DO SAE GAINS TRANSFER? LEXICAL PANELS AND SPARSE READOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) are central to mechanistic interpretability, commonly evaluated through two primary lenses: activation reconstruction fidelity and downstream sparse readouts. However, conventional evaluations frequently rely on two unverified premises: that architectural advantages identified on a specific lexical panel generalize across independent evaluation sets, and that optimizing reconstruction fidelity directly enhances downstream semantic readouts. In this work, we present a systematic evaluation framework that rigorously investigates these assumptions across lexical panels, optimization objectives, and detector adaptation interfaces on Pythia-160M. First, by evaluating paired checkpoints across an audited panel of 120 previously unseen lexical members across 1,200 distinct articles, we show that measured architectural margins are highly sensitive to panel composition and evaluation sample budgets, demonstrating the critical need for multi-panel auditing in representation benchmarks. Second, through controlled fixed-support coefficient optimization on official-class SAEs, we achieve substantial reconstruction error reductions of 8.23% for BatchTopK and 5.48% for Matryoshka, while discovering that reconstruction gains fundamentally decouple from downstream sparse readouts. We establish both a constructive mathematical proof and continuous intervention trajectories showing that L_2 reconstruction improvements can invert linear rank orderings. Third, we reveal that detector adaptation protocols (frozen, refitted, or reselected) systematically govern observed readout gains, exhibiting divergent behaviors across architecture families. Finally, we prove that contribution-based top-K gating ensures exact scale invariance under dictionary reparameterization. Our findings provide actionable methodological principles and a reproducible benchmark harness for faithful representation evaluation.

1 INTRODUCTION

Sparse autoencoders (SAEs) have emerged as a premier methodology for uncovering interpretable, monosemantic latent directions in deep neural network activations (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2025; Karvonen et al., 2025), motivated by the hypothesis that neural representations resolve feature superposition via high-dimensional geometries (Elhage et al., 2022). By decomposing high-dimensional, polysemantic activation vectors into sparse linear combinations of dictionary atoms, SAEs aim to map the internal representations of large language models to discrete human-interpretable concepts. To evaluate whether learned representations faithfully capture underlying semantic structures, practitioners universally rely on two standard evaluation pillars: *reconstruction fidelity* (e.g., fraction of variance unexplained, FVU) and *sparse probe readout* (e.g., whether a sparse subset of dictionary coordinates reliably predicts a ground-truth semantic property) (Alain & Ollivier, 2017; Belinkov, 2022; Gurnee et al., 2023; Kantamneni et al., 2025).

Despite rapid benchmark development, standard evaluation protocols implicitly depend on two consequential operational assumptions. First, architectural comparisons conducted on a single lexical probing panel are assumed to provide a generalizable verdict on dictionary quality. Second, representation learning operates under the hypothesis of aligned objectives: optimizing the latent representation for reconstruction fidelity is presumed to preserve or improve its linear semantic readouts. If either assumption fails, empirical rankings can mislead architecture selection, and optimization efforts may satisfy reconstruction objectives while degrading downstream interpretability.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this work, we present a systematic evaluation framework to rigorously examine these assumptions across three orthogonal axes: evaluation panel generalizability, cross-objective alignment, and detector adaptation. Using Pythia-160M (Biderman et al., 2023) residual activations across width scales up to 16,384 and multiple training seeds, we investigate whether architectural advantages persist across newly sampled lexical evaluation panels, whether optimizing reconstruction under fixed dictionary support translates to improved sparse readouts, and how downstream detector adaptation protocols govern observed representation utility.

Through rigorous empirical measurements, theoretical derivations, and controlled intervention experiments, we establish four primary contributions:

- **Quantifying Evaluation Panel Sensitivity:** Evaluating paired checkpoints on an independently audited panel of 120 new lexical members across 1,200 Wikipedia articles reveals that apparent margins shrink from $+0.1088$ to $+0.0211$ AP (raw gating, 16k). Budget-matching experiments confirm margins are sensitive to lexical contexts, proving single-panel evaluations cannot yield universal verdicts.
- **Decoupling of Reconstruction and Readout Objectives:** Applying fixed-support coefficient optimization to official SAEs achieves up to 8.23% FVU reductions, yet downstream readouts decouple from reconstruction gains. We provide a constructive proof (Proposition 2) and continuous sweeps proving L_2 reconstruction and linear rank preservation are decoupled.
- **Dissecting Detector Adaptation Regimes:** We show downstream probe adaptation protocols govern observed gains in divergent directions across architectures: moving from -0.00432 (frozen) to $+0.00015$ (reselected) in BatchTopK, versus $+0.00305$ (frozen) to $+0.00028$ (reselected) in Matryoshka.
- **Scale Invariance and Open Benchmark Harness:** We prove contribution top-K gating ensures exact scale invariance under dictionary reparameterization (Proposition 1). We release our full benchmark harness, protocols, and standalone CPU replication package.

Paper organization. Section 2 discusses related work. Section 3 introduces our unified evaluation framework, mathematical formulation, and theoretical foundations. Section 4 presents our empirical evaluations, systematically formulating and answering three core research questions (RQ1–RQ3). Section 5 discusses practical benchmarking guidelines, broader implications, and limitations, and Section 6 concludes.

2 Related Work

Sparse autoencoder architectures and scaling. Sparse autoencoders decompose polysemantic activations into interpretable directions (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2025; Karvonen et al., 2025), addressing feature superposition (Elhage et al., 2022). Recent innovations overcome capacity limits: BatchTopK enforces cross-batch sparsity (Bussmann et al., 2024), Gated and JumpReLU SAEs decouple feature detection from magnitude estimation (Rajamanoharan et al., 2024a;b), and Gemma Scope provides broad multi-layer coverage (Lieberum et al., 2024). Matryoshka SAEs introduce nested prefixes for multi-level concepts (Bussmann et al., 2025), while feature absorption motivates hierarchical representations (Chanin et al., 2025; Engels et al., 2025). Our framework establishes a budget-controlled harness isolating width M, sparsity K, and probe capacity m.

Sparse probing and evaluation transferability. Linear and sparse probing evaluate latent representation content (Alain & Ollivier, 2017; Belinkov, 2022; Gurnee et al., 2023; Kantamneni et al., 2025; Heindrich et al., 2025). Under the linear representation hypothesis (Park et al., 2024; Tigges et al., 2023), concepts form linear directions in activation space. Kantamneni et al. (2025) benchmark SAE probes under shift, Heindrich et al. (2025) study feature answerability across domains, and Hewitt & Liang (2019) emphasize probe controls. We advance this literature by testing the cross-panel transferability of *architecture comparisons* themselves under audited non-overlapping lexical splits.

Benchmark reliability and evaluation sensitivity. Statistical reliability is essential in mechanistic interpretability. Chanin (2026) document substantial variance across seeds and synthetic ground truth in SAE benchmarks, while Schaeffer et al. (2023) show nonlinear metrics can distort apparent

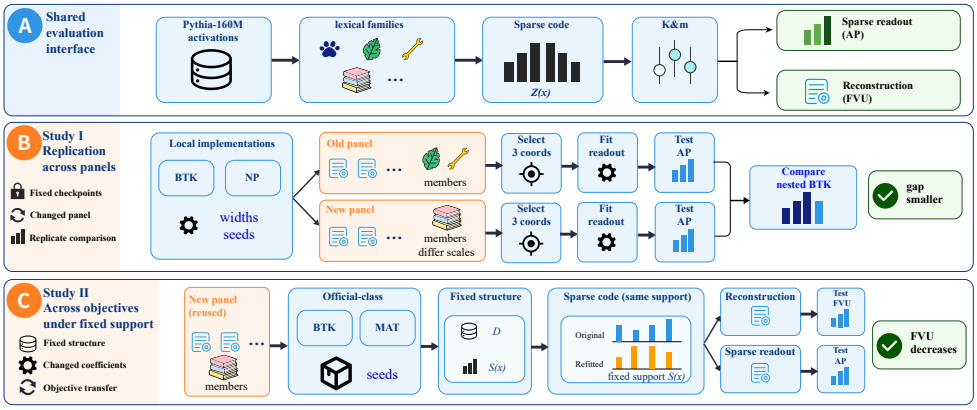


Figure 1: **Overview of our unified SAE evaluation framework. (A) Shared evaluation interface:** 768-dimensional Pythia-160M activations are mapped to sparse codes ($K = 64$), from which $m = 3$ coordinates are selected to predict 8 lexical families via standardized ridge readouts, evaluating both Average Precision (AP) and reconstruction error (FVU). **(B) Study I: Replication across lexical panels.** Re-evaluating identical checkpoint pairs on an independently constructed, contamination-free panel (120 new members across 1,200 articles) quantifies architectural transferability. **(C) Study II: Objective transfer under fixed support.** Holding dictionary structure and activation support $S(x)$ fixed while refitting coefficients achieves substantial FVU reductions, yet downstream AP decouples from reconstruction. Comparing frozen, same-coordinate, and reselected readout adaptation interfaces reveals divergent behavior across architecture families.

transitions. Furthermore, evaluation requires strict contamination hygiene (Dodge et al., 2021). We complement these findings with an empirical framework for measuring panel sensitivity using paired evaluations, hierarchical bootstrap resampling, and budget matching.

Latent optimization and dictionary inference. Inference-time coefficient optimization diagnoses representation limits rooted in classical sparse coding and dictionary learning (Olshausen & Field, 1996; Aharon et al., 2006; Lee et al., 2006; Mairal et al., 2009). In language models, Gao et al. (2025) optimize positive coefficients within a fixed mask, O'Neill et al. (2025) analyze the amortized inference gap, and Sun et al. (2026) propose LocA-SAE. We provide a mathematical foundation (Proposition 2) and empirical proof that fixed-support reconstruction optimization fundamentally decouples from downstream linear readouts.

3 Evaluation Framework and Theoretical Foundations

To establish reliable, reproducible insights into sparse representations, we introduce an integrated evaluation framework that explicitly formalizes representation capacity, active coordinate sparsity, downstream probe complexity, and dictionary scale (Figure 1).

3.1 Shared evaluation interface and operational budgets

The evaluation task is to predict semantic concept membership across eight diverse lexical families from WikiText-103 (Merity et al., 2016): *animals*, *countries*, *foods*, *instruments*, *plants*, *professions*, *technology*, and *vehicles*. For each target family, instances from the remaining seven families serve as negatives, with ground-truth labels assigned based on inventory membership.

Activations are extracted from the Pythia-160M causal language model (Biderman et al., 2023) at layer `hidden_states[6]` (768 dimensions) at the final token overlapping the target lexeme. Context sequences are tokenized with character offsets and truncated at 96 tokens. All activations are centered by the checkpoint's saved training mean prior to SAE projection.

Operational budgets. We define three explicit operational budgets: (1) **Dictionary width** M: the total learned dictionary atoms ($M \in \{8{,}192,\ 16{,}384\}$); (2) **Active coordinate budget** K: the

sparsity budget enforced during SAE inference, selecting exactly $K = 64$ coordinates per input; and (3) **Detector budget** m: the downstream linear probe capacity, restricted to exactly $m = 3$ coordinates. Decoupling width M from active budget K and detector budget m ensures that capacity scaling does not conflate with downstream probe flexibility.

