization or post-decode norm restoration prevents collapse; the magnitude-bypass family

requires restoration, while the adaptive family requires encoder norm. Per-feature base+delta can partially survive with only one, but works best with encoder normalization (§B.2.3).

5. Moderate per-feature freedom (base+delta or $G = 4$) outperforms both fully global and fully free parameterizations (§B.2.5).
6. Initialization must not overshoot: $\sqrt{d}$ works when norms exceed the init; norm-adaptive is needed otherwise (§B.2.4).

B.3. Empirical Convergence

What does the optimizer choose when given freedom? This section reports the learned values of a and a_i at the headline setting.

Table 4. Learned a across layers (Qwen3-8B, community recipe, 500M tokens). [exp42c]

Layer	Variant	a (or mean a_i)	% near 0	Max a_i
L9	Global	0.025	—	—
L9	Per-Feature	-0.011	—	< 0.5
L18	Global	0.258	—	—
L18	Per-Feature	0.076	23%	< 0.5
L27	Global	0.257	—	—
L27	Per-Feature	0.362	23%	< 0.5

At L9 the optimizer drives a to near zero (pure cosine); at deeper layers it retains modest norm dependence ($a \approx 0.26$). No feature at any layer or token budget learns $a_i > 0.5$; the inner-product regime ($a = 1$) is never preferred. Per-feature means are consistently lower than the global a, suggesting that when features can individually tune their norm dependence, most want less magnitude rather than more.

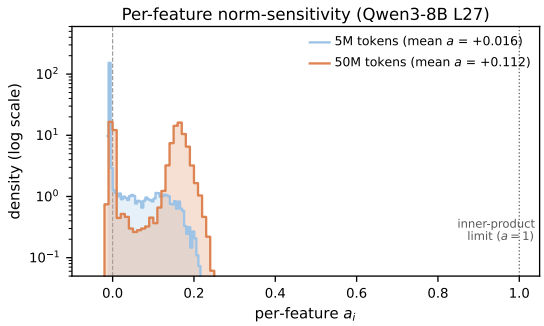

Figure 13. Per-feature a_i distribution at three token budgets (Qwen3-8B L27). Mass shifts toward larger a_i with more data; no $a_i > 0.5$ in any setting. [exp42c]

Table 5. $\{a_i\}$ statistics across token budgets. [exp42c] [exp16] [exp17]

Setting	Tokens	Mean a_i	% near 0	Max a_i
Qwen L27	5M	≈ 0.01	88–99%	< 0.5
Qwen L27	50M	0.113	60%	< 0.5
Qwen L27	500M	0.362	23%	< 0.5
Qwen L18	500M	0.076	23%	< 0.5

At short budgets (5M), 88–99% of features remain near $a_i \approx 0$. With more data, the distribution shifts: at 500M only 23% stay near zero. This suggests that early in training, features learn content directions first (cosine is sufficient), and only later differentiate their norm dependence as finer-grained reconstruction demands emerge. Even at 500M, the distribution remains far below the inner-product limit ($a = 1$), confirming that the optimizer consistently prefers direction-dominated scoring.

Pinned cosine ($a = 0$). Freezing a at 0 and training only b produces a fixed-scale cosine encoder. The trained-$\{a_i\}$ distributions above show this is suboptimal: the optimizer wants $a > 0$ at deeper layers, and pinned cosine cannot adapt to

per-token norm variation. This variant is included in the ablation table for completeness but is not recommended; global a adds one scalar and strictly dominates.

B.4. Beyond BatchTopK: Other Selectors

The main text uses BatchTopK throughout. Because the cosine score modifies only the encoder pre-activation and not the sparsity mechanism, it composes with any selector. We test whether the cosine advantage is specific to BatchTopK's batch-wide competition or reflects inner-product scoring more generally, by re-running the headline comparison under two additional per-token selectors at the same setting (Qwen3-8B L18, 50M tokens, $d_{\text{sae}} = 65{,}536$, matched $L_0 = 80$):

- **Per-token TopK** (Gao et al., 2024): keep the k largest activations *per token* rather than a single batch-wide budget.
- **AbsTopK** (Zhu et al., 2025): keep the k largest by absolute value per token, preserving sign.

Table 6. **The cosine advantage persists across selectors, and decomposes into two magnitude channels.** Sparse-probing top-1 at matched $L_0 = 80$ (Qwen3-8B L18, 50M tokens). The cosine encoder improves top-1 over inner-product under every selector, not only BatchTopK. The *unit-enc* arms are inner-product with ℓ_2-normalized encoder rows: they remove the weight-norm $\|w_i\|$ but keep the input norm $\|x\|$. Removing $\|w_i\|$ alone rescues features from the dead state (decisively under AbsTopK: $92.4\% \to 0.0\%$) but does *not* recover probing quality, which stays at the inner-product level; only the cosine score, which also removes $\|x\|$, recovers probing. [exp63]

Selector	Encoder	FVE	Dead%	Probe top-1
BatchTopK	inner-product	0.723	0.0	0.530
BatchTopK	cosine (per-feat)	0.726	0.0	**0.648**
per-token TopK	inner-product	0.641	91.7	0.731
per-token TopK	+ unit-enc	0.680	80.5	0.645
per-token TopK	cosine (global)	0.711	6.1	**0.802**
AbsTopK	inner-product	0.636	92.4	0.690
AbsTopK	+ unit-enc	0.670	0.0	0.668
AbsTopK	cosine (per-feat)	0.692	6.2	**0.827**

The advantage is not BatchTopK-specific. The cosine encoder improves sparse-probing top-1 over the inner-product encoder under all three selectors (Table 6): $+11.8$ points under BatchTopK, $+5$ to $+7$ under per-token TopK, and $+12$ to $+14$ under AbsTopK. The effect therefore reflects inner-product scoring in general, not the batch-wide competition specific to BatchTopK.

Batch-wide competition explains part, but not all, of the gap. Our mechanism account (§2) predicts that per-token selection should *reduce* the cosine advantage: when each token receives exactly k slots, the per-token input-norm scalar multiplies every feature's inner-product score equally and cancels from the within-token ranking, removing the channel by which high-norm tokens claim disproportionate slots. The advantage does shrink under per-token TopK ($+11.8 \to +5$–7), consistent with this account. It does not vanish, and under AbsTopK it is as large as under BatchTopK, so a selector-independent component remains. The decomposition below attributes that residual to the second magnitude channel, the encoder-row norm $\|w_i\|$, which per-token selection does not neutralize.

Two magnitude channels: weight norm drives feature death, input norm drives probing quality. The inner-product score $\|x\|\,\|w_i\|\cos(x, w_i)$ carries magnitude from two sources: the input norm $\|x\|$ and the encoder-row norm $\|w_i\|$. The *unit-enc* arms isolate them, normalizing w_i (removing $\|w_i\|$) while leaving $\|x\|$ in the score; under per-token selection $\|x\|$ multiplies every feature equally and cancels from the within-token ranking, so unit-enc effectively ranks by $\|w_i\|$-free content within each token. Two things separate:

- **Dead features are a weight-norm effect.** Under per-token selection the inner-product encoder loses most of its dictionary (91.7% and 92.4% dead, despite the same auxiliary dead-feature loss used everywhere): a few large-norm rows win slots on most tokens and starve the rest. Unit-normalizing the encoder rows rescues them, decisively under AbsTopK ($92.4\% \to 0.0\%$ dead) and partially under per-token TopK ($91.7\% \to 80.5\%$).
- **Probing quality is an input-norm effect.** Rescuing the features does *not* make them more interpretable: unit-enc probing stays at the inner-product level (0.645 and 0.668), far below cosine (0.802, 0.827). Only the cosine score, which additionally strips $\|x\|$, recovers probing. As long as the score scales with token magnitude, the learned features remain

norm-conditioned regardless of whether they survive.

Cosine scoring is the only encoder here that removes *both* magnitude sources, which is why it alone wins on feature survival *and* probing.

Untested: penalty-trained selectors. We did not obtain matched-sparsity JumpReLU (Rajamanoharan et al., 2024b) or Gated (Rajamanoharan et al., 2024a) results. Unlike the fixed-k selectors above, these reach a target L_0 through a sparsity penalty rather than a hard budget. At our 50M-token mechanism-sweep budget the learned threshold initializes correctly at $L_0 \approx 80$, but reconstruction pressure drives it back up within a few hundred steps and the straight-through penalty cannot hold it: across penalty strengths spanning more than a $30\times$ range, and at two dictionary sizes ($d_{\text{sae}} = 16{,}384$ and $65{,}536$), the converged L_0 stayed in the thousands rather than near 80. Reaching matched sparsity requires the substantially longer training schedules used in their original work, so a like-for-like comparison is left to future work.

C. Diagnostic: Cosine vs. Inner Product as a Causal Predictor

Extended version of §4.2. Given fixed decoder directions from a trained Standard SAE, does cosine score have higher absolute correlation with a causal projection-ablation effect than inner product?

Definitions. *Projection ablation*: for unit decoder direction d_f, $\text{ablate}(x; f) = x - (d_f^\top x)\, d_f$, re-installed via a forward hook at the ablation layer. *KL-divergence ablation effect*: $y_t = \text{KL}\big(\, p_{\text{orig}}(\cdot \mid x_t) \,\|\, p_{\text{abl}}(\cdot \mid x_t; f)\,\big)$.

Win-rate protocol. For each feature f with unit decoder direction d_f and m held-out tokens: $s_t^{\text{cos}} = \cos(x_t, d_f)$, $s_t^{\text{ip}} = \langle x_t, d_f \rangle$, and y_t is the KL-divergence ablation effect defined above. Decoder directions d_f are taken from trained Standard SAEs. Cosine wins iff $|\text{corr}(s^{\text{cos}}, y)| > |\text{corr}(s^{\text{ip}}, y)|$, averaged over features. The 50% null corresponds to no systematic advantage for either scoring rule. Projection ablation makes the perturbation magnitude exactly $\langle x, d_f \rangle$, biasing KL toward $|\langle x, d_f \rangle|$, which means the diagnostic is *conservative* with respect to cosine (Figure 14).

