AN IDENTIFIABILITY HIERARCHY FOR MECHANISTIC