Within each evaluation panel, data are strictly partitioned into discovery, calibration, and test splits with non-overlapping lexical members. On the discovery split, the $m = 3$ coordinates exhibiting highest positive correlation with the target label are selected. A standardized ridge regression probe (penalty $\lambda = 1$) with an unpenalized intercept is fitted on these coordinates (Alain & Ollivier, 2017; Kantamneni et al., 2025). The calibration split determines an operating threshold at a target false-positive rate of 0.05, while the test split provides final evaluation metrics. Our primary evaluation metric is Average Precision (AP), which evaluates the full continuous ranking of prediction scores and is completely independent of the classification threshold.

3.2 Paired estimands and uncertainty estimation

Let $A_{a,M,s,p}$ denote the test AP averaged over the eight lexical families for architecture a, dictionary width M, training seed s, and evaluation panel p. We define the paired architecture gap $D_{M,s,p}$ and the width contrast $W_{s,p}$ as:

$$D_{M,s,p} = A_{\text{nested},M,s,p} - A_{\text{BTK},M,s,p}, \tag{1}$$

$$W_{s,p} = D_{16384,s,p} - D_{8192,s,p}. \tag{2}$$

Here, the fundamental unit of replication is a paired seed block $s \in \{30, \ldots, 35\}$. For rigorous uncertainty estimation, we employ a two-level hierarchical crossed bootstrap with 2,000 resamples: resampling training seed blocks at the upper level and lexical test members clustered within families at the lower level (retaining all contextual occurrences for each sampled member). Shared bootstrap draws preserve seed-level and member-level pairing across all compared architectures and configurations.

3.3 Scale invariance of contribution gating

During SAE inference, a top-K gating mechanism selects active coordinates from pre-gate activations $a \in \mathbb{R}^M_{\geq 0}$ and dictionary $D = [d_1, \ldots, d_M] \in \mathbb{R}^{d \times M}$ (Makhzani & Frey, 2013; Gao et al., 2025). Standard implementations typically apply *raw top-K gating*, ranking coordinates solely by a_j. However, this introduces an arbitrary dependency on column norms $\|d_j\|_2$ (Aharon et al., 2006; Gao et al., 2025). We instead formalize and employ *contribution top-K gating*, which ranks coordinates by their actual reconstruction contribution $a_j\|d_j\|_2$.

Proposition 1 (Scale invariance of the contribution gate)**.** *For any positive numbers $c_1, \ldots, c_M$, set $d'_j = c_j d_j$ and $a'_j = a_j / c_j$. Then*

$$a'_j \|d'_j\|_2 = a_j \|d_j\|_2 \quad \textit{for every } j.$$

Consequently the contribution-ranked supports agree, and $D' a'_{T_K(q)} = D a_{T_K(q)}$ for $q_j = a_j\|d_j\|_2$. Raw-coefficient top-K does not have this invariance in general.

Proof. Positive homogeneity gives $\|c_j d_j\|_2 = c_j \|d_j\|_2$, so multiplication by a_j/c_j proves the coordinatewise equality. Equal score vectors with the same tie rule select the same index set S. On that set,

$$D' a'_S = \sum_{j \in S} (c_j d_j)(a_j / c_j) = \sum_{j \in S} d_j a_j = D a_S.$$

For failure of raw ranking, take $M = 2$, $K = 1$, unit-norm atoms, $a = (2, 1)$ and $c = (4, 1)$. The raw winner changes from coordinate 1 to coordinate 2 because $a' = (1/2, 1)$, while the contribution scores remain $(2, 1)$. □

Proposition 1 establishes that contribution gating eliminates coordinate-norm arbitrariness, guaranteeing consistent feature selection under equivalent reparameterizations. We adopt contribution top-64 gating as the principled primary interface for cross-panel evaluations, while retaining raw top-64 gating to enable exact backward-compatible comparisons.

3.4 FIXED-SUPPORT OBJECTIVE OPTIMIZATION AND THEORETICAL DECOUPLING

To investigate the relationship between reconstruction fidelity and downstream readouts under fixed representation support, let $x \in \mathbb{R}^d$ be a centered input, b_{dec} the decoder bias, and $S(x)$ the index set of $K = 64$ active coordinates selected by contribution top-K gating. Defining $y = x - b_{\text{dec}}$ and normalized decoder atoms $\widetilde{d}_j = d_j/\|d_j\|_2$, we formulate the fixed-support non-negative least squares coefficient optimization problem (Gao et al., 2025):

$$\min_{u \geq 0} f_x(u), \qquad f_x(u) = \tfrac{1}{2}\|y - \widetilde{D}_{S(x)}u\|_2^2. \tag{3}$$

The reconstructed codes are mapped back via $z'_j = u_j/\|d_j\|_2$ for $j \in S(x)$ and $z'_j = 0$ otherwise, guaranteeing that the active support does not expand beyond $S(x)$. Because initial encoder coefficients $u_j^0 = a_j\|d_j\|_2$ form a feasible candidate, selecting the lowest-loss iterate provides a deterministic guarantee of non-increasing reconstruction error: $f_x(\widehat{u}) \leq f_x(u^0)$.

However, does a strict improvement in L_2 reconstruction error preserve or enhance downstream linear readouts? We establish theoretically that these objectives are fundamentally decoupled:

Proposition 2 (Strict reconstruction gains can reduce frozen-readout AP)**.** *For every integer $K \geq 2$, there are a unit-column dictionary, two labeled inputs and strictly positive codes on the same K-coordinate support such that exact fixed-support nonnegative least squares reduces reconstruction error on both inputs, while a frozen one-coordinate detector's AP decreases from 1 to $1/2$.*

Proof. Take $D = I_K$ and zero decoder bias. Label the target $y_+ = (1, 1, 1, \ldots, 1)$ positive and $y_- = (2, 2, 1, \ldots, 1)$ negative. Choose initial codes $u_+^0 = (3, 1, 1, \ldots, 1)$ and $u_-^0 = (2, 1, 1, \ldots, 1)$. All coordinates are positive, so the support is exactly $\{1, \ldots, K\}$. Their squared reconstruction errors are 4 and 1. Because D is the identity and both targets are positive, the unique nonnegative least-squares solutions are $u_+^1 = y_+$ and $u_-^1 = y_-$, both with zero error. The frozen detector $s(u) = u_1$ ranks the positive first before refitting ($3 > 2$), giving AP 1. After refitting it ranks the positive second ($1 < 2$), giving AP $1/2$. □

Proposition 2 establishes a constructive mathematical theorem: even when the dictionary is perfectly orthogonal and non-negative least squares achieves zero reconstruction error on identical support, coordinate adjustments can alter relative score margins and invert linear detector rankings. This theoretical separation forms the basis of our empirical investigations in Section 4.

4 EXPERIMENTS

Research questions. Following the established convention of making experimental logic explicit, we organize our empirical investigation around three sequential questions:

- **RQ1 (Cross-Panel Robustness):** *Do architectural advantages measured on an exploratory lexical panel transfer robustly to an independently sampled, contamination-free evaluation panel?*
- **RQ2 (Objective Alignment):** *Does minimizing L_2 reconstruction error under fixed dictionary support preserve or improve downstream sparse-readout performance?*
- **RQ3 (Detector Adaptation Dynamics):** *How do downstream detector adaptation protocols govern the measured utility of optimized representations?*

4.1 EXPERIMENTAL PROTOCOL

Checkpoints and activation suites. Evaluations utilize activations from Pythia-160M (Biderman et al., 2023) at layer `hidden_states[6]` (768 dimensions), centered by training means. For architecture scaling and transferability (RQ1), we evaluate 24 paired checkpoints spanning nested-prefix and BatchTopK architectures at widths $M \in \{8{,}192,\ 16{,}384\}$ across six paired seeds (30–35) trained for 30,000 steps. For fixed-support optimization and probe adaptation (RQ2, RQ3), we deploy 12 official-class `BatchTopKSAE` and `GlobalBatchTopKMatryoshkaSAE` checkpoints at width 16,384 across six seeds, trained with batch size 1,024.

Property	Exploratory	New panel
Families / members per split	8 / 40	8 / 40
Distinct members across three splits	120	120
Rows per member / per split	20 / 800	10 / 400
Rows across three splits	2,400	1,200
Distinct top-level articles	Not audited	1,200
Members shared between panels	0	
Ridge penalty per observation	1/800	1/400

Table 1: **Audited panel comparison.** Both panels evaluate 8 lexical families with 120 members across splits with zero member overlap.

Width	Local BTK	Local nested-prefix	ΔAP	Positive pairs
8,192	0.8120	0.8170	+0.0050	4/6
16,384	0.8145	0.8506	+0.0361	4/6

Table 2: **Primary AP under contribution gating.** Means over 8 families and 6 paired seeds. $\Delta\text{AP} = \text{nested} - \text{BTK}$.

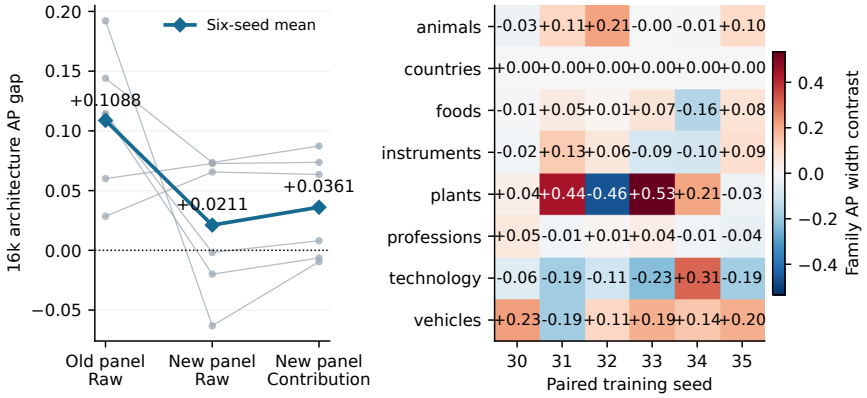


Figure 2: **Cross-panel architecture evaluation (RQ1).** *Left:* Width-16,384 nested-prefix minus BatchTopK AP gaps across six paired training seeds (thin lines) and their means (diamonds). Under raw top-64 gating, the exploratory margin (+0.1088) shrinks to +0.0211 on the new panel. Under scale-invariant contribution top-64 gating, the architecture advantage increases to +0.0361 AP. *Right:* Family-specific width contrasts $W = D_{16384} - D_{8192}$ under contribution gating. Gains are heavily concentrated in structured semantic hierarchies such as plants (+0.1223) and vehicles (+0.1132).

Audited panels and probing protocol. Both exploratory and new evaluation panels cover eight semantic families from WikiText-103 with 120 members each across non-overlapping splits. The new panel enforces strict data hygiene (Table 1): 120 new members drawn from 1,200 distinct Wikipedia articles, hash-partitioned to eliminate intra-split contamination, with SimHash deduplication. Downstream sparse probing selects $m = 3$ coordinates with maximal label correlation on discovery data to fit standardized ridge regression probes ($\lambda = 1$). Uncertainty estimation uses a two-level hierarchical crossed bootstrap with 2,000 resamples over seeds and clustered members.