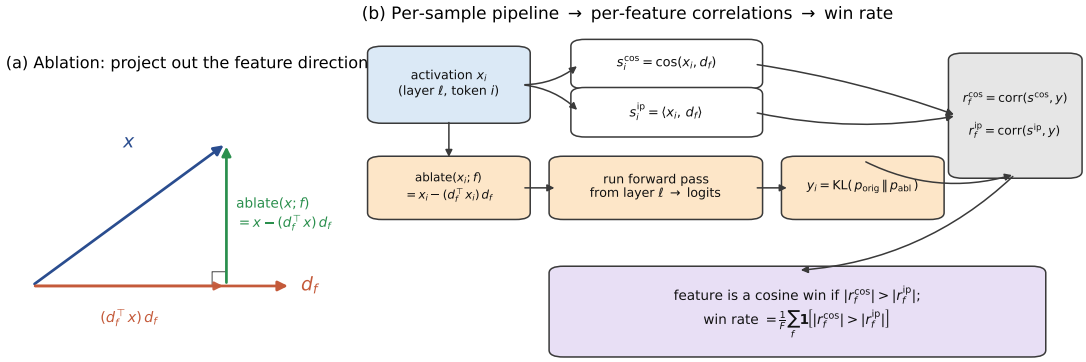

Figure 14. Win-rate diagnostic protocol. **(a)** Projection ablation removes a decoder direction from the residual stream; the downstream KL-divergence measures its causal importance. **(b)** Per-sample cosine and inner-product scores are correlated with this causal effect; the scoring rule with higher absolute correlation "wins" for that feature.

Depth dependence on RMSNorm. Qwen3-8B: L9 $\approx 44\%$, L18 $\approx 65\%$, L27 $\approx 78\%$. Norm-patching at L18 produces $\sim 10\times$ less downstream-KL change than direction-patching at fixed norm. Cross-model RMSNorm: Mistral-7B 74–88%; Gemma-2-2B matches Qwen with a low value at L20. LayerNorm (Pythia-2.8B / Pythia-6.9B / Falcon-7B) drops from $\sim 90\%$ shallow to 40–57% deep (Fig. 15). [exp25] On constructed token pairs (high-cosine / low-norm vs. low-cosine / high-norm), the high-cosine token has the larger KL effect on $\sim 90\%$ of features at deep layers.

Caveats. Within a single norm quartile the rate drops to 40–70%, so the 70–90% rate is partly between-quartile (Simpson's paradox; Figure 16); the training-time results in §4.1 do not depend on this measurement. The inflation gap grows with depth (+18% at L9, +23% at L18, +33% at L27), consistent with deeper layers having more norm variation driving the between-quartile effect. The magnitude-bypass 69% rate at 500M (§4.3) does not replicate at 5M tokens (27–53% across four restoration-on variants). [exp44] [exp46] [exp19]

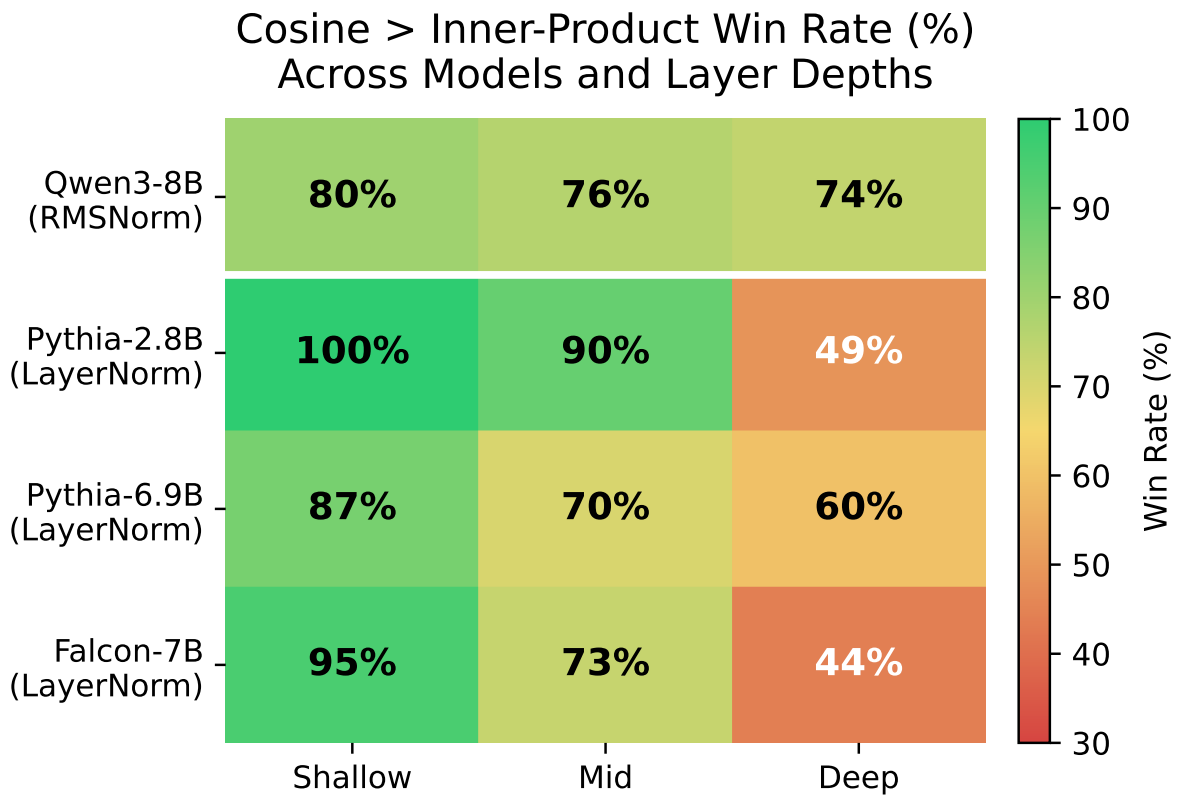

Figure 15. Cosine-vs.-inner-product win rate by depth. Deep RMSNorm: 70–90%. Deep LayerNorm: at chance. [exp25]

C.1. SAE-Free Direction vs. Norm Patching

The cos>inner diagnostic in §C relies on SAE decoder directions. This subsection removes that dependency entirely: for 500 random cross-prompt token pairs at each layer, we decompose the residual-stream activation into direction ($\hat{x} = x/\|x\|$) and magnitude ($\|x\|$), then measure KL-divergence from the unpatched output after swapping each component independently (Figure 17).

Scaling test. Replacing a single token's direction with a random unit vector (at fixed norm) causes $2.2\times$ (L9), $3.5\times$ (L18), and $11.6\times$ (L27) more KL than scaling its norm by 0.5–$5\times$ at fixed direction. The depth gradient is consistent with RMSNorm progressively erasing magnitude from the sublayer-input path.

D. Sparse-Probing Coverage at Matched FVE

Setup. `Qwen/Qwen3-8B-Base` layer 18, 500M FineWeb tokens, $d_{\text{sae}} = 65{,}536$, recipe in §J. Four architectures: Standard, Adaptive Cosine SAE, Per-Feature Adaptive Cosine SAE, Magnitude-Bypass SAE. Body presentation: §4.1.

Not held fixed across variants. Encoder score range (cosine $\in [-1, 1]$ vs. unbounded inner product); learned scale parameter a ($\leq 0.1\%$ extra parameters); encoder initialization (norm-aware vs. $\sqrt{d}$); activation dynamic range; wall-clock time (8–14% overhead).

Reconstruction parity. FVE $\in [0.767, 0.771]$; KL-div score $\in [0.984, 0.985]$; CE-loss score $\in [0.991, 0.993]$; dead features $\leq 4.3\%$.

Sparse probing. Top-1 differences from Standard (mean over three SAE-training seeds): $+14.1\%$ (Adaptive Cosine SAE) and $+14.6\%$ (Per-Feature Adaptive Cosine SAE); Magnitude-Bypass SAE is $+11.6\%$ (single seed). Per-dataset breakdown: Table 9. The dataset-removal robustness table below is computed on the single representative seed ($+13.3\%$ / $+14.9\%$):

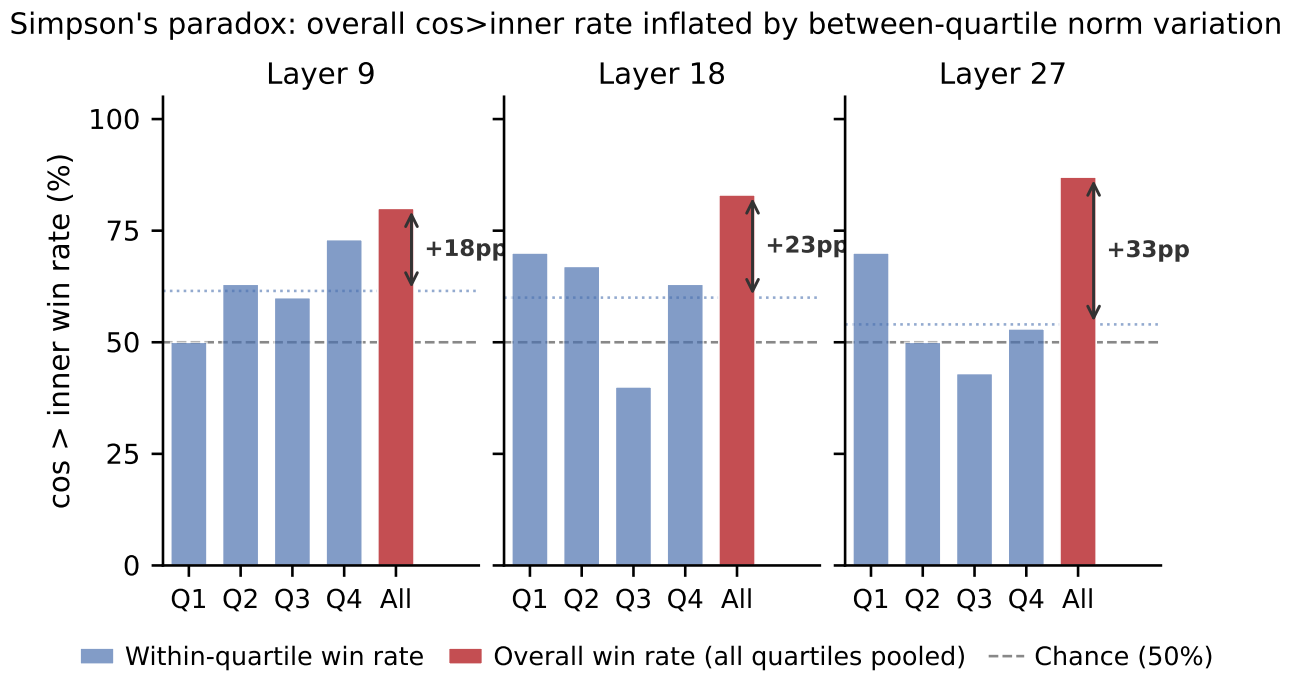

Figure 16. **Simpson's paradox in the cos>inner diagnostic.** Within individual norm quartiles (blue), the win rate hovers at 40–70%; the overall rate (red) is 80–87%. The gap grows with depth as residual-stream norm variation increases. This confirms that the between-quartile component (norm variation corrupting cross-token TopK selection) drives most of the overall cos>inner rate. Standard SAE, Qwen3-8B, 50M tokens. [exp19]