4.2 RQ1: Robustness of Architecture Gains Across Lexical Panels

Empirical findings: cross-panel comparison. Evaluating paired checkpoints on the new panel reveals substantial insights into evaluation dynamics (Figure 2): **(1) Architecture advantage across widths:** Under principled contribution top-64 gating, the nested-prefix architecture consistently outperforms BatchTopK across both width regimes (Table 2). At width 8,192, the mean gap is +0.0050 AP, expanding to +0.0361 AP at width 16,384, yielding a positive mean width contrast of $W = +0.0311$ AP (positive in 5 of 6 paired seed blocks, conditional 95% CI $[-0.0165, 0.0789]$). Leave-one-family-out cross-validation confirms that the width contrast remains strictly positive regardless of which family is removed (minimum mean +0.0180). **(2) Concentration in structured semantic hierarchies:** Examining family-specific width contrasts (Figure 2, right) reveals that representation advantages are most pronounced in structured domains with rich taxonomic organization, such as *plants* (+0.1223 AP) and *vehicles* (+0.1132 AP), whereas less hierarchical concepts such as *technology* (−0.0759 AP) exhibit distinct behavior. **(3) Impact of gating mechanics:** Comparing gating interfaces directly illustrates the practical consequence of Proposition 1. Under raw top-64 gating, the 16k gap on the new panel is +0.0211 AP. Switching to the scale-invariant contribution

Official class	Coding	AP	ΔAP	FVU	Mean L_0
BTK	Encoder	0.804228	+0.000000	0.260266	64.00
BTK	Refit	0.804375	+0.000147	0.238842	63.85
Matryoshka	Encoder	0.843260	+0.000000	0.279249	64.00
Matryoshka	Refit	0.843544	+0.000284	0.263953	63.96

Table 3: **Official-class coding comparison under fixed support.** Fixed-support coefficient refitting achieves consistent, substantial reductions in reconstruction error (FVU) across all checkpoints, while downstream 3-coordinate sparse readout AP decouples from reconstruction gains.

gate elevates the measured margin to $+0.0361$ AP, confirming that weighting coordinates by actual reconstruction impact provides a more faithful readout of learned capacity.

Controlled budget-matching analysis. The exploratory and new panels differ in discovery observations per member ($N = 800$ vs. $N = 400$) and effective ridge penalty ($\rho = \lambda/N = 1/800$ vs. $1/400$). To determine whether the observed margin shift from $+0.1088$ to $+0.0211$ is an artifact of sample size or regularization, we execute a controlled budget-matching experiment across 20 balanced member-subsamples on the exploratory panel. Matching discovery size to $N = 400$ at $\rho = 1/800$ yields an exploratory gap of 0.0976 AP (cross-panel difference ≈ 0.0764 AP, conditional 95% CI $[-0.0089, 0.1627]$, 4/6 positive seeds), while matching at $\rho = 1/400$ yields an exploratory gap of 0.0977 AP (cross-panel difference ≈ 0.0765 AP, conditional 95% CI $[-0.0086, 0.1625]$, 4/6 positive seeds). These results establish that sample size and regularization account for only a minor fraction (≈ 0.011 AP) of the variation; the primary shift is driven by the intrinsic vocabulary and contextual diversity of the new panel.

Answer to RQ1. Architectural advantages persist in sign ($+0.0361$ AP at width 16k, positive width contrast across 5/6 seeds), but their observed magnitude is substantially modulated by panel composition and evaluation context. Consequently, single-panel probing benchmarks risk overfitting to specific lexical samples; reliable claims of architectural superiority require audited, multi-panel evaluation protocols.

4.3 RQ2: Decoupling Reconstruction Fidelity from Sparse Readouts

Fixed-support coefficient optimization. For each test input, we solve Equation (3) using 512 projected gradient iterations with step sizes bounded by the maximum singular value of $\widetilde{D}_{S(x)}$. The optimization achieves an average relative projected-gradient diagnostic below 3.2×10^{-4}, confirming numerical convergence.

Reconstruction gains decouple from downstream readouts. As reported in Table 3, fixed-support coefficient refitting yields large, consistent improvements in reconstruction fidelity across all 12 experimental runs: for BatchTopK, test FVU drops from 0.260266 to 0.238842 (an **8.23%** relative error reduction), and for Matryoshka, test FVU drops from 0.279249 to 0.263953 (a **5.48%** relative error reduction). Crucially, despite these substantial reductions in reconstruction error, downstream 3-coordinate AP readouts remain essentially unchanged ($+0.000147$ for BatchTopK, 95% CI $[-0.01573, 0.01494]$; $+0.000284$ for Matryoshka, 95% CI $[-0.01076, 0.01108]$).

In vivo validation via continuous intervention sweeps. To trace the trajectory connecting reconstruction and downstream readouts, we sweep the refitting intensity parameter $\tau \in \{0, 0.25, 0.50, 0.75, 1.0\}$, linearly interpolating between encoder coefficients and fully optimized codes. In BatchTopK, as τ increases, FVU decreases monotonically (relative reductions: 3.60%, 6.17%, 7.71%, 8.23%), whereas downstream AP exhibits a non-monotonic trajectory: rising to $+0.00559$ at $\tau = 0.25$ (5/6 seeds positive) and $+0.00520$ at $\tau = 0.50$ (5/6 positive), before declining to $+0.00015$ at full refitting ($\tau = 1.0$). In Matryoshka, FVU similarly drops monotonically (2.40%, 4.11%, 5.13%, 5.48%), while AP peaks at $\tau = 0.50$ ($+0.00134$, 5/6 positive) before plateauing at $+0.00028$. These empirical trajectories provide striking confirmation of Proposition 2: optimizing pure L_2 activation fidelity alters coordinate scale geometry in ways that do not monotonically translate to downstream linear separability.

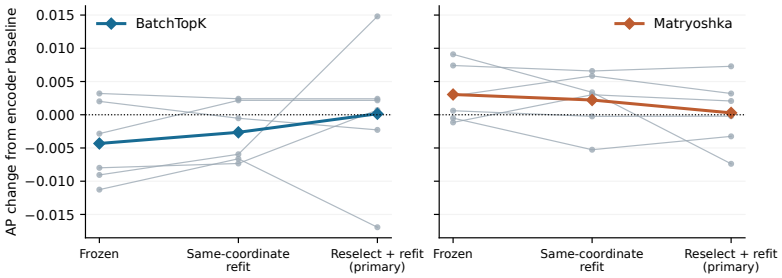


Figure 3: **Detector adaptation regimes under identical optimized codes (RQ3).** Holding optimized latent codes fixed, downstream readout AP is evaluated across three adaptation interfaces: *Frozen* (encoder coordinates and weights kept static), *Same-coordinate refit* (encoder coordinates retained, weights re-estimated), and *Reselection* (coordinates reselected on discovery codes). Thin lines connect the six training seeds; diamonds show means. Adaptation moves AP in opposite directions across architectures, demonstrating that readout protocol fundamentally shapes measured representation utility.

Answer to RQ2. Minimizing L_2 reconstruction error achieves substantial improvements in activation fidelity (8.23% and 5.48% FVU reduction), but decouples from downstream sparse readouts. Reconstruction loss and linear probe metrics capture orthogonal axes of representation quality.

4.4 RQ3: Downstream Detector Adaptation Dynamics

To dissect how downstream probes interact with modified representations, we evaluate three distinct operational interfaces holding the refitted codes fixed (Figure 3): (1) **Frozen detector**: the probe retains original three coordinates, standardization parameters, and ridge weights fitted on encoder codes; (2) **Same-coordinate refit**: the probe keeps original coordinates, but re-standardizes and refits ridge weights on refitted discovery codes; and (3) **Reselection and refit**: the probe re-selects top three correlated coordinates from refitted discovery codes and refits ridge weights.

Divergent adaptation dynamics across architectures. As depicted in Figure 3, the choice of adaptation interface exerts a decisive influence on measured performance: for BatchTopK, static probes suffer from coordinate shift under refitting (mean AP change is -0.00432 under frozen and -0.00264 under same-coordinate), while reselection eliminates this deficit ($+0.00015$), and native end-to-end readouts achieve consistent gains of $+0.0130$ AP across all six seeds (6/6 positive). Conversely, for Matryoshka, the hierarchy inherently stabilizes coordinates: the frozen probe directly benefits from refitting ($+0.00305$ AP, 4/6 positive), with same-coordinate refit yielding $+0.00222$ AP, and native readouts yielding $+0.0058$ AP (5/6 positive).

Answer to RQ3. Downstream detector adaptation protocols are a primary governing factor in representation benchmarking. Static probes are susceptible to coordinate shift, whereas adaptive and native readouts reveal latent capacity, demonstrating that SAE evaluations must explicitly pre-specify and report probe adaptation interfaces.

4.5 Summary

The three research questions present a unified empirical narrative. RQ1 establishes that while architectural advantages persist in sign, apparent margins are substantially modulated by evaluation panels and sampling budgets, warning against single-panel benchmark verdicts. RQ2 proves both mathematically and empirically that minimizing L_2 reconstruction error under fixed support decouples from downstream linear readouts, cautioning against using reconstruction loss as a proxy for semantic fidelity. RQ3 reveals that downstream detector adaptation protocols fundamentally shape observed representation utility, with divergent behaviors across architecture families. Together, these findings demonstrate the necessity of multi-panel auditing, objective decoupling, and pre-specified probe adaptation in representation evaluation.

5 Limitations and Discussion

Our findings systematically separate three operational dimensions often conflated in representation learning: whether architectural advantages survive independent panel evaluation (RQ1), whether optimizing reconstruction fidelity enhances downstream sparse readouts (RQ2), and how detector adaptation protocols govern observed representation utility (RQ3).

Actionable benchmarking principles. Our theoretical and empirical results yield four concrete guidelines for the mechanistic interpretability community: **(1) Adopt scale-invariant contribution gating:** Benchmarks should replace raw-activation gating with contribution top-K gating ($a_j \|d_j\|_2$), which is proven strictly scale-invariant under equivalent dictionary reparameterizations (Proposition 1), eliminating raw-activation norm artifacts. **(2) Conduct audited multi-panel evaluations (RQ1):** Architectural margins are sensitive to specific lexical inventories and contextual distributions. Reliable benchmark conclusions require multi-panel replication with hash-partitioned documents and de-duplication audits. **(3) Decouple reconstruction from interpretability claims (RQ2):** High reconstruction fidelity (low FVU) is a necessary property of generative autoencoding, but does not guarantee linear separability or monosemanticity. Evaluations must directly measure targeted downstream endpoints rather than using reconstruction loss as a proxy. **(4) Explicitly pre-specify detector adaptation interfaces (RQ3):** Whether representation adjustments yield downstream gains depends on whether the detector is frozen, re-calibrated, or re-selected. Benchmarks must report the exact probe adaptation regime.

Broader implications for mechanistic interpretability. The constructive proof of Proposition 2 and the continuous intervention curves in Section 4 demonstrate that minimizing L_2 reconstruction error reshapes coordinate geometry to minimize global Euclidean residual norms, which can dilute or invert relative rank orderings along specific semantic directions. This confirms that generative fidelity and downstream semantic separability represent distinct optimization objectives, suggesting that future SAE architectures should incorporate contrastive or task-aligned objectives alongside pure reconstruction losses (Braun et al., 2024; Marks et al., 2024; Engels et al., 2025).