Table 7. Core SAEBench metrics, Qwen3-8B L18, 500M tokens, $d_{\text{sae}} = 65{,}536$. [exp40/42c/44]

	Std.	Adapt. Cos.	PF Adapt. Cos.	Mag.-Byp.
FVE	0.770	0.769	**0.771**	0.767
KL-div score	0.985	0.985	0.985	0.984
CE-loss score	0.993	0.993	0.993	0.991
Dead %	0.0	0.0	0.0	4.3
Probing top-1	0.667	0.800	**0.815**	0.783
Probing top-2	0.731	**0.853**	0.853	0.798
Probing top-5	0.789	**0.891**	0.883	0.805
Overall	0.944	0.957	**0.959**	0.914

Decoder direction overlap. Mutual nearest-neighbor matching: 73–78%. Strict $>$ 0.95-cosine + Jaccard: 6–17%. Restricted to alive-and-paired subset at $n = 3$–6: $< 1\%$ (§H). Headline comparisons are not paired feature-to-feature tests. [exp56b]

Per-feature interpretability. Matched across architectures at the 500M headline (describe-then-predict, 1,000 features/arm): 19.2% (Per-Feature Adaptive Cosine SAE), 21.3% (Adaptive Cosine SAE), 20.1% (Standard) (§F). Without the auxiliary loss, the alive-feature gap reopens and total interpretable features differ by $\approx 4.5\times$ at matched per-feature rate (§F.1).

Compute overhead. Training wall-clock vs. Standard: $+8\%$ (Adaptive Cosine SAE), $+10\%$ (Per-Feature Adaptive Cosine SAE), $+14\%$ (Magnitude-Bypass SAE). Inference: one extra $\log \|x_c\|$ + exp per token before BatchTopK (Adaptive Cosine SAE, Per-Feature Adaptive Cosine SAE); post-decode norm-and-multiply for Magnitude-Bypass SAE.

D.1. Per-Dataset Sparse Probing

Table 9 reports the single-feature top-1 by dataset for all four architectures. Per-Feature Adaptive Cosine SAE has higher or equal top-1 to Standard on seven of eight datasets; amazon_sentiment is the one dataset on which Standard is higher.

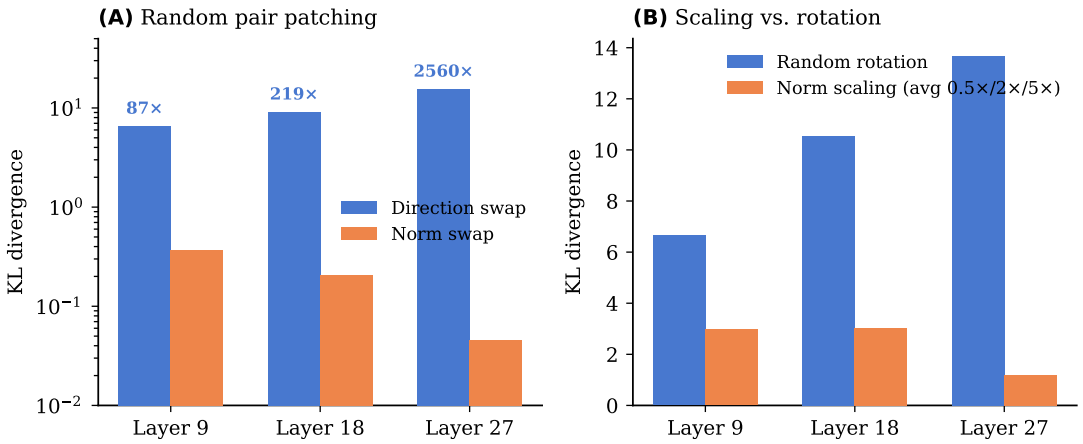

Figure 17. **The model reads direction, not magnitude.** KL-divergence caused by swapping direction (blue) vs. norm (orange) between random token pairs at three Qwen3-8B layers. Direction patches cause 87–2,560× more disruption; the ratio grows with depth as fewer subsequent layers remain for magnitude to influence direction. $n = 500$ pairs per layer. [exp8]

Table 8. Top-1 vs. Standard, with high-margin datasets removed. [exp40]

Removed	Adapt. Cos.	PF Adapt. Cos.	Mag.-Byp.
none	+13.3%	+14.9%	+11.6%
europarl	+10.3%	+12.0%	+8.6%
europarl + github-code	+9.0%	+9.1%	+5.6%

D.2. Per-Architecture Sparse-Probing Bars

Figure 18 plots the per-dataset top-1 numbers from Table 9 alongside aggregate top-1/2/5 across the eight datasets, comparing Standard against Per-Feature Adaptive Cosine SAE; this is the provenance figure referenced from Fig. 2 and Table 1.

E. Sample Efficiency and Cross-Model Generalization

Without auxiliary loss. The headline of §4.1 keeps the auxiliary loss enabled in both arms. With the auxiliary loss removed, the alive-feature differences become visible: on Gemma-2-2B at 50M tokens, Standard shows 54–69% dead vs. $\approx 0\%$ for Adaptive Cosine SAE (Table 11). At 500M on Qwen3-8B with $\sqrt{d}$ init, $d_{\text{sae}} = 16{,}384$, no auxiliary loss, the global-a Adaptive Cosine SAE shows $+6.3\%$ FVE and $2.39\times$ alive features over Standard at L27 (Table 10). Both architectures plateau by 300M tokens.

Cross-model coverage. RMSNorm: Qwen3-8B, Gemma-2-2B, Mistral-7B. LayerNorm: Pythia-2.8B, Pythia-6.9B, Falcon-7B, Pythia-70M. Token budgets vary (500M for the Qwen L18 run, 50M for most cross-model rows, 5M for some); rows in Table 12 are not normalized for budget or recipe.

Variance. $n = 3$ seeds at Qwen L27, 50M tokens: FVE 0.737 (Adaptive Cosine SAE) vs. 0.657 (Standard); seed-wise SD < 0.001 in both arms; alive-feature ratio $3.26\times$. [exp34] At the 500M L18 headline, three SAE-training seeds give a stable top-1 gap of $+14.1\%$ (global) and $+14.6\%$ (per-feature), FVE matched to ± 0.0002 (Appendix K). [exp61] Cos $>$ inner above 50% at L27: $p < 2.4 \times 10^{-6}$. Per-feature interpretability rate difference (50M/L27): $p = 0.88$; matched also at the 500M headline (§F.2). [exp33] [exp62]

E.1. Depth Dependence Within Qwen3-8B

Figure 20 shows the four main variants at L9, L18, and L27 (all 50M tokens, same recipe). The key observations:

1. *Architectures converge at shallow layers.* At L9 ($\|x\| \approx 58 \approx \sqrt{d}$), all four variants achieve 0% dead features and FVE within 0.6%. The norm/sqrt(d) ratio is near unity, so the scale mismatch that drives divergence at deeper layers does not arise.

Table 9. Single-feature top-1 by dataset. [exp40]

Dataset	Std.	Adapt. Cos.	PF Adapt. Cos.	Mag.-Byp.
europarl (languages)	0.644	0.986	**0.992**	0.978
github-code (programming)	0.515	0.700	**0.813**	0.784
bias_in_bios set 1	0.669	0.798	**0.862**	0.786
bias_in_bios set 2	0.559	0.762	0.758	**0.766**
bias_in_bios set 3	0.612	**0.778**	0.766	0.763
amazon_reviews categories	0.712	0.713	**0.715**	0.675
amazon_sentiment	0.915	**0.924**	0.880	0.795
ag_news (topics)	0.705	**0.736**	0.735	0.720

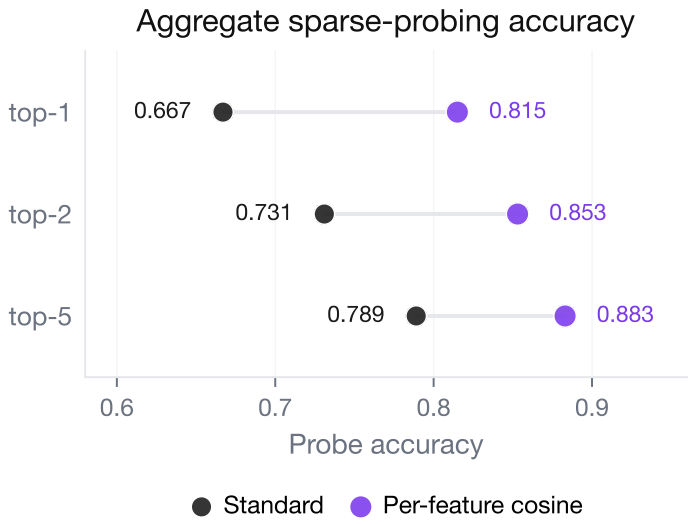

Figure 18. Per-feature top-1 by dataset (Standard vs. Per-Feature Adaptive Cosine SAE) and aggregate top-k. [exp40]

2. *Cos>inner is below* 50% *at L9.* Inner product is a better causal predictor at L9 (33–44%), consistent with magnitude carrying genuine signal at shallow layers where RMSNorm has not yet accumulated. The diagnostic crosses 50% between L9 and L18, and reaches 67–78% at L27.
3. *Per-feature (no anchor) collapses only at L27.* At L18 ($\|x\|/\sqrt{d} = 1.53$), the unconstrained per-feature variant survives with 0% dead and the best FVE (0.726). At L27 ($\|x\|/\sqrt{d} = 6.3$), it collapses to 83.4% dead. The cascade (§B.2.2) is gated by the norm-to-init ratio, not by absolute depth.
4. *Learned a tracks the magnitude-as-noise gradient.* Global a rises from 0.025 (L9) to 0.257 (L27); at L9 the optimizer wants pure cosine, at L27 it retains modest norm sensitivity for reconstruction. No value exceeds 0.3; the inner-product regime ($a = 1$) is never approached.