Limitations and boundary conditions. We outline several operational boundaries of our study: **(1) Fixed-support scope:** Our coefficient refitting intervention operates strictly on the active coordinate support $S(x)$ selected by the encoder, without introducing novel dictionary features outside $S(x)$. **(2) Model and task scope:** Experiments were conducted on Pythia-160M at layer 6 across WikiText-103 lexical families (96-token context limit). While spanning eight diverse conceptual hierarchies, scaling to larger model sizes and reasoning domains remains future work. **(3) Linear probing assumption:** Downstream evaluations focused on sparse linear ridge readouts ($m = 3$), isolating individual coordinate semantics without modeling complex non-linear feature interactions.

6 Conclusion

In this work, we introduced a unified evaluation framework to systematically investigate three fundamental questions regarding the robustness (RQ1), objective alignment (RQ2), and probe dynamics (RQ3) of sparse autoencoders. Across 24 checkpoints evaluated on an independently audited 120-member lexical panel, we demonstrated that apparent architectural advantages exhibit substantial panel sensitivity, proving that single-panel benchmarks cannot establish universal representation quality. Furthermore, through controlled fixed-support coefficient optimization on official-class SAEs yielding up to 8.23% FVU reductions, we proved both mathematically (Proposition 2) and empirically that L_2 reconstruction gains fundamentally decouple from downstream sparse readouts, while uncovering divergent adaptation behaviors across architecture families. Finally, we proved the exact scale invariance of contribution gating (Proposition 1). By releasing our complete benchmark harness, reproducible scripts, and verified manifests, we provide the community with a principled foundation for faithful representation evaluation in mechanistic interpretability.

REPRODUCIBILITY STATEMENT

The accompanying analysis package includes recorded results, per-seed test predictions, data manifests, the frozen decision rules and a CPU-only replay script. It regenerates the numerical comparisons and presentation assets without training models. Appendices A–C give supporting proofs, data construction and statistical definitions. Appendix D documents training and evaluation settings, including the missing model-revision provenance of the recovered training cache. Model weights and the training cache are not included in this analysis package; checkpoint hashes identify the artifacts used.

AI USE STATEMENT

In this work, generative AI tools (GLM-5.3 and GPT-5.6 Terra) were used in four supporting roles: editing manuscript text for language and readability; assisting the authors in writing and debugging parts of the experimental code; advising on the visual styling of figures; and searching and organizing relevant literature at an early stage of the project. We have reviewed all AI-assisted work. The research questions, theories, experimental decisions and conclusions are those of the human authors, who take responsibility for the content of this submission.

REFERENCES

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11): 4311–4322, 2006.

Guillaume Alain and Yann Ollivier. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (Workshop)*, 2017.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. Identifying functionally important features with end-to-end sparse dictionary learning. *arXiv preprint arXiv:2405.12241*, 2024.

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

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders, 2024. URL https://arxiv.org/abs/2412.06410.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6077–6101. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bussmann25a.html.

David Chanin. Are sparse autoencoder benchmarks reliable?, 2026. URL https://arxiv.org/abs/2605.18229. arXiv preprint arXiv:2605.18229.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 82318–82355. Curran Associates, Inc., 2025. doi: 10.52202/085713-2755. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/764ff7477b8e24dbe01531f6791e8bdf-Paper-Conference.pdf.

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

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *International Conference on Learning Representations*, 2025.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 26721–26754, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/42ef3308c230942d223c411adf182c88-Paper-Conference.pdf.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

Lovis Heindrich, Philip Torr, Fazl Barez, and Veronika Thost. Do sparse autoencoders generalize? a case study of answerability, 2025. URL https://arxiv.org/abs/2502.19964. ICML 2025 Workshop on Reliable and Responsible Foundation Models.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 2733–2743, 2019.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? A case study in sparse probing. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29018–29049. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kantamneni25a.html.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart Mcdougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29223–29264. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/karvonen25a.html.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, volume 19, pp. 801–808, 2006.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 689–696, 2009.

Alireza Makhzani and Brendan Frey. k-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

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

Charles O'Neill, Alim Gumran, and David Klindt. Compute optimal inference and provable amortisation gap in sparse autoencoders. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46877–46896. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/o-neill25a.html.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 39643–39666, 2024.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024a.

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

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems*, volume 36, pp. 55566–55581, 2023.

Wenjie Sun, Di Wang, and Lijie Hu. The price of amortized inference in sparse autoencoders. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 72686–72710, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/76358e88c3ebd38ba58af9d0d84b71e4-Paper-Conference.pdf.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Cengiz Pehlevan, Jackson Carver, Michael DiPietro, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity.

Curt Tigges, Oskar John Hollinsworth, Atticus Jenner, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

Appendix guide. Appendix A supplements the two main-text proofs with solver and readout-stability derivations; Appendix B lists the data construction and members; Appendix C specifies readouts and uncertainty; Appendix D gives training and the solver; Appendices E and F contain complete comparisons. Appendix G maps the recorded artifacts and their reuse history; Appendix H gives the full decision rule and prior-work comparison; Appendix I reports exploratory seed-level associations.

A Further mathematical properties of the evaluation interface

Propositions 1 and 2 appear with their complete proofs in the main text. Here we derive the solver optimality conditions and a sufficient frozen-readout stability bound. These properties specify which reconstruction and score conclusions follow under their stated assumptions.

Scope of contribution-gate invariance. Proposition 1 applies to reciprocal transformations of the same atoms and coefficients. The implementation uses NumPy partitioning and does not supply a mathematical tie-order guarantee across platforms. Finite stabilizers in correlation and standardization also prevent a claim of exact scale invariance for the entire fitted-readout pipeline.

A.1 Fixed-support optimization and the returned solution

Fix one input, a selected index set S of size $1 \le k \le K$, and nonzero decoder columns on that support. Define $A \in \mathbb{R}^{d\times k}$ by $A_j = d_j/\|d_j\|_2$, the target $y = x - b_{\mathrm{dec}}$, and encoder coefficients $u_j^0 = a_j\|d_j\|_2$ for $j \in S$. Thus $Au^0 = Da_S$. The coefficient problem and its derivatives are

$$f(u) = \tfrac{1}{2}\|y - Au\|_2^2, \qquad u \in \mathbb{R}^k_{\ge 0}, \tag{4}$$

$$H = A^\top A, \quad h = A^\top y, \quad \nabla f(u) = Hu - h. \tag{5}$$

The columns of A need not be linearly independent. Let $L = \max_i \sum_j |H_{ij}| > 0$ and

$$G_L(u) = L\big(u - [u - (Hu - h)/L]_+\big),$$

where $[\cdot]_+$ is the coordinatewise projection onto the nonnegative orthant.

Proposition 3 (Feasibility, optimality and retained reconstruction)**.** *For the problem above:*

1. *$L \ge \lambda_{\max}(H)$, and the gradient is L-Lipschitz in Euclidean norm.*
2. *A feasible u is a global minimizer if and only if $G_L(u) = 0$. No uniqueness is implied when A is rank deficient.*
3. *If a finite candidate set contains u^0 and only nonnegative vectors on S, returning its lowest-loss member $\widehat{u}$ gives $f(\widehat{u}) \le f(u^0)$ and positive support contained in S. The corresponding original-scale coefficients $\widehat{a}_j = \widehat{u}_j/\|d_j\|_2$ have the same reconstruction.*

Proof. For any eigenpair (λ, v) of H, choose i with $|v_i| = \max_j |v_j| > 0$. Then $|\lambda||v_i| \le \sum_j |H_{ij}||v_j| \le L|v_i|$. The symmetric positive-semidefinite matrix H therefore has spectral norm at most L, and $\|\nabla f(u) - \nabla f(v)\|_2 = \|H(u - v)\|_2 \le L\|u - v\|_2$.

The fixed-point equation $G_L(u) = 0$ is equivalent, coordinate by coordinate, to $\nabla_j f(u) = 0$ when $u_j > 0$ and $\nabla_j f(u) \ge 0$ when $u_j = 0$. These conditions imply, for every feasible v, $\nabla f(u)^\top (v - u) \ge 0$. Expanding the quadratic yields

$$f(v) - f(u) = \nabla f(u)^\top (v - u) + \tfrac{1}{2}\|A(v - u)\|_2^2 \ge 0,$$

so u is a global minimizer. Conversely, if a positive coordinate has a nonzero gradient, a sufficiently small feasible step in the opposite coordinate direction decreases f. If a zero coordinate has negative gradient, a sufficiently small positive step decreases f. Both contradict minimality and establish the fixed-point conditions.

For the last assertion, the minimum over a finite set containing u^0 cannot exceed $f(u^0)$. Nonnegative candidates indexed only by S cannot create a positive coefficient outside S. Finally, $\sum_{j\in S} d_j\widehat{a}_j = \sum_{j\in S}(d_j/\|d_j\|_2)\widehat{u}_j = A\widehat{u}$. □

The implementation retains the best feasible iterate and also checks the reconstructed squared error before returning the result. This guarantee concerns the returned iterate; acceleration does not make every intermediate iterate monotone. For the official-class experiment (E26), the reported residual is $\|G_L(\widehat{u})\|_\infty / \max(\|h\|_\infty, 10^{-12})$. Its exact zero characterizes optimality. A small nonzero value is a numerical diagnostic, not an objective-gap or coefficient-error certificate without further conditioning assumptions. The 512-step run is reported as an approximate solve; no finite-step exactness or convergence rate is claimed.

A.2 A SUFFICIENT CONDITION FOR FROZEN-READOUT RANK STABILITY

The next statement explains which additional quantities a reconstruction argument would need to control a fixed readout. For input i, let S_i be the original selected index set, A_i its unit-column decoder and u_i^0, u_i^1 the original and refitted nonnegative coefficients. A frozen affine detector has a constant term γ and an effective coefficient vector v_i on S_i: $s_i^t = \gamma + v_i^\top u_i^t$, $t \in \{0, 1\}$. For a detector fitted on original-scale coordinates J with ridge weights β_j and discovery standard deviations σ_j, these effective coefficients are

$$(v_i)_j = \frac{\beta_j}{(\sigma_j + 10^{-6})\|d_j\|_2} \quad (j \in J \cap S_i),$$

with zero entries for other support coordinates; discovery centering is absorbed into γ.

Proposition 4 (Conditional score and ordering bound)**.** *Suppose every A_i has full column rank, with smallest singular value $\rho_i > 0$, and define $r_i^t = y_i - A_i u_i^t$. Then*

$$|s_i^1 - s_i^0| \le B_i := \frac{\|v_i\|_2}{\rho_i}\big(\|r_i^0\|_2 + \|r_i^1\|_2\big). \tag{6}$$

If the initial scores are all distinct and $|s_i^0 - s_j^0| > B_i + B_j$ for every $i \neq j$, their ordering is unchanged. For fixed binary labels with at least one positive, average precision is then unchanged. A non-worsening refit also gives $B_i \le 2\|v_i\|_2\|r_i^0\|_2/\rho_i$.

Proof. Full column rank implies $\rho_i\|u_i^1 - u_i^0\|_2 \le \|A_i(u_i^1 - u_i^0)\|_2$. The right-hand side equals $\|r_i^0 - r_i^1\|_2$ and is at most $\|r_i^0\|_2 + \|r_i^1\|_2$ by the triangle inequality. Applying Cauchy–Schwarz to $v_i^\top(u_i^1 - u_i^0)$ proves Equation (6). If $s_i^0 > s_j^0$, then

$$s_i^1 - s_j^1 \ge s_i^0 - s_j^0 - |s_i^1 - s_i^0| - |s_j^1 - s_j^0| > (B_i + B_j) - B_i - B_j = 0.$$

The reversed original order is treated by swapping i, j. Thus all pairwise orders remain strict and identical. With no ties, AP depends only on the ordered binary labels, which remain unchanged. Finally, $f(u_i^1) \le f(u_i^0)$ implies $\|r_i^1\|_2 \le \|r_i^0\|_2$, giving the last bound. □

Full rank and score margins are assumptions, not measurements established for the trained dictionaries in this study. Reselecting or refitting a readout changes its coefficients and falls outside this proposition. The rank condition matters: with $A = [d, d]$ for a unit vector d, the codes $(2, 1)$ and $(1, 2)$ reconstruct the same target $3d$ exactly, but a readout of the first coordinate changes from 2 to 1. Zero reconstruction residual alone therefore cannot bound coordinate changes for singular supports.

Relation to the main-text counterexample. Proposition 2 disproves a universal guarantee for a frozen readout. Proposition 4 supplies one sufficient route to preserving that readout's ranking; the required conditioning and margins are not established for the learned supports in this study. Reselecting or refitting a readout changes the question and is evaluated empirically.

B PANEL CONSTRUCTION AND LEXICAL INVENTORIES

B.1 ARTICLES BEFORE LEXICAL SELECTION

The new panel is built from the WikiText-103 raw-v1 training split. The parser recognizes a top-level article title only when a stripped line matches a single-equals title, of the form `= title =`. Lines after a title belong to that article until the next top-level title. Candidate paragraphs have strictly more

than 200 and fewer than 4,000 characters. The parser yields 29,443 articles. It excludes 2,061 articles containing a paragraph whose exact hash matches a source segment of the exploratory panel. This is an exclusion relative to that recorded panel, not a claim of absence from the language model's pretraining data.

Before choosing members, each eligible article is assigned to one split. The UTF-8 string formed from its numeric article identifier, a zero byte and its title is hashed with SHA-256. The first digest byte modulo three maps to discovery, calibration or test. This deterministic partition acts on article identity. It is not stratified by family or selected after inspecting model scores.

Candidate vocabulary and its order are fixed in the count-probe script. Candidates are matched as whole lexical strings: neither boundary can touch an ASCII letter, digit or underscore. Countries are case-sensitive; other families use case-insensitive matching. Longer alternatives are matched first. For each split and family, the probe takes the first five unused candidates occurring in at least 15 eligible articles. An article contributes at most one count per candidate. A candidate selected in an earlier split cannot be selected again. The count probe checks that its vocabulary does not overlap the exploratory panel's selected members, without loading a model or SAE. Tables 6 and 7 give both panels' complete inventories.

B.2 Selection, snippets and duplicate checks

For each selected member, the builder retains its first occurrence in an eligible article and extracts a snippet with up to 160 preceding characters and 80 following characters. The saved target span must reproduce the matched surface form exactly. A per-member pseudorandom ordering uses seed $20272400 + 1000s + 100f + c$, where s, f, c are zero-based split, sorted-family and member indices. It chooses ten rows subject to the following exclusions:

1. an article already selected anywhere in the new panel;
2. an already selected exact paragraph, lowercased whitespace-normalized paragraph, or exact snippet;
3. an exact paragraph or snippet from the exploratory panel;
4. a paragraph within SimHash Hamming distance three of any paragraph already selected in the new panel.

The SimHash uses lowercase alphanumeric words, caps each word count at four, and uses a 64-bit BLAKE2b hash per word. Weighted bit votes determine the fingerprint, with a nonnegative vote setting a bit to one. The SimHash test is a lexical duplicate heuristic, not a semantic-duplicate guarantee. In particular, its all-pairs check is among the selected new paragraphs; it does not certify semantic independence from all earlier data. Failure to fill any ten-article quota would stop panel construction. A final shuffle within split uses seed $20272490 + s$.

Recorded quantity	Value
Parsed articles	29,443
Articles excluded for exploratory-panel overlap	2,061
Selected rows / distinct articles	1,200 / 1,200
Rows per split / family / member	400 / 50 / 10
Distinct new members / overlap with old members	120 / 0
Minimum selected-paragraph SimHash distance	7
Minimum / maximum eligible articles per selected member	15 / 1339
Member-disjoint splits	True
No old-panel article or exact-segment overlap	True

Table 4: Recorded new-panel construction checks. Each split contains 50 positive and 350 negative rows for a one-versus-rest family detector.

B.3 Token alignment and interpretation

Tokenization retains offsets and truncates snippets at 96 tokens. The activation is extracted from the last token whose span overlaps the target lexeme, at `hidden_states[6]` of Pythia-160M. Saved

snippets contain some later characters, but the model's causal mask prevents them from influencing the selected token. The archived activation tensor is 1200×768. Table 5 reports the target-token counts across all three splits, rather than assuming all words occupy one token.

Labels come from inventory membership and do not require that the surrounding text express the corresponding sense. Food terms in titles and ambiguous plant or technology terms retain their inventory labels. Country names and multiword expressions use the same target-span extraction procedure. The task therefore measures lexical-family readout at a designated occurrence. It does not use human annotation of contextual meaning.

Family	Minimum	Maximum	Mean
animals	1	3	1.480
countries	1	3	1.013
foods	1	3	1.127
instruments	1	4	2.187
plants	1	3	1.493
professions	1	3	1.187
technology	1	3	1.060
vehicles	1	3	1.333

Table 5: Number of tokenizer tokens overlapping the selected lexical target, aggregated over 150 rows per family.

Family	Discovery	Calibration	Test
animals	bear, deer, fox, mouse, snake	shark, whale, dolphin, eagle, owl	zebra, penguin, donkey, camel, leopard
countries	United States, Ireland, Sweden, Norway, Poland	Argentina, Chile, Peru, Iraq, Iran	Denmark, Finland, Netherlands, Indonesia, Malaysia
foods	bread, cheese, meat, soup, cake	sandwich, steak, bacon, cereal, egg	meal, breakfast, beef, candy, pork
instruments	organ, keyboard, piano, bass, horn	guitar, drum, harp, violin, trumpet	recorder, percussion, flute, cello, viola
plants	tree, flower, grass, rose, oak	bamboo, orchid, tomato, potato, onion	lily, maple, cedar, birch, moss
professions	manager, director, officer, worker, journalist	musician, architect, dentist, accountant, mechanic	driver, surgeon, veterinarian, therapist, politician
technology	machine, robot, algorithm, platform, cloud	digital, online, AI, smartphone, tablet	interface, monitor, printer, satellite, virtual
vehicles	car, truck, bus, train, boat	vehicle, rocket, automobile, helicopter, wagon	aircraft, bike, ferry, ambulance, tank

Table 6: Complete exploratory-panel inventory: five members per family per split and 20 source segments per member. These are the members of the panel used to select the width comparison.

Family	Discovery	Calibration	Test
animals	goat, pig, rat, frog, bee	ant, spider, butterfly, turtle, crab	squirrel, beaver, goose, crocodile, alligator
countries	Belgium, Greece, Portugal, Israel, Ukraine	Pakistan, Thailand, Vietnam, Austria, Switzerland	Singapore, Nigeria, Kenya, Morocco, Colombia
foods	pizza, salad, chocolate, butter, sugar	salt, fruit, vegetable, ice cream, burger	sausage, dinner, lunch, dessert, cream
instruments	accordion, trombone, banjo, clarinet, saxophone	oboe, synthesizer, mandolin, tambourine, harmonica	ukulele, bassoon, cornet, bugle, lute
plants	pine, wheat, corn, bean, cactus	cabbage, palm, herb, shrub, fern	barley, sunflower, daisy, lavender, lotus
professions	designer, analyst, author, editor, consultant	chef, photographer, developer, programmer, coach	carpenter, pharmacist, banker, electrician, secretary
technology	browser, laptop, chip, hardware, sensor	processor, storage, wireless, mobile, automation	scanner, semiconductor, circuit, battery, camera
vehicles	bicycle, ship, taxi, van, airplane	motorcycle, subway, tractor, jeep, tram	sedan, locomotive, yacht, spaceship, carriage

Table 7: Complete new-panel inventory: five members per family per split and ten distinct articles per member. All 120 entries are distinct across splits and disjoint from the exploratory-panel inventory.

C Readout definition and statistical analysis

C.1 Discovery-only feature selection and ridge fitting

Let $Z \in \mathbb{R}^{N\times M}$ contain the gated discovery codes and $y \in \{0,1\}^N$ the labels of one family. For each coordinate, the evaluator computes

$$c_j = \frac{\sum_i (Z_{ij} - \bar{Z}_j)(y_i - \bar{y})}{\max\{\sqrt{\sum_i (Z_{ij} - \bar{Z}_j)^2}\sqrt{\sum_i (y_i - \bar{y})^2}, 10^{-8}\}}.$$

Only strictly positive correlations are eligible. A stable descending sort selects the top three coordinates; fewer than three eligible coordinates would stop the evaluation. The code first caps candidates at 256, which does not change the top-three selection. This cap is not an extra search over three-coordinate combinations.

For the selected coordinates J, use discovery means μ_j and population standard deviations σ_j (division by N, not $N-1$). Define $V_{ij} = (Z_{ij} - \mu_j)/(\sigma_j + 10^{-6})$ and the augmented design $X = [\mathbf{1}, V]$. The fitted intercept and weights are

$$\widehat{\theta} = (X^\top X + \Lambda)^{-1} X^\top y, \qquad \Lambda = \mathrm{diag}(0, \lambda, \lambda, \lambda), \quad \lambda = 1. \tag{7}$$

This is the minimizer of the *sum* of squared errors plus the squared weight penalty; the intercept is unpenalized. The system is invertible even if selected columns are dependent: for a nonzero vector (b, w),

$$(b, w)^\top (X^\top X + \Lambda)(b, w) = \|b\mathbf{1} + Vw\|_2^2 + \lambda \|w\|_2^2 > 0.$$

If $w \neq 0$, the second term is positive; if $w = 0$, then $b \neq 0$ and the first is $Nb^2 > 0$. The assumption $N > 0$ holds in both panels.

The same parameters transform and score calibration and test rows. Discovery has 800 rows in the exploratory panel and 400 in the new panel. Holding $\lambda = 1$ in the summed objective therefore corresponds to penalties $1/800$ and $1/400$ when expressed per observation. This convention is part of the repeated procedure and another reason that the panel change does not isolate lexical novelty alone.