The depth story connects the mechanism (§4.3) to the architecture (§B.2): RMSNorm progressively erases magnitude from the residual stream at deeper layers, making inner-product scoring progressively worse as a feature detector. The cosine encoder's advantage grows with depth because the "noise" it removes (magnitude) becomes a larger fraction of the total signal at deeper layers. This explains why the headline result (L18, 500M) is a moderate case; deeper layers show an even larger divergence between cosine and inner-product dictionaries, but at the cost of requiring more careful parameterization (base+delta rather than free per-feature).

E.2. Cross-Budget Sparse Probing

Comparing our 50M-token cosine SAEs against the independently trained reference SAE (Karvonen et al., 2025) at 500M tokens (same model, same BatchTopK recipe, different codebase):

Table 10. Reference runs at 500M Qwen3-8B. Aux: with the auxiliary loss; No-aux: without. [exp40] [exp42c]

Run	Layers	d_{sae}	Recipe	Cosine vs. Standard
A	L9, L18, L27	16k	No-aux	init-contaminated at L18/L27
B	L18	65k	Aux	Standard reference, 0% dead
C	L9, L18, L27	16k	No-aux, $\sqrt{d}$ init	+6.3% FVE, 2.39× alive at L27
D	L18	65k	Aux	FVE parity

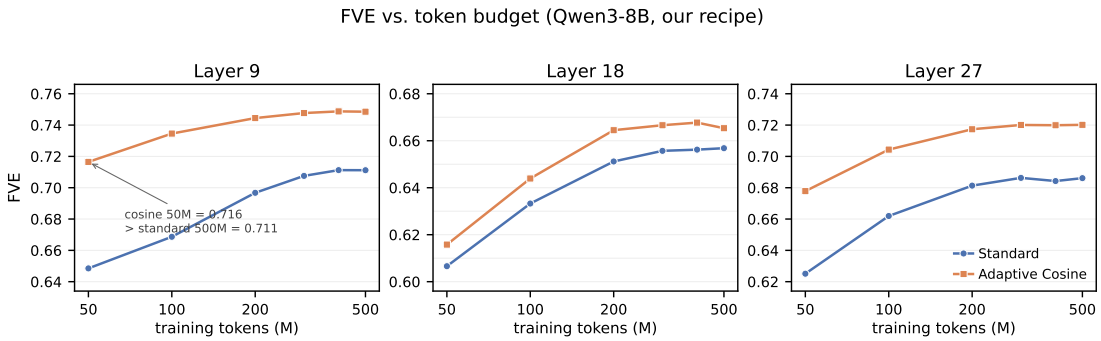

Figure 19. FVE vs. token budget on Qwen3-8B (500M, our recipe). Adaptive Cosine SAE at the 50M checkpoint reaches Standard's 500M FVE at L9. [exp42c]

The reference SAE matches our own standard baseline within noise at L18 (both ≈ 0.679 at 500M), confirming it is representative. The cross-budget gap is largest at L9, where the cosine encoder's direction-only scoring is most beneficial relative to the shallow-layer norm distribution.

E.3. Sparsity-Budget Sweep

We vary the BatchTopK sparsity budget k (active features per token) at 50M tokens, L18, $d_{\text{sae}} = 65{,}536$, training Standard and Per-Feature Adaptive Cosine SAEs at $k \in \{10, 20, 40, 80, 160\}$ (the headline uses $k = 80$). [exp66] Two findings (Table 14). First, the sparse-probing advantage is robust across the full budget range: the per-feature top-1 gain over Standard is $+8$ to $+11\%$ at every k, with no collapse at low k (it is in fact largest, $+11.4\%$, at $k = 40$). Second, per-feature interpretability stays matched at every k: the describe-then-predict rate (§F.2 protocol, 200 features per arm) tracks between the two architectures within sampling noise, the arms trading the lead. Both architectures' interpretability rates decline as k grows, since more simultaneously-active features are individually harder to characterize. The sparsity sweep thus reinforces the central decomposition: the cosine advantage is in feature *discovery* (probing), not in per-feature interpretability, and this holds across sparsity budgets.

E.4. Cross-Model Summary

On Gemma-2-2B at 50M, $d_{\text{sae}} = 9{,}216$: fixed-SAE top-1 sparse-probing difference $+3.4 \pm 0.3\%$. [exp57c] On Pythia-2.8B, Pythia-6.9B, Falcon-7B, the cosine top-1 advantage at deep layers does not hold; cos > inner drops from $\sim 100\%$ at shallow to 40% at deep (Pythia-2.8B L24 sharpest). [exp25]

Scaling with model size and expansion ratio. A 3×3 matrix (Qwen3-1.7B/4B/8B $\times$ 4×/8×/16× expansion, 50M tokens, same recipe, three SAE-training seeds per cell) isolates the two potential drivers (Figure 9). Model size is the primary factor: the row mean rises from $+5.3\%$ at $d_{\text{model}} = 2048$ to $+11.7\%$ at $d_{\text{model}} = 2560$, then holds ($+9.2\%$ at $d_{\text{model}} = 4096$); the gain is large at every size but not strictly monotonic in d_{model}. Expansion ratio has no systematic effect (column means $+9.4$/$+8.9$/$+7.9\%$, overlapping within seed noise). Per-cell seed SD is small ($\approx$ 1–3%). This revises the earlier interpretation that dictionary size drives the gap; the Gemma-2-2B result ($+3.4\%$ at $d_{\text{model}} = 2304$, 4× expansion) is explained by its small model dimension, not its small dictionary. [exp67]

Table 11. Without auxiliary loss, Gemma-2-2B, 50M tokens. [exp35]

Layer	Variant	FVE	Dead %	Alive	Gini
L7	Standard	0.759	54.2	4,176	0.890
L7	Adapt. Cos.	**0.798**	0.1	**9,098**	0.662
L13	Standard	0.696	68.8	2,802	0.906
L13	Adapt. Cos.	**0.747**	0.0	**9,024**	0.648
L19	Standard	0.788	60.2	3,616	0.875
L19	Adapt. Cos.	**0.855**	0.0	**9,161**	0.651

Table 12. Cross-model behavior. *cos > inner*: per-feature win rate on the diagnostic of §C (50% null); "$a \to b$" on LayerNorm rows is shallow→deep. *FVE* Δ: cosine − standard, in absolute FVE units. *Alive* $\times$: alive-feature ratio of cosine over standard. [exp25] [exp35]

Model	Norm	cos > inner	FVE Δ	Alive $\times$
Qwen3-8B	RMS	62–89%	+0.6 to +8.0	1.7–3.3
Gemma-2-2B	RMS	52–90%	+2.5 to +6.7	1.4–3.2
Mistral-7B	RMS	74–88%	+0.6 to +3.0	varies
Pythia-2.8B	LN	100→40	+0.8 to +5.0	2.3 (L24)
Pythia-6.9B	LN	83–90→50–70	+1.8 to +3.3	varies
Falcon-7B	LN	90–100→40–47	+5.0 (L24)	4.5 (L8)

E.5. FVE and Dead-Feature Trends Without Auxiliary Loss

By-layer view of the no-auxiliary-loss runs cited in §4.1 (Qwen3-8B, 50M tokens, $d_{\text{sae}} = 16{,}384$). FVE (Fig. 21, left) places Adaptive Cosine SAE and Per-Feature Adaptive Cosine SAE above Standard at L9 and L27, with the gap narrowing at L18. Dead-feature rate (right) separates the cosine variants at L18: Per-Feature stays below Standard, while the single-global-a Adaptive variant matches Standard there and only recovers at L27. With the auxiliary loss restored, the dead-feature gap collapses to 1.9% at L18 (500M tokens).

F. Feature Interpretability

Setup. We score features with an LLM judge under a describe-then-predict protocol: the judge writes a description from a feature's top activating contexts, then predicts which held-out tokens it fires on; "interpretable" means a feature is human-recognizable in the sense that its activations are predictable from a short natural-language description ($\geq 50\%$ held-out prediction accuracy). At the 500M headline (Qwen3-8B L18, 1,000 stratified features per architecture, Sonnet judge), per-feature interpretability is matched across architectures (Table 16, §F.2): 20.1% (Standard), 21.3% (global a), 19.2% (per-feature). The same is true at 50M / L27 under an alternative 1–5 rubric (40.0% vs. 37.0%, $p = 0.88$). [exp62] [exp33] The cosine advantage is therefore in feature *discovery* (§G), not in the legibility of individual features.

Decoder direction overlap. > 0.7 cosine match: 31% of 1000 sampled features. Strict-identity overlap is 6–17% (§4.1). [exp56b]

F.1. Total Interpretable Features Without the Auxiliary Loss

Per-feature rates being matched, the count of interpretable features tracks how many features stay alive. With the auxiliary dead-feature loss (the deployed recipe), both architectures keep $\geq$ 50k alive. Without it, the alive-feature gap reopens: at 100M tokens, scoring 1,000 stratified features on a 1–5 rubric (≥ 4 interpretable), the norm-preserving cosine variant keeps far more features alive and therefore yields $\approx 4.5\times$ more interpretable features in total, at a matched per-feature rate (Table 15). This is the no-auxiliary-loss stress test of §4.4, not the headline recipe, and the arm shown is the norm-preserving (Magnitude-Bypass) variant.

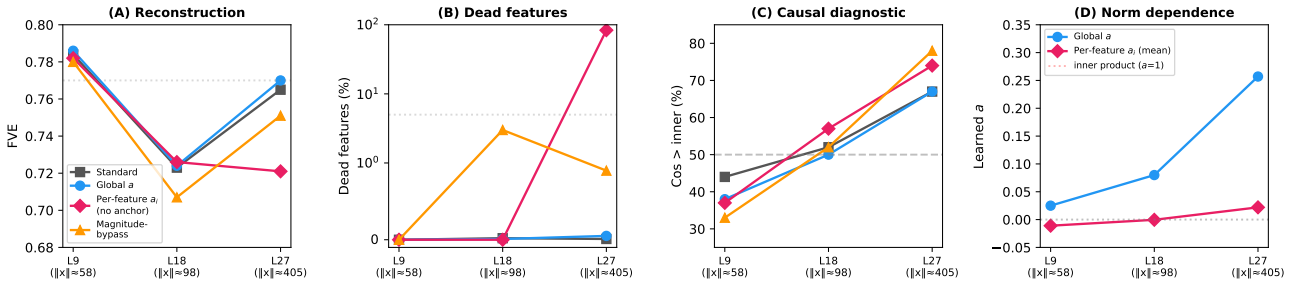

Figure 20. **Architecture behavior across depth.** Qwen3-8B, 50M tokens, all four main variants. **(A)** FVE converges at L9 and diverges at L27. **(B)** Per-feature (no anchor) collapses catastrophically at L27; magnitude-bypass shows mild dead features at L18. **(C)** Cos>inner crosses 50% between L9 and L18, reaching 78% for magnitude-bypass at L27. **(D)** The optimizer drives a toward zero at shallow layers and toward ~ 0.26 at deep layers, never approaching $a = 1$. [exp43c/43d/43]

Table 13. Sparse-probing top-1 across token budgets. The cosine SAE at $10\times$ fewer tokens surpasses the standard reference at L9 and L27. [exp51]

Layer	Cosine 50M	Standard 500M	Δ
L9	**0.786**	0.613	+17.3%
L27	**0.849**	0.762	+8.7%

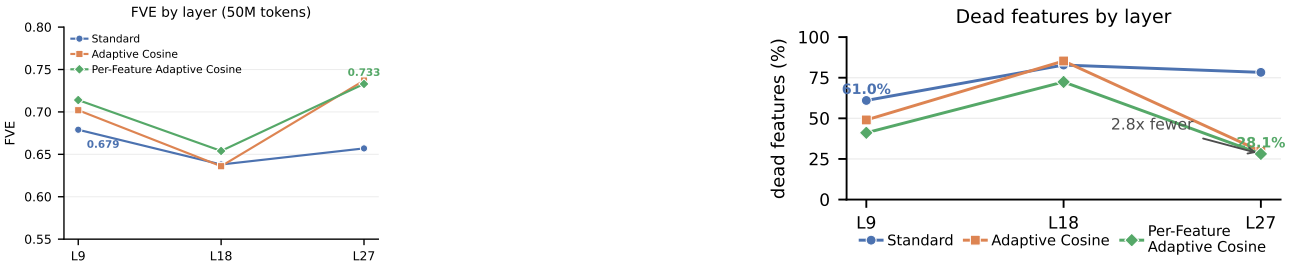

Figure 21. Without auxiliary loss, FVE and dead-feature rate by layer (Qwen3-8B, 50M tokens). At 500M tokens the L18 dead-feature gap (Per-Feature Adaptive Cosine SAE vs. Standard) is 1.9%. [exp17] [exp42c]

Table 15. LLM-judged interpretability without the auxiliary loss (100M tokens, 1,000 stratified features, 1–5 rubric, ≥ 4 interpretable). Per-feature rates matched; the $\approx 4.5\times$ total-count gap comes from more features staying alive, not higher per-feature quality. [exp40]

Variant (100M, no aux)	Per-feature ≥ 4	Alive	Total ≥ 4
Standard	82.1%	3,529	$\approx$ 2,900
Magnitude-Bypass (norm-preserving)	80.1%	16,332	$\approx$ **13,100**

F.2. LLM Interpretability at 500M Tokens

At 500M tokens with the production recipe (matched alive counts via aux-k), LLM-judged describe-then-predict rates are matched across architectures: 1,000 stratified features per arm, scored by a Sonnet judge, fall within a 2.1-point band (Table 16). Per-feature cosine is statistically indistinguishable from standard (19.2% vs. 20.1%), with global cosine marginally highest (21.3%). Alive counts are matched (17.3–18.7k), so this is a like-for-like per-feature comparison.

This is the expected reading given the rest of the analysis: per-feature interpretability, like per-feature steering power (§G), is matched across architectures, and the cosine advantage lives in which features the dictionary discovers rather than in the legibility of any individual feature. A smaller 200-feature pilot (Karvonen et al. 2025-style stratification) suggested a frequency crossover, with per-feature cosine ahead on low-frequency features; that pattern does not survive at 1,000 features (low-frequency rates fall within 2 points across arms), so we attribute it to small-sample noise and do not claim a frequency-dependent interpretability effect.

Table 14. Sparsity-budget sweep (Qwen3-8B L18, 50M tokens, $d_{\text{sae}} = 65{,}536$). *Top-1*: SAEBench single-feature probing. *Interp*: describe-then-predict interpretable rate (200 features/arm). Std = Standard, PF = Per-Feature Adaptive Cosine. [exp66]

k	10	20	40	80	160
Top-1 Std	0.653	0.695	0.678	0.731	0.693
Top-1 PF	**0.733**	**0.782**	**0.792**	**0.816**	**0.788**
gap	+8.0	+8.7	+11.4	+8.5	+9.5
Interp Std	0.530	0.395	0.350	0.350	0.180
Interp PF	0.495	0.440	0.325	0.300	0.215

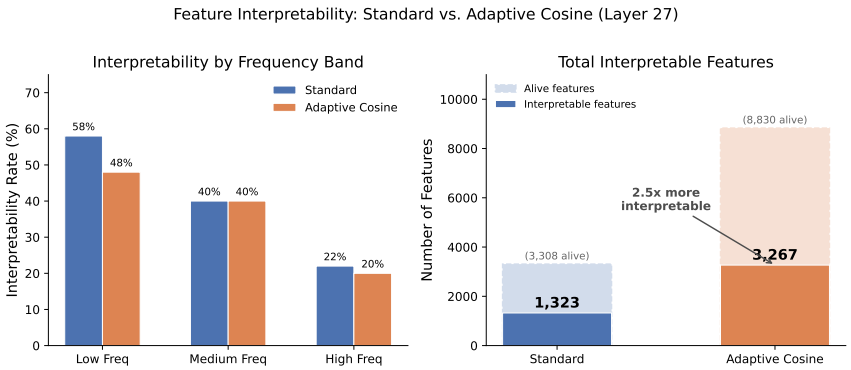

Figure 22. Per-feature interpretability rates are matched across architectures; when the alive-feature gap is open (no auxiliary loss), the cosine encoder nonetheless yields more interpretable features in total because more features are alive. [exp62]

F.3. Concrete Feature Examples

Table 17 makes "interpretable" concrete: for each architecture we show one feature the judge could describe and then use to predict held-out firing (accuracy 1.0), and one it could not (≤ 0.1). The interpretable features are crisp lexical or syntactic detectors under every score (e.g. the word "you," the adverbial "-ly" suffix, "the" following "and"); the uninterpretable ones fire on heterogeneous token sets with no description that predicts them (e.g. subword fragments completing a split name). Both dictionaries contain features of each kind, which is why per-feature rates are matched (Table 16); consistent with the salient-concept control (§L), the cosine advantage is distributional, not a matter of individual cosine features being more legible.

G. Mechanism: Gradient Equalization and Q4 Reconstruction

This section uses the three cosine variants from §3: *global* a (single learned exponent), *per-feature* ($a_i = a_{\text{base}} + \delta_i$), and *magnitude-bypass* (pinned a=0 with norm restoration; Appendix B).

Encoder-gradient asymmetry. For an active feature under a fixed BatchTopK/ReLU mask, the inner-product score gradient is $\partial s_i / \partial w_i = x_c$, so the encoder update inherits a factor of $\|x_c\|$. Cosine scoring replaces this with a normalized direction times a scalar $\|x_c\|^a$; the trained $a \ll 1$ (Table 4).

Equalization signature. Cosine scoring compresses the gradient ratio toward unity. The median per-feature Q4/Q1 ratio drops from $1.55\times$ (Standard) to $1.03\times$ (per-feature cosine). Only 13.5% of cosine features exceed Q4/Q1 > 2, versus 35.3% under standard scoring. Q1-specialized features increase from 18 to 562 (Fig. 23). [exp28]

Recipe and dictionary-size context. On Gemma-2-2B ($d_{\text{sae}} = 9{,}216$, $4\times$ expansion, 50M tokens), the community SAEBench recipe's auxiliary loss eliminates dead features for both architectures, narrowing the cosine advantage to $+3.4\%$. The scaling matrix (§E.4) shows that model dimension, not expansion ratio, drives the gap; the headline $+14.6\%$ emerges at Qwen3-8B's $d_{\text{model}} = 4096$. [exp57c] [exp57]

Discovery vs. separability. How much of the probing gap comes from features the cosine encoder discovers versus cleaner encoding of shared features? We restrict both 500M dictionaries to 8,661 mutually matched features (activation correlation ≥ 0.7 and decoder cosine > 0.7). On matched features alone, top-1 difference shrinks from $+14.87\%$ to $+1.95\%$; top-5

Table 16. LLM-judged interpretability at 500M tokens, 1,000 stratified features per architecture, Sonnet judge. Interpretable iff $\geq 50\%$ prediction accuracy. Per-feature rates are matched; the cosine advantage is in feature discovery (§G), not per-feature interpretability. [exp62]

Variant	Rate	Low-freq	Med-freq	High-freq
Standard	20.1%	22.3%	21.6%	14.8%
Global a	**21.3**%	24.3%	21.4%	18.0%
Per-feature	19.2%	23.5%	19.5%	14.3%