C.2 Calibration threshold and average precision

For a family, assume $n_- > 0$ and finite calibration scores, and let its n_- negative calibration scores in descending order be $q_{(1)} \geq \cdots \geq q_{(n_-)}$ and set $a = \lfloor 0.05 n_- \rfloor$. The threshold is the next representable number above $q_{(a+1)}$. Predictions use $s \geq \tau$. At most a negatives strictly exceed $q_{(a+1)}$, so at most a can meet this threshold, including when scores are tied. Thus calibration FPR is at most $a/n_- \leq 0.05$. In the new panel, $n_- = 350$ and $a = 17$, giving an upper bound of $17/350 \approx 0.04857$. This finite calibration-set statement is not a bound on test FPR. Calibration is separate from coordinate selection and weight fitting.

AP is computed from all distinct score thresholds, independently of τ. Index equal-score groups in decreasing score order. Let W_g and T_g be cumulative example mass and positive mass through group g, and let $\Delta T_g = T_g - T_{g-1}$ with $T_0 = 0$. For total positive mass $T_G > 0$,

$$\mathrm{AP} = \sum_{g: W_g > 0} \frac{\Delta T_g}{T_G} \frac{T_g}{W_g}. \tag{8}$$

Original rows have unit mass. Bootstrap rows use nonnegative integer multiplicities. Grouping all equal scores at one threshold prevents label order within a tie from altering AP. Macro AP averages the eight family APs with equal weight. FVU instead divides total squared reconstruction error by $\sum_{i \in \mathrm{test}} \|x_i - \bar{x}_{\mathrm{test}}\|_2^2$; it uses neither family labels nor the calibration threshold.

C.3 What the three coding readouts hold fixed

Interface	Coordinates	Scaling and ridge weights	Threshold
Frozen detector	Encoder discovery set	Encoder discovery fit	Encoder calibration fit
Same-coordinate refit	Encoder discovery set	New arm's discovery fit	New arm's calibration fit
Reselect and refit (primary)	New arm's discovery set	New arm's discovery fit	New arm's calibration fit

Table 8: Each coding arm uses the same split roles. Only the first row keeps the full detector unchanged, as assumed by Proposition 4.

In every comparison, the encoder's reselect-and-refit result is the baseline. Its three interfaces coincide because they derive from the same encoder codes. Native-threshold coding is a separate diagnostic with a different activation budget, not a substitute for the 64-coordinate primary interface.

C.4 Crossed seed/member bootstrap

The bootstrap resamples saved test predictions. It never repeats discovery selection or fitting. For each replicate b, independently within each of the eight fixed families, draw five member indices with replacement from that family's five test members. If $C_{fc}^{(b)}$ is the multiplicity of member c in family f, then all ten of that member's articles receive the same weight $C_{fc}^{(b)}$. This preserves the article cluster. Evaluate each seed's eight family APs with Equation (8), including the resampled members of other families as negatives.

Next sample six seed indices $S_1^{(b)}, \ldots, S_6^{(b)}$ with replacement from the six trained seed blocks. The *same* member weights and seed indices apply to both architectures, both widths and both coding arms in the relevant paired contrast. Write $A_{a,w,s}^{(b)}$ for the resampled macro AP and $D_{w,s}^{(b)} = A_{\mathrm{nested},w,s}^{(b)} - A_{\mathrm{BTK},w,s}^{(b)}$. For example,

$$D_{16}^{(b)} = \tfrac{1}{6} \sum_{r=1}^{6} \left(A_{\mathrm{nested},16,S_r^{(b)}}^{(b)} - A_{\mathrm{BTK},16,S_r^{(b)}}^{(b)} \right), \tag{9}$$

$$W^{(b)} = \tfrac{1}{6} \sum_{r=1}^{6} \left(D_{16,S_r^{(b)}}^{(b)} - D_{8,S_r^{(b)}}^{(b)} \right). \tag{10}$$

There are 2,000 replicates. Reported intervals are the empirical 2.5th and 97.5th percentiles. The new-panel comparison uses random seed 202624, drawing members and then seed blocks inside each replicate. The coding diagnostic uses random seed 202626, first generating multinomial member-count arrays and then seed draws. The replay script preserves these schedules exactly. Their descriptive intervals condition on the observed family set, corpus, trained checkpoints and fitted readouts. Coverage for a fully rerun selection pipeline or a population of new families is not established.

Member-only intervals keep all six seeds and resample members. Seed-only intervals in the archive use a nominal Student-t calculation with five degrees of freedom. We include them in Table 16 to identify which uncertainty each summary retains; the crossed intervals are those reported in the main text. Bootstrap fractions above zero are resampling summaries, not posterior probabilities.

C.5 Decision rules and selection history

The frozen confirmation rule in Table 21 is a joint research decision rule. It is not advertised as a level-0.05 hypothesis test. Even its sign-count component alone would have probability $(\binom{6}{5} + \binom{6}{6})/2^6 = 7/64$ under an independent symmetric-sign null; the actual joint rule additionally imposes magnitude and audit conditions. No error-rate calibration for that joint rule is supplied. The post-hoc bootstrap does not replace an unmet condition.

For the coding development decision, the Matryoshka primary AP delta must have a positive mean, at least five positive seed pairs, a positive crossed- interval lower endpoint, and positive leave-one-family-out deltas. The observed 3/6 positive pairs and interval spanning zero leave that rule unmet. Here, leaving out a family removes its AP term from the macro average; its rows remain negatives for the other family detectors, and no detector is refitted. This experiment reused the already opened panel. Neither diagnostic interfaces nor individual families are promoted to primary endpoints after inspection.

D Training and coefficient-solver details

The historical training cache contains two million 768-dimensional float16 activations. Its SHA-256 is

`b8b246dc1ea337df360cf3c06fdb4b4bbf25d717af48a86d6d59638232989786`.

The recovered extraction script specifies Pythia-160M, `hidden_states[6]`, the first 1,500 characters, truncation at 384 tokens, nonpadding positions with stride four, and a cap of two million states. The tensor stores only activations; this script documents intended extraction but does not attest the exact model revision or execution history. Evaluation instead uses the last target token in article segments.

The primary local models each use 30,000 steps, batch size 1,024, learning rate 3×10^{-4}, Adam defaults $(0.9, 0.999)$ and gradient clipping at 1. They sample with replacement from the same centered cache. Local nested-prefix endpoints are $(512, 2048, 8192)$ and $(1024, 4096, 16384)$ at the two widths. A single full-dictionary batch-top-K mask is shared across prefixes; the three prefix mean-squared reconstruction losses have equal weight. The local decoder is normalized at initialization but not constrained at every update. These are the models used in the selected architecture comparison.

The official repository is `https://github.com/bartbussmann/matryoshka_sae`, commit `c7c57877bb81df67fdac611f9c31ffc10495db0b`. The exact classes are `BatchTopKSAE` and `GlobalBatchTopKMatryoshkaSAE`. In our reference runs, auxiliary top-k is 256 with penalty $1/32$; inactive-feature tracking uses a horizon of 1,954 batches. Input unit normalization is disabled. Computation uses float32 with TF32 disabled. Decoder normalization and gradient projection follow the official step order before the Adam update. For the native-threshold diagnostic, evaluation calls `encode` to avoid threshold mutation in `BatchTopKSAE.forward`. The primary coding arms use the official pre-gate encoder activations followed by contribution-top64. Checkpoints also preserve inactive counters.

Component	Historical local variants	Official-class reference
Decoder constraint	Initialization normalization only	Per-step normalization/projection
Encoder bias	Used	Official encoder path does not use stored encoder bias
Initialization	Local Kaiming setting	Official constructor initialization
Matryoshka objective	Local nested-prefix loss	Includes official bias-only term and averaging
Auxiliary loss	Local global selection; detached residual	Per-example selection; residual in graph
Native deployment	Recovery-specific saved threshold	Official saved EMA threshold

Table 9: Several implementation components change together. This is a package comparison, not a one-factor ablation. Both reference classes use the shared local training budget described in the main text.

D.1 Configuration shared across runs

Setting	Value
Model / activation dimension	Pythia-160M / 768
Hook	`hidden_states[6]`
Seed blocks	30, 31, 32, 33, 34, 35
SAE updates / batch size	30,000 / 1,024
Adam learning rate / gradient norm cap	3×10^{-4} / 1
Input transform / unit normalization	Saved training-mean subtraction / disabled
Local widths / official-class width	8,192 and 16,384 / 16,384
Training top-K / primary evaluation top-K	64 (batch) / 64 (per input)
Readout coordinates / ridge coefficient	3 / 1 (sum-loss objective)
Calibration target / classification rule	$\alpha = 0.05$ / score $\geq \theta$
Refit iterations / row batch	512 / 128
Official training and coding precision	float32; TF32 disabled

Table 10: Training and evaluation settings. A nested model has multiple prefix reconstruction objectives; equal numbers of optimizer updates are not a guarantee of equal training computation across architectures.

Training uses 30,000 draws of batches of 1,024 activations sampled with replacement: 30.72 million training examples per run from the same two-million- activation cache. The primary width test reuses existing checkpoints rather than training on either lexical panel. No discovery, calibration or test labels enter SAE training. The two-million-row cache is the shared training input, while the lexical panels are used for readout evaluation.

D.2 The implemented fixed-support solver

The following procedure describes the official-class experiment. Mathematical notation treats decoder atoms as columns; the code stores them as rows. The target subtracts both the saved training mean and the SAE decoder bias. The encoder arm selects its 64 coordinates by contribution score and retains its original coefficients. For each input, normalize selected decoder atoms and convert coefficients to contribution units before optimization.

Algorithm: accelerated projected updates with retained feasible code

1. Set $H = A^\top A$, $h = A^\top y$, $L = \max\{\max_i \sum_j |H_{ij}|, 10^{-12}\}$ and $Q(u) = u^\top H u - 2h^\top u$. Initialize $u = v = u^0$, $t = 1$ and $u_{\text{best}} = u^0$.
2. For each of 512 iterations, form $u_{\text{next}} = [v - (Hv - h)/L]_+$. If $Q(u_{\text{next}}) < Q(u_{\text{best}})$, replace u_{best}.
3. Set $t_{\text{next}} = (1 + \sqrt{1 + 4t^2})/2$ and $v = u_{\text{next}} + ((t-1)/t_{\text{next}})(u_{\text{next}} - u)$. Then set $u = u_{\text{next}}$ and $t = t_{\text{next}}$.

4. After the loop, recompute decoded squared error for the retained code and the encoder code. If the retained code has larger computed error, return the encoder code instead.
5. Compute the projected-gradient diagnostic on the *returned* coefficients. Scatter the selected coefficients into an otherwise zero M-vector and divide by decoder norms to restore the original coordinate scale for readout.

The objective Q differs from $2f$ by the constant $-\|y\|_2^2$, so it ranks candidates identically in exact arithmetic. The decoded-error comparison handles numerical discrepancies between the Gram-form objective and direct reconstruction. The evaluation asserts nonnegative finite coefficients, at most 64 positive coordinates, no expansion of the original *positive* support, and non-worsening per-row reconstruction within the recorded floating-point tolerance. The latter positive-support assertion is an empirical audit of these runs: Proposition 3 by itself only constrains coefficients to the selected index set, which could include zeros.