Table 17. Representative interpretable (prediction accuracy 1.0) and uninterpretable (≤ 0.1) features per architecture at the 500M headline, with the LLM-written description and top activating tokens. Both architectures produce clean and fragmentary features alike. [exp62]

Architecture	Feat.	Auto-interp description (abbrev.)	Acc.
Standard	37058	word "you" (subject pronoun)	1.0
Standard	16263	head noun following a modifier (no predictive pattern)	0.1
Global a	52155	adverbial "-ly" suffix	1.0
Global a	37895	mixed post-verb temporal tokens (no predictive pattern)	0.0
Per-feature	19859	"the" immediately after "and"	1.0
Per-feature	33290	subword fragment completing a split first name	0.1

from $+10.56\%$ to $+2.70\%$. The standard SAE's unmatched features contribute nothing beyond its matched set (full $-$ matched at top-5: $0.7831 - 0.7837$). The cosine SAE's unmatched features add $+7.8\%$ ($0.8888 - 0.8108$). [exp58b]

The advantage is therefore distributional, not a failure of standard scoring to represent salient concepts. For hand-picked content concepts (a programming language, a natural language), *both* dictionaries contain a cleanly selective feature: the top concept-aligned feature fires strongly on-concept and is silent off-concept under either score (Appendix L). The gap lives in the aggregate over the eight probing categories and the long tail of features, consistent with the matched-feature decomposition above, rather than in any single obvious concept. [exp65]

Unmatched features by norm quartile. Where do each model's unique features fire? Among strict-unmatched features at L18, 86.3% of Standard activations fall in Q4 vs. 56.9% for per-feature cosine. The mean norm of Standard-unique activations is 9,689 ($48\times$ the sampled token mean). On tokens where per-feature cosine-unique features fire, Standard fires 328 features/token vs. 122 for per-feature cosine. [exp58c]

Decoder geometry control. For ten cached TPP probe directions, $w_{\text{probe}} W_{\text{dec}}^{\top}$ has matched concentration: entropy ratio 1.00001, effective-dimension ratio 1.00007. [exp58a]

Feature steering behavioral equivalence. To verify that the sparse-probing gap reflects feature *discovery*, not per-feature *power*, we directly steer individual features at amplification factors $0.5\times$, $2\times$, and $5\times$ and measure downstream KL-divergence. Across 11 top-activating standard features and 9 cosine features (50 prompts each), mean KL ratios are $1.04\times$ ($0.5\times$), $1.00\times$ ($2\times$), and $1.00\times$ ($5\times$): cosine and standard features produce indistinguishable behavioral effects at matched amplification. This confirms that the $+14.9\%$ probing gap is not explained by standard features being individually "stronger"; rather, cosine discovers more concept-aligned directions. [exp55d]

Causal ablation cleanliness. Sparse probing is a proxy; we therefore test the features causally. Ablating the top-N probe-selected features and decomposing the logit change into *intended* (target-class) and *unintended* (collateral) effects, the cosine dictionary ablates far more cleanly. The precision ratio (intended/unintended effect) rises with N for cosine, from $8.4\times$ at $N{=}2$ to $21.8\times$ at $N{=}500$, while standard *degrades* from $5.6\times$ to $1.4\times$ (Table 18). The gain is driven by the denominator, not the numerator: at $N{=}500$ the intended effect is comparable across architectures (0.44 Standard vs. 0.37 cosine), but collateral damage stays nearly flat for cosine ($0.0007 \rightarrow 0.017$) while growing two orders of magnitude for standard ($0.0028 \rightarrow 0.30$). The cleanliness is therefore a property of *which* features the dictionary holds, not of any per-feature separability advantage: it is the norm-conditioned features (§G), absent from the cosine dictionary, that bleed diffuse side effects when ablated as a set. This is consistent with the matched single-feature steering above and with the small ($+2.0\%$) separability component of the probing gap; the cosine dictionary spends capacity on content rather than norm, now established through intervention rather than a linear proxy. [exp57b]

Q4 reconstruction. Q1–Q3 FVE is matched; Q4 splits: Standard -183.5, global cosine 0.33, per-feature cosine 0.25, magnitude-bypass -0.20. Reconstruction-norm ratio (output-norm / input-norm) on Q4: Standard $9.5\times$; cosine variants

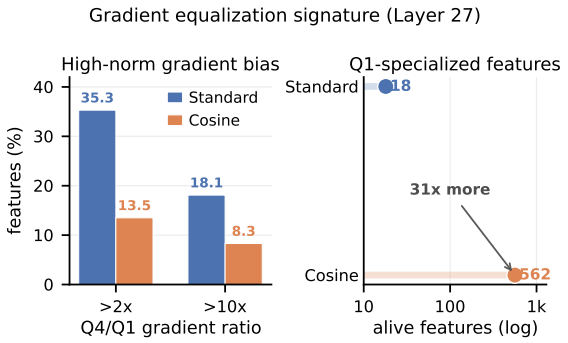

Figure 23. Encoder-gradient ratio Q4 (high-norm) / Q1 (low-norm). Standard: 35.3% of features have Q4/Q1 > 2. Per-feature cosine: 13.5%. [exp28]

Table 18. **Ablation precision (intended/unintended logit effect) vs. number of ablated features.** Numeric form of Fig. 7. Higher is cleaner. Cosine sharpens with N; standard collapses toward $1\times$ as collateral (not weaker intended effect) overwhelms it. Intended effects are comparable across architectures; the gap is in collateral. Qwen3-8B L18, 500M. [exp57b]

N ablated	2	10	50	100	500
Standard	$5.6\times$	$5.9\times$	$4.7\times$	$2.5\times$	$1.4\times$
Cosine	$\mathbf{8.4}\times$	$\mathbf{9.5}\times$	$\mathbf{14.1}\times$	$\mathbf{13.9}\times$	$\mathbf{21.8}\times$

$\approx 1\times$. [exp55c]

Controls. Can cosine scoring be applied post-hoc to a trained Standard SAE? No: FVE drops -18 to -33%, L_0 jumps from 80 to 500, and $< 11\%$ of features overlap with from-scratch cosine (§G.2). [exp29] [exp17] Input normalization without unit-norm encoder rows kills 33% of alive features. [exp45] Global cosine has higher FVE than Standard at every learning rate tested. [exp30]

Dictionary size control. Giving the standard encoder $3\times$ more dictionary slots at matched L0 makes dead rates *worse*, not better: 89.4% dead (49k, k=80) vs. 77.4% (16k, k=80), while a 16k global cosine achieves 28.2% dead (Figure 25). The gradient concentration is a property of the scoring function, not dictionary capacity. With $3\times$ dictionary *and* $3\times$ L0 (k=240, $3\times$ parameters, $3\times$ compute), the standard SAE can slightly exceed the cosine SAE's FVE (0.751 vs. 0.737), confirming that cosine is $\sim 3\times$ more parameter-efficient. [exp26]

Cos $>$ inner across architectures. Which variant actually shows a cos>inner metric difference? At L18 (community recipe), only magnitude-bypass reaches 69%. global cosine ($a = 0.258$), per-feature cosine (mean $a_i = 0.076$), and Standard all cluster at 62–63%. Only the architecture that removes magnitude entirely from z separates on this metric. [exp44]

Architectures and the cos-vs-inner difference. All cosine variants improve sparse probing: magnitude-bypass $+11.6\%$ top-1, global cosine $+13.3\%$, per-feature cosine $+14.9\%$. Only magnitude-bypass shows the cos-vs-inner metric difference. The RMS-geometric-alignment mechanism is depth-dependent; the gradient-equalization signature appears for all cosine variants. At L9, cos $>$ inner falls below 50%, yet both cosine variants still improve sparse probing. [exp40/42c/44]

G.1. Norm-Stratified FVE

Per-quartile values are reported once in Table 19 above. Reconstruction-norm ratio on Q4 tokens: Standard $9.5\times$; cosine variants $\approx 1.0\times$. The reference SAE of Karvonen et al. (2025) reproduces the pattern with Q4 FVE $= -136$. [exp55c]

G.2. Post-Hoc Normalization and Score Swapping

Cosine scoring applied at inference time to a trained Standard SAE: FVE -18 to -33%, L_0 from 80 to 500, Jaccard with from-scratch global cosine 10.6%. [exp29]

Direct score swap (both directions). We train a Standard and an adaptive-cosine BatchTopK SAE at the headline setting (Qwen3-8B L18, 50M tokens, $d_{\text{sae}} = 65{,}536$, $k = 80$) and swap the score geometry at inference: the same weights are

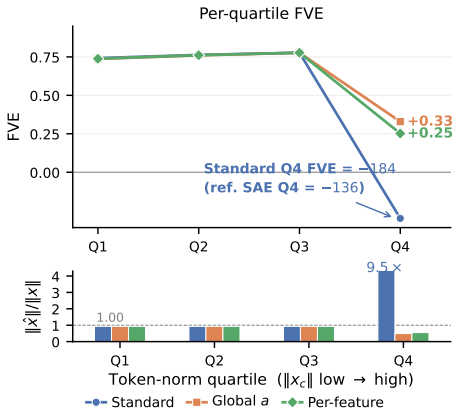

Figure 24. Per-quartile FVE (top) and reconstruction-norm ratio $\|\hat{x}\|/\|x\|$ (bottom). [exp55c]

Table 19. FVE by activation-norm quartile (Q1 lowest, Q4 highest). [exp55c]

	Std.	Global a	Per-feat.	Mag.-byp.
Q1	0.738	0.732	0.733	0.738
Q2	0.763	0.760	0.762	0.762
Q3	0.778	0.777	0.778	0.775
Q4	-183.5	0.329	0.252	-0.200

read with the other encoder (the adaptive-cosine score recovers inner product at $a = 1$). This tests the mechanism more directly than gradient reweighting (§G.1): does the score, holding weights fixed, move the result? Table 20 separates two effects. [exp64]

Table 20. **Score swap at inference: FVE is training-bound, probing is read-time.** Same checkpoint, scored two ways. Swapping the score destroys FVE in *both* directions (reconstruction is fixed at training time), but sparse-probing top-1 tracks the *score used at read time*, not the score the dictionary was trained with. [exp64]

Checkpoint	Scored as	FVE	Probe top-1
Standard	inner (native)	0.709	0.694
Standard	cosine (swap)	0.599	**0.763**
Cosine	cosine (native)	0.707	0.766
Cosine	inner (swap)	0.612	**0.698**

FVE falls by ≈ 0.11 under either swap: a dictionary trained for one geometry reconstructs poorly when read with the other, confirming reconstruction quality is set during training. Sparse probing behaves oppositely: Standard weights read with the cosine score reach 0.763 (matching the from-scratch cosine SAE's 0.766), and cosine weights read with inner product fall to 0.698 (matching from-scratch Standard's 0.694). Probing quality therefore follows whichever geometry reads the dictionary, consistent with the decomposition of §B.4: the input-norm channel that governs probing is applied at read time, whereas the encoder weights (and their FVE) are fixed.