Batching the solver uses 128 rows. All scientific coding calculations use float32 with TF32 disabled; a separate float64 synthetic solver witness compares three 12-coordinate, 24-dimensional problems with SciPy NNLS. That witness checks the implementation on those problems, not convergence of all learned supports. Table 20 reports every learned run's returned residual and fallback count. Reconstruction guarantees and the limits of the residual diagnostic are derived in Appendix A.

E Complete architecture-comparison results

Width	Seed	Raw BTK	Raw nested	Contrib. BTK	Contrib. nested
8,192	30	0.822316	0.817202	0.819374	0.858037
8,192	31	0.759124	0.781895	0.818990	0.783830
8,192	32	0.801792	0.844043	0.801263	0.811325
8,192	33	0.863541	0.787889	0.861207	0.791167
8,192	34	0.771094	0.802257	0.776709	0.802259
8,192	35	0.814456	0.850307	0.794440	0.855481
16,384	30	0.825140	0.890711	0.832139	0.895645
16,384	31	0.784574	0.782758	0.783455	0.791454
16,384	32	0.824013	0.760858	0.804409	0.794861
16,384	33	0.869014	0.849046	0.867840	0.861446
16,384	34	0.802844	0.875509	0.805228	0.878868
16,384	35	0.796491	0.869974	0.793877	0.881205

Table 11: All new-panel absolute macro AP scores, with the same checkpoint in the raw and contribution gates. The detector is selected and fitted separately for each gate.

Gate	Seed	D_{8192}	D_{16384}	W
Raw	30	-0.005114	+0.065571	+0.070685
Raw	31	+0.022771	-0.001817	-0.024588
Raw	32	+0.042252	-0.063155	-0.105407
Raw	33	-0.075652	-0.019968	+0.055684
Raw	34	+0.031163	+0.072665	+0.041502
Raw	35	+0.035851	+0.073482	+0.037631
Contribution	30	+0.038662	+0.063507	+0.024844
Contribution	31	-0.035159	+0.007999	+0.043158
Contribution	32	+0.010062	-0.009549	-0.019610
Contribution	33	-0.070040	-0.006394	+0.063646
Contribution	34	+0.025550	+0.073640	+0.048090
Contribution	35	+0.061041	+0.087328	+0.026287

Table 12: Every paired seed block. D_M is nested-prefix minus BTK AP at width M; $W = D_{16384} - D_{8192}$.

Gate	Width	Local class	AP	Recall	FPR	Youden's J
Raw	8,192	BTK	0.8054	0.7637	0.0608	0.7030
Raw	8,192	Nested	0.8139	0.7329	0.0349	0.6980
Raw	16,384	BTK	0.8170	0.7696	0.0668	0.7027
Raw	16,384	Nested	0.8381	0.7413	0.0374	0.7039
Contrib.	8,192	BTK	0.8120	0.7729	0.0620	0.7110
Contrib.	8,192	Nested	0.8170	0.7354	0.0380	0.6974
Contrib.	16,384	BTK	0.8145	0.7642	0.0681	0.6961
Contrib.	16,384	Nested	0.8506	0.7396	0.0373	0.7023

Table 13: All aggregate new-panel endpoints, averaged across six seeds. Recall and FPR use discovery-fitted detectors with calibration thresholds; the 5% calibration target does not constrain test FPR. AP is the primary endpoint.

Family	BTK 8k	Nested 8k	BTK 16k	Nested 16k
animals	0.962728	0.876470	0.949471	0.926907
countries	1.000000	1.000000	1.000000	1.000000
foods	0.802964	0.909340	0.824824	0.937090
instruments	0.877598	0.764045	0.893420	0.792470
plants	0.660727	0.496543	0.632494	0.590565
professions	0.947844	0.951617	0.926859	0.937396
technology	0.376591	0.637974	0.515153	0.700649
vehicles	0.867524	0.900143	0.773710	0.919564

Table 14: Family AP averaged across six seeds under the primary contribution gate. The family set is fixed rather than sampled from a population of concepts.

Family	W in family	W without family
animals	+0.063694	+0.026409
countries	+0.000000	+0.035508
foods	+0.005890	+0.034666
instruments	+0.012603	+0.033707
plants	+0.122255	+0.018043
professions	+0.006765	+0.034541
technology	-0.075888	+0.046349
vehicles	+0.113234	+0.019331

Table 15: Family-specific and leave-one-family-out width contrasts under the primary contribution gate. Deleting one family does not refit any detector.

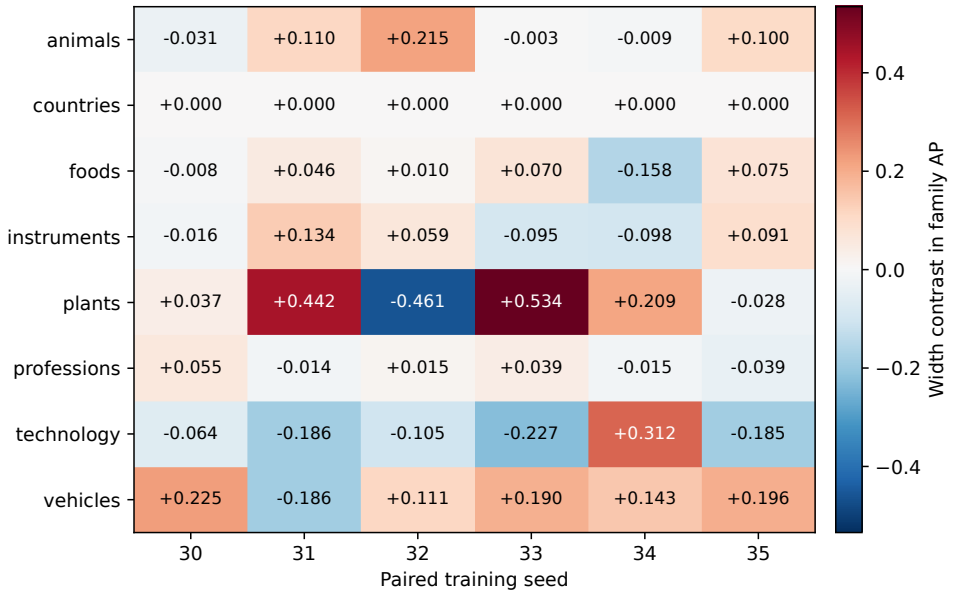


Figure 4: The primary width contrast for every family and seed block. Each cell uses the paired difference of the 16k and 8k architecture gaps. This expands the fixed-family averages into their recorded seed contributions; it is descriptive and does not supply additional independent runs.

Uncertainty summary	95% interval: D_{16384}	95% interval: W
Seed-only nominal t	[-0.00957, +0.08175]	[+0.00092, +0.06122]
Member clusters, fixed seed mean	[+0.01109, +0.06375]	[-0.00198, +0.06352]
Crossed seeds and member clusters	[-0.00648, +0.08556]	[-0.01650, +0.07891]

Table 16: Alternative conditional uncertainty summaries for the primary new-panel result. Only the crossed seed/member intervals are used in the main text; none replaces the frozen decision rule.

F ADDITIONAL CODING INTERFACES

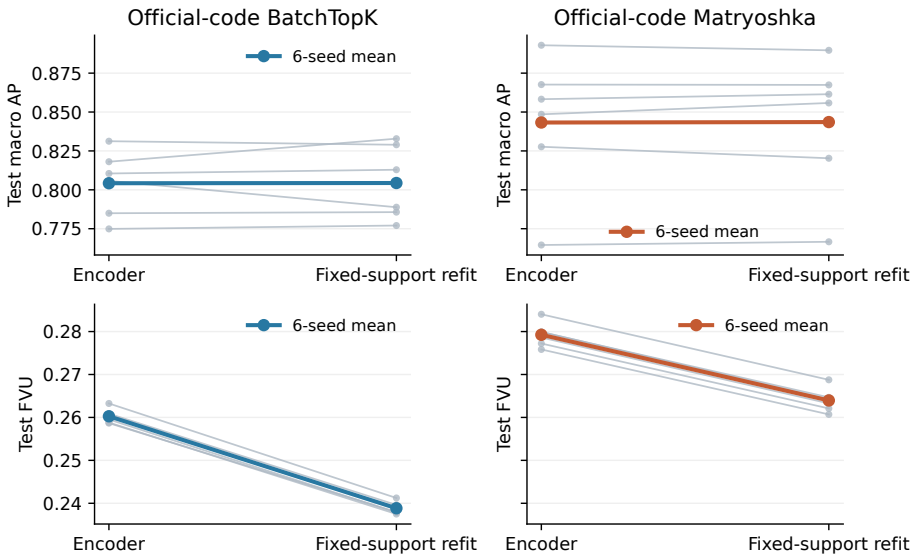


Figure 5: Absolute reconstruction and primary readout endpoints for the same six official-class runs in each architecture. Thin lines are seed pairs; thick lines are means. The primary AP readout is selected and fitted separately for each coding arm.

Class	Seed	Encoder AP	Refit AP	ΔAP
BTK	30	0.818059	0.832864	+0.014804
BTK	31	0.810499	0.812904	+0.002405
BTK	32	0.831242	0.828964	-0.002278
BTK	33	0.805740	0.788811	-0.016929
BTK	34	0.774877	0.777043	+0.002166
BTK	35	0.784950	0.785662	+0.000712
Matryoshka	30	0.858256	0.861453	+0.003197
Matryoshka	31	0.764525	0.766594	+0.002069
Matryoshka	32	0.867648	0.867430	-0.000218
Matryoshka	33	0.827673	0.820291	-0.007382
Matryoshka	34	0.848531	0.855821	+0.007290
Matryoshka	35	0.892929	0.889677	-0.003252

Table 17: Primary AP for all official-code reference runs.

For Matryoshka, fixed-support refitting changes AP by +0.003048 with the original detector frozen and +0.002219 when the same selected coordinates are refitted. The primary reselection-and-refit change is +0.000284. BatchTopK's corresponding secondary changes are -0.004322 and -0.002643. These outcomes remain distinct from the primary endpoint. Native threshold encoding gives mean AP 0.817265 for BatchTopK and 0.846331 for Matryoshka, with mean positive L_0 of 83.0 and 89.2. These larger activation budgets do not form a matched 64-coordinate comparison.

The native-threshold scores above are the original online measurements. Replaying saved float32 predictions changes the plants-family AP of one Matryoshka run by -0.0004762 in each of the same-coordinate-refit and reselection interfaces. The resulting six-seed macro-AP discrepancy is -0.00000992 in each interface. The evaluator casts scores to float32 for storage, which can change ties; the original higher-precision scores were not retained to establish the exact cause. Encoder and fixed-support-refit predictions reproduce their reported AP values to numerical precision.

The 512-step solver required zero finite-precision fallback rows across 14,400 recodings. Mean refitting times were 0.786 and 0.765 seconds per 1,200 rows on the recorded GPU; these single-run measurements are not a throughput benchmark. Training the twelve reference SAEs took 49.83 minutes in total and the evaluation took 60.73 seconds. All 43 archived output hashes match the remote manifest, including all twelve checkpoints. Recomputing primary AP from the saved prediction arrays differs from the reported values by at most 4.44×10^{-16}.

Class	Coding	Readout	AP	ΔAP	Positive
BTK	Encoder	Frozen	0.804228	+0.000000	0/6
BTK	Encoder	Same coords.	0.804228	+0.000000	0/6
BTK	Encoder	Reselect	0.804228	+0.000000	0/6
BTK	Refit	Frozen	0.799906	-0.004322	2/6
BTK	Refit	Same coords.	0.801585	-0.002643	2/6
BTK	Refit	Reselect	0.804375	+0.000147	4/6
BTK	Native	Frozen	0.810533	+0.006305	6/6
BTK	Native	Same coords.	0.810726	+0.006498	6/6
BTK	Native	Reselect	0.817265	+0.013037	6/6
Matryoshka	Encoder	Frozen	0.843260	+0.000000	0/6
Matryoshka	Encoder	Same coords.	0.843260	+0.000000	0/6
Matryoshka	Encoder	Reselect	0.843260	+0.000000	0/6
Matryoshka	Refit	Frozen	0.846308	+0.003048	4/6
Matryoshka	Refit	Same coords.	0.845479	+0.002219	4/6
Matryoshka	Refit	Reselect	0.843544	+0.000284	3/6
Matryoshka	Native	Frozen	0.848622	+0.005362	5/6
Matryoshka	Native	Same coords.	0.849013	+0.005752	5/6
Matryoshka	Native	Reselect	0.846331	+0.003071	4/6

Table 18: Every coding/readout interface, averaged across six seeds. AP changes are relative to encoder/reselect-and-refit. Native-threshold rows use a different L_0 and the original online scores, with the disclosed float32 replay discrepancy. The primary endpoint is fixed-support refit with reselection and refitting.

Family	BTK encoder	ΔAP	Mat. encoder	ΔAP
animals	0.971021	-0.006269	0.898380	-0.002668
countries	1.000000	+0.000000	1.000000	+0.000000
foods	0.838896	+0.010954	0.862120	+0.004763
instruments	0.879682	+0.007302	0.834343	+0.005818
plants	0.635458	+0.005865	0.638146	-0.006589
professions	0.940775	-0.002054	0.944696	+0.002538
technology	0.310486	+0.018196	0.636740	-0.000678
vehicles	0.857505	-0.032821	0.931658	-0.000913

Table 19: Primary family AP and its fixed-support-refit change, averaged over six seeds in each official class.

Class	Seed	Encoder FVU	Refit FVU	Mean L_0	Residual	Fallback
BTK	30	0.260861	0.239620	63.860	3.13e-04	0
BTK	31	0.258671	0.237959	63.875	2.72e-04	0
BTK	32	0.263251	0.241235	63.880	2.23e-04	0
BTK	33	0.258812	0.237497	63.830	2.58e-04	0
BTK	34	0.260471	0.238884	63.830	2.30e-04	0
BTK	35	0.259529	0.237859	63.850	2.53e-04	0
Matryoshka	30	0.278619	0.263265	63.965	2.52e-04	0
Matryoshka	31	0.275833	0.260701	63.958	2.39e-04	0
Matryoshka	32	0.277217	0.262093	63.953	3.19e-04	0
Matryoshka	33	0.279870	0.264671	63.953	2.90e-04	0
Matryoshka	34	0.284026	0.268760	63.955	2.94e-04	0
Matryoshka	35	0.279932	0.264226	63.975	2.99e-04	0

Table 20: Per-run reconstruction and numerical diagnostics. Residual is the maximum relative projected-gradient diagnostic over 1,200 returned codes; fallback counts also cover all three splits. Mean L_0 is test-only.

G SELECTION HISTORY AND ARTIFACT MAP

The exploratory member panel and width comparisons were inspected before the new-panel protocol was frozen. A 12,288-width exploratory condition was added under an adaptive rule; it is not a separate confirmation. The decoder-scale gate diagnostic also preceded the new-panel primary gate choice. After the new-panel outcome, coefficient diagnostics reused that now-open panel. The official-code reference fixed its primary endpoint before scientific evaluation but remained a development experiment. These stages should not be pooled into an independent confirmation count.

The archive retains the original identifiers E21–E26 to connect the paper to its data. E21/E22 contain exploratory results, E23 the scale-ranking diagnostic, E24 the new-panel comparison, E25 local fixed-dictionary diagnostics, and E26 the official-code reference. The accompanying package contains E21–E24 and E26 summaries, E24/E26 predictions, panel manifests and frozen protocols. It omits the model weights, training cache and E25 development artifacts. Its asset script checks that historical and new-panel checkpoint hashes match and emits the tables and figures without training or fitting a detector. The CPU replay recomputes primary AP, paired differences and the archived crossed-bootstrap intervals. FVU is re-aggregated from recorded run-level measurements; recomputing it from activations requires the omitted models and cache. Local directory prefixes in the packaged metadata are anonymized, and a manifest records both the original and packaged file hashes.

G.1 FROM APPENDIX ENTRIES TO SAVED EVIDENCE

The supplementary `implementation_reference/` directory contains snapshots of the actual panel builder, detector, gate, coefficient solver, official-class training wrapper and uncertainty scripts. Its separate manifest records source and packaged hashes. This is code for inspection: the original execution layout and omitted model/cache dependencies are required to run those entry points. The portable CPU replay is instead `replay.py` at the package root.

The two asset generators have no access to models or activations. `build_assets.py` renders the main comparisons and verifies checkpoint identity across panels. `build_appendix.py` expands recorded summaries into the complete inventories, family/seed results, readout interfaces and solver tables, and checks that the family-by-seed width contrasts average back to their archived family values. Its source hashes and generated paths are recorded in a separate manifest. The extra frequency-probe metadata retain the candidate order and article counts, including candidates not selected for the final panel.

The analysis archive includes all inputs required by these CPU scripts. The LaTeX archive includes the rendered tables and vector figures, so building the PDF requires neither Python nor a GPU. The mathematical appendix is self-contained; its statements give supporting properties under the stated assumptions rather than new evidence for the empirical comparisons.

H COMPLETE PROTOCOL AND PRIOR-WORK COMPARISON

H.1 THE FULL NEW-PANEL CONFIRMATION RULE

All conditions below were fixed before evaluating the new panel. The main text reports the paired AP outcomes; this table preserves the complete joint rule, including identity and activation-budget checks.

Frozen confirmation component	Observed	Outcome
Mean width contrast $\bar{W} \geq 0.03$	0.031069	Met
At least five positive W_s	5/6	Met
Every leave-one-family-out $\bar{W} > 0$	min 0.018043	Met
Mean larger-width gap $\bar{D}_{16384} \geq 0.03$	0.036089	Met
At least five positive $D_{16384,s}$	4/6	Unmet
Exact positive input budget in every row	64	Met
Recorded identity/hash checks	Passed	Met

Table 21: All components of the locally frozen joint rule. The unfulfilled larger-width seed-consistency condition leaves the comparison `not_confirmed`, despite the positive width-contrast estimate.

H.2 EARLIER GATE DIAGNOSTIC

An exploratory-panel diagnostic at width 12,288 changed the architecture gap from +0.0725 under raw top64 to +0.0767 under contribution top64. This cell preceded the new-panel comparison and helped select its primary gate. It is not an independent replication. The earlier width-12,288 choice was adaptive, as documented in Appendix G.

H.3 RELATION TO PRIOR STUDIES

Study	Main comparison or intervention	Relation to the present test
SAEBench (Karvonen et al., 2025)	Architectures across multiple quality metrics	Motivates keeping reconstruction and readout endpoints separate.
Sparse-probe utility (Kantamneni et al., 2025)	SAE and non-SAE predictors under data and distribution changes	Addresses predictor usefulness; our test compares two SAE implementations.
Answerability (Heindrich et al., 2025)	Readout performance across datasets, including transfer and dataset-specific selection	Studies feature generalization; we repeat an architecture comparison on new members.
Benchmark reliability (Chanin, 2026)	Evaluation reseeding, synthetic ground truth and training trajectories	Studies metric reliability; we retain a selected comparison's new-panel decision rule.
Inference optimization (O'Neill et al., 2025; Sun et al., 2026)	Changing inference for a learned dictionary	Motivates the coding diagnostic; our solve retains the encoder's support.
This study	A selected width comparison on a new inventory; then a coding diagnostic on the opened panel	Fixed checkpoints in the first test; official classes and paired endpoints in the second.

Table 22: Closest prior work and its relation to the present replication and coding tests.

Gao et al. (2025) also optimize nonzero activations under a fixed sparsity mask and a positivity constraint, comparing reconstruction and next-token cross-entropy (their Section 5.1). Our coding test uses that class of diagnostic with a specified three-coordinate lexical readout and paired training-seed/member uncertainty. The coefficient optimization itself is established prior work.

I EXPLORATORY RECONSTRUCTION–READOUT ASSOCIATION

Figure 6 pairs each run's relative test FVU reduction with its primary AP change. This post-hoc view uses six observations per architecture. Pearson correlations are -0.290 for BatchTopK and -0.344 for Matryoshka; Spearman correlations are -0.600 and -0.200. The quantities are computed within each architecture because pooling the two classes would mix between-class differences with variation across seeds.

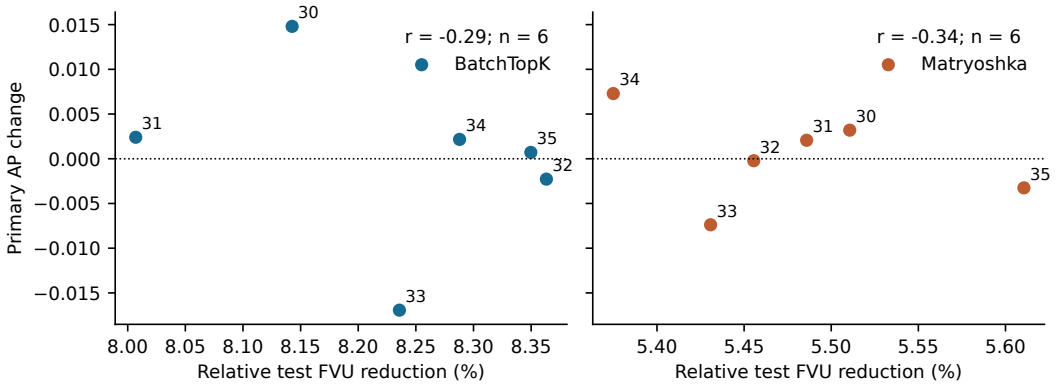


Figure 6: Descriptive associations within each official SAE class. Each point is one seed, labeled 30–35. AP uses the primary reselected/refitted readout; FVU reduction is computed relative to that run's encoder baseline. Six seeds provide six observations, not twelve independent observations of one common correlation.

The reconstruction reductions occupy a narrow range: 8.007–8.363% for BatchTopK and 5.375–5.611% for Matryoshka. Removing one seed changes the Pearson coefficient to values between -0.482 and -0.099 for BatchTopK and between -0.849 and $+0.133$ for Matryoshka. These statistics describe the recorded runs; they do not establish a zero population correlation or an effect of reconstruction quality on AP. In particular, these seed-level scatters cannot substitute for varying the optimization strength within a fixed checkpoint and evaluating the resulting codes with a fixed readout.