This is not a post-hoc shortcut. Reading a Standard dictionary with the cosine score recovers the probing number only by moving off its trained operating point: FVE drops to 0.599 and L_0 drifts to 55 (Table 20), so the result is a worse autoencoder that happens to select more interpretable features, not a usable SAE. A deployable dictionary needs reconstruction *and* interpretable selection at a matched sparsity, and only end-to-end cosine training delivers both; the swap isolates *why* (the score sets selection) without substituting for training (the decoder must be fit under that score). This is the inference-time analogue of the destructive post-hoc result above and of exp29.

Continued-training swap. Swapping the score and continuing training for 10M tokens recovers FVE (to ≈ 0.71) but does not reorganize the dictionary: the alive-feature set is nearly unchanged (Jaccard ≥ 0.998 with the pre-swap dictionary in both directions) and decoder directions barely move (mean cosine 0.97–0.98). At this budget the swap re-fits magnitudes

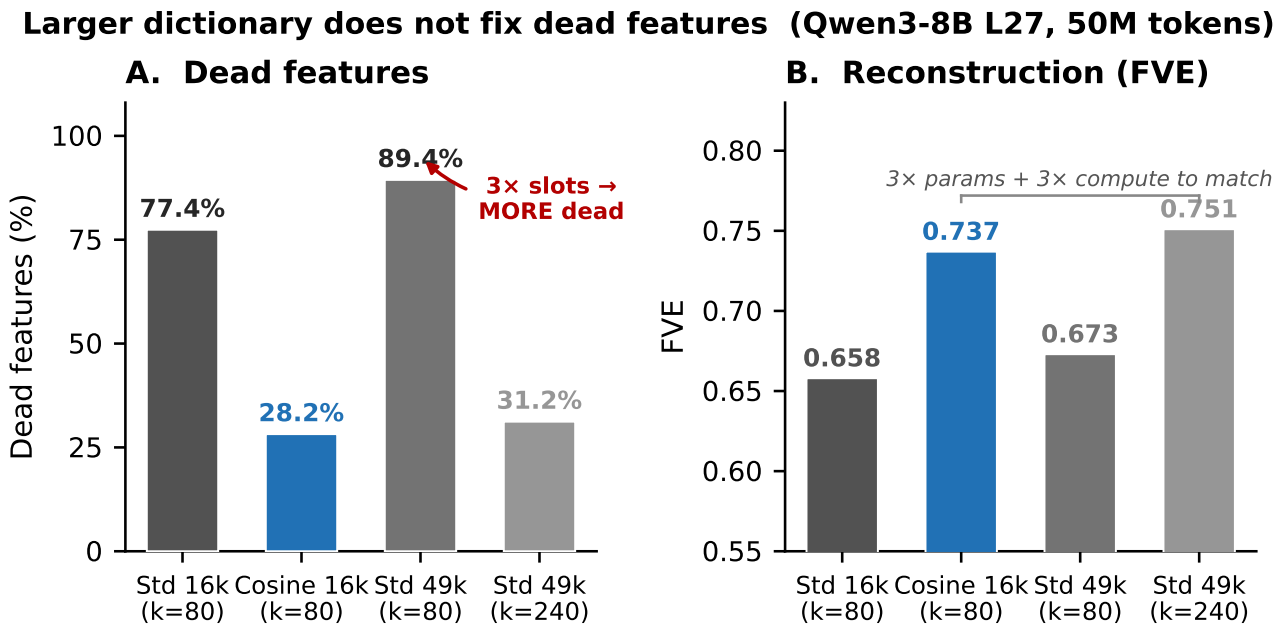

Figure 25. **Larger dictionary does not fix dead features.** Qwen3-8B L27, 50M tokens. **(A)** Tripling dictionary slots at the same L0 increases the dead rate from 77.4% to 89.4%; the cosine SAE at $1/3$ the dictionary achieves 28.2%. **(B)** Matching cosine's FVE requires $3\times$ parameters and $3\times$ L0. The dead-feature problem is a scoring-function pathology, not a capacity limitation. [exp26]

and thresholds rather than re-sorting features; whether a longer continued-training budget would induce reorganization is left open. [exp64]

G.3. Per-Feature Gradient Ratio

For each alive feature i: average $\|\nabla_{w_i}\mathcal{L}\|$ over tokens in each $\|x_c\|$ quartile, take Q4-to-Q1 ratio, report median across alive features (Fig. 26). At Qwen L9/L18/L27 with 10M tokens: median Q4/Q1 is 1.6–2.0× (Standard) vs. 0.8–1.0× (global cosine); Q1-specialized features grow $\sim 4\times$ under cosine. [exp54]

Reweighting test. If gradient asymmetry were the primary cause, equalizing it should close the gap. Reweighting the Standard reconstruction loss equalizes per-quartile gradient contributions by construction, without modifying the score. We test mild reweighting ($1/\|x_c\|$) and strong reweighting ($1/\|x_c\|^2$), spanning partial to near-complete cancellation of the encoder-gradient norm factor. At 50M tokens with FVE ≥ 0.97, per-feature cosine reaches top-1 0.648. Standard moves from 0.530 (no reweighting) to 0.532 (mild) and 0.536 (strong). Across $k \in \{1, 2, 5\}$, reweighting closes only 1.9–6.8% of the cosine difference. The forward pass still inflates every pre-activation on Q4 tokens, BatchTopK still selects the same features, and the same norm-conditioned firing pattern reappears. [exp59]

H. Limitations and Scope

Scope. All results are on the residual stream; no MLP-internal, attention-internal, head-output, or pre-norm activations. Causal diagnostics use greedy decoding and deterministic activation collection. Multi-layer / 50M+ runs cover Qwen3-8B fully; Gemma-2-2B partially; Mistral-7B after the init fix; Pythia and Falcon at 5M. No model is > 8B; no QK-norm or per-head-norm models. Training data: FineWeb (English-dominant). SAEBench is English-only (europarl is English-vs-other classification, not multilingual concept probing). No claim about multilingual residual streams, vision, audio, multimodal, or instruction-tuned / RLHF models.

Where the metric difference weakens. At Qwen L9, cos $>$ inner is below 50%. Deep LayerNorm: cos $>$ inner drops from 100% to 40% on Pythia-2.8B between L8 and L24 (inner-to-KL correlation 0.441 vs. cos-to-KL 0.271); Gemma-2-2B L20 (RMSNorm) is at $\approx 53\%$. Within-norm-quartile rates on Qwen are 40–70% (Simpson's paradox; the 70–90% rate is partly between-quartile). The architectural FVE / alive-feature differences persist across these settings. [exp25]

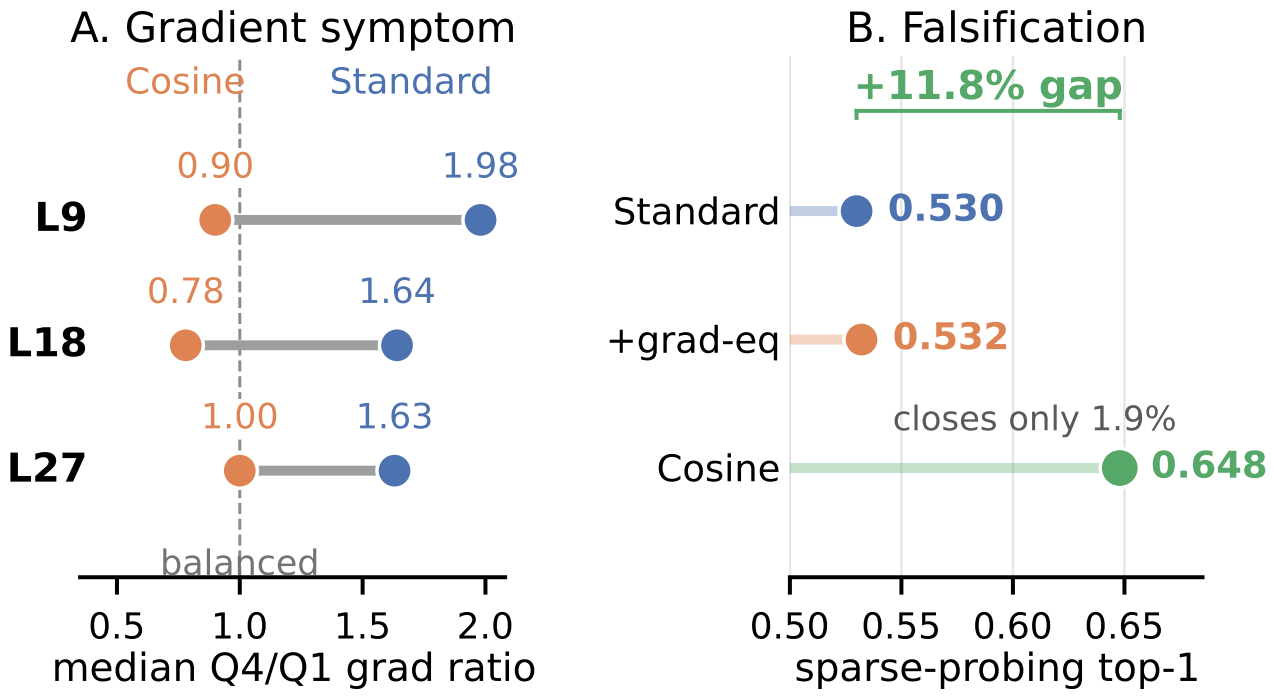

Figure 26. **Gradient asymmetry exists, but does not explain the probing gap. (A)** Median per-feature Q4/Q1 gradient ratio across Qwen layers (10M tokens). **(B)** Equalizing per-quartile gradients in the standard encoder by reweighting the reconstruction loss ($1/\|x_c\|$; 50M tokens) negligibly shifts top-1. Per-feature ratio definition and full reweighting sweep in text below.

Architecture-specific failure modes. Without auxiliary loss: Adaptive Cosine SAE retains $3.3\times$ alive-feature ratio over Standard. With auxiliary loss: both architectures reach $\sim 0\%$ dead. Per-Feature Adaptive Cosine SAE: 83% dead at 50M / L27 (65k independent scale parameters do not converge under low budget at high norms). Magnitude-Bypass SAE: 4.3% persistent dead at 500M (auxiliary-loss gradients ~ 6 orders weaker through bounded $[0, 1]$ activations; §B.2.3). At 5M the auxiliary loss is inert for Magnitude-Bypass SAE (§4.3). Adaptive Cosine SAE is the architecture without these failure modes in our coverage.

Initialization. Mistral has $\|x\| \approx 6$ vs. Qwen ≈ 400. $\sqrt{d}$ init on Mistral: $> 95\%$ dead. Norm-adaptive init helps at 5M but reduces FVE at 50M+ on Qwen. Default: $\sqrt{d}$ when $\|x\|$ is within $2\times$ of $\sqrt{d}$, else $\log(\text{mean}\,\|x\|)$ (§B.2.4). [exp42]

Evaluation scope. The 500M Qwen L18 headline reproduces across three SAE-training seeds (Appendix K); per-dataset breakdowns and the Magnitude-Bypass arm remain single-seed. Cross-layer and cross-model SAEBench replications are pending (§H.1). No feature steering, circuit-level interventions, or downstream task accuracy. Sentiment is the only top-1 reversal in Table 9 (Standard 0.915 vs. Per-Feature Adaptive Cosine SAE 0.880).

Decoder under norm shift. Under norm-noise training (Uniform$(0.5, 2.0)$), cosine FVE drops 4.5–5.8% vs. 1.8–3.4% for Standard. [exp31] The encoder is norm-invariant; the decoder reconstructs x and depends on norm. Where train and eval norms diverge, the headline difference may erode. *adaptive_l2* achieves FVE 0.738; postnorm-loss achieves SAE $\rightarrow$ KL 0.380; postnorm forces $a \rightarrow -0.4$. [exp22] [exp14]

Architectures discover different features. Strict-identity overlap of alive features between Standard and Per-Feature Adaptive Cosine SAE at L18: $< 1\%$ at $n = 3$–6 paired, consistent with the non-canonical nature of SAE dictionaries (Leask et al., 2025). Headline aggregate comparisons are across disjoint feature sets. Inner-product ablation $(x \cdot f)f$ scales with $\|x\|$, biasing the comparison toward Standard; under norm-invariant ablation, the Standard SAE $\rightarrow$ KL difference shrinks by 70% at L27. [exp56b]

H.1. Open Gaps

Coverage gaps. Whether Standard's dead features overlap the cosine-only sparse-probing features is not known. LLM-judged interpretability ran at L27 only; L9 / L18 untested. No 8B+ LayerNorm model at 50M+ tokens. Cross-architecture coverage stops at 5M for some models.

Production-scale variance. Gradient-equalization measurements are at 5M; 500M re-measurement is pending.

Architecture-design open questions. Per-Feature Adaptive Cosine SAE vs. Magnitude-Bypass SAE as the preferred magnitude-stripping architecture is unsettled. The base+δ parameterization as a fix for Per-Feature Adaptive Cosine SAE's 50M dead-feature regime has not been validated at scale. No mechanistic account exists for why Per-Feature Adaptive Cosine SAE leads at top-1 while Adaptive Cosine SAE leads at top-5.

Beyond sparse probing. OOD shift and stochastic decoding (high temperature, nucleus) are not tested.

I. Extended Discussion

Triangulation across architectures. Magnitude-Bypass SAE ($+11.6\%$ top-1, single seed), Adaptive Cosine SAE ($+14.1\%$), and Per-Feature Adaptive Cosine SAE ($+14.6\%$) all gain at matched FVE (§4.1). [exp40/42c/44] Fitted $a = 0.258$ at the headline setting; $a_i \in [-0.5, 0.5]$ across 65,536 features (Table 4). [exp42c]

Mechanism dissociation. The cos-vs-inner score advantage requires fully removing magnitude (Magnitude-Bypass SAE only); the gradient-equalization signature appears for all cosine variants (§4.3). Cos $>$ inner is below 50% at L9, yet both cosine probes still improve sparse probing at L9.

Open questions. Can magnitude be retained where informative (e.g. sentiment) without dominating feature detection? Do the dead features of the standard SAE correspond to the cosine features that drive sparse probing? Decoder-cosine overlap between cosine and standard dictionaries is 6–17% at the > 0.95 threshold (§4.1, consistent with Leask et al. (2025)); a stitching or meta-SAE analysis across cosine seeds is needed to distinguish stable-target densification from dictionary resampling. [exp56b]

J. Training Recipe

Training uses the Marks-lab `dictionary_learning` library (`https://github.com/saprmarks/dictionary_learning`) accessed via `sae-bench` (Karvonen et al., 2025). The recipe matches OpenAI's TopK SAE training (Gao et al., 2024) except for the BatchTopK selection rule (Bussmann et al., 2024) and the encoder-score swap (§B).

Table 21. Headline training recipe (Qwen3-8B L18, 500M tokens). All four architectures (Standard, Adaptive Cosine SAE, Per-Feature Adaptive Cosine SAE, Magnitude-Bypass SAE) share these settings. The Per-Feature Adaptive Cosine headline runs use the training/eval code in exp42c. [exp40] [exp42c]

Setting	Value
Model / site	`Qwen/Qwen3-8B-Base`, residual, layer 18
Activation / dictionary dim	$d_{\text{model}} = 4096$, $d_{\text{sae}} = 65{,}536$
Tokens / steps	500M tokens, 244,140 steps
Batch / sparsity	batch 2048, BatchTopK $k = 80$
Optimizer / LR	Adam, LR $5 \cdot 10^{-5}$, seed 42
LR schedule	1000-step warmup; constant until 80%, then linear decay
Auxiliary loss	weight $1/32$, $k_{\text{aux}} = 2048$, dead threshold 10M tokens
Decoder	unit-norm decoder rows after each step; geometric-median b_{dec}
Stability	gradient clipping max norm 1.0; bf16 activations
Eval cache	2M held-out tokens

Data. Training tokens come from the FineWeb `sample-10BT` subset (`HuggingFaceFW/fineweb`), tokenized with each model's bundled tokenizer. SAEBench probing datasets are at the versions shipped with `sae-bench`.

Reproducibility. Code and full experiment scripts are available at https://github.com/SilenNaihin/cosine-scored-saes. The 500M-token headline SAE checkpoints (Standard, Global a, Per-Feature Adaptive Cosine) are released at https://huggingface.co/Silen/cosine-scored-saes-qwen3-8b.

K. Statistical Summary

Table 22. Effect sizes and significance.

Claim	Strength
FVE difference $+8.0\%$ at L27, 50M tokens, no auxiliary loss [exp34]	41.7σ, $n = 3$ seeds
Dead-feature difference 48.7% [exp17]	37.5σ
Cos > inner above 50% at L27 [exp25]	$p < 2.4 \times 10^{-6}$
Dictionary efficiency: 16k Adaptive Cosine SAE alive > 49k Standard alive [exp17]	Cohen's $h = 1.358$
Gradient Q4 dominance: 35.3% vs. 13.5% [exp28]	Cohen's $h = 0.519$
Per-feature interpretability difference (50M/L27) [exp33]; matched at 500M headline (§F.2) [exp62]	$p = 0.88$ (n.s.)
Sparse probing top-1 $+14.6\%$ (per-feature), $+14.1\%$ (global) [exp40/42c/44] [exp61]	$n = 3$ seeds at 500M (Table 23)

Table 23. Multi-seed reproducibility at the 500M headline (Qwen3-8B L18, $d_{\text{sae}} = 65{,}536$). Mean $\pm$ SD over three SAE-training seeds $\{42, 123, 456\}$; seed 42 is the run reported in Table 1. [exp61]

	Standard	Global a	Per-feature
FVE	0.7702 ± 0.0002	0.7690 ± 0.0000	0.7707 ± 0.0002
Probing top-1	0.667 ± 0.003	0.808 ± 0.006	0.813 ± 0.013
Top-1 gap	—	$+14.1\%$	$+14.6\%$
Learned a	—	0.258 ± 0.001	0.076 ± 0.000

Across three full 500M SAE-training seeds the headline reproduces: FVE matches to ± 0.0002, the top-1 gap is stable at $+14.1\%$ (global) and $+14.6\%$ (per-feature), and the learned exponent a stays far below the inner-product limit with negligible seed variance. The reproducibility-only Magnitude-Bypass arm and the per-dataset breakdown (Table 9) are reported at the single seed used throughout the main text.

L. Salient-Concept Control

The probing and ablation advantages are aggregate (§G); they do not mean standard scoring fails to represent obvious concepts. To check this directly, we rank each 500M dictionary's features by selectivity for a content concept (mean activation on concept-positive minus concept-negative prompts) and ablate the top-K as a set, measuring the within-arm precision (on-concept effect / off-concept effect) as a function of K. [exp65] For both salient concepts tested (Python code, French text), *both* architectures have a top concept feature that fires strongly on-concept and is silent off-concept (off-concept ablation KL ≈ 0 for $K \leq 10$ under either score), so neither dictionary lacks the concept. The cosine advantage is therefore not visible at the single-salient-concept level; it emerges in the aggregate over the eight SAEBench probing categories and the long feature tail, consistent with the matched-feature decomposition (§G) and the aggregate ablation-precision result (Table 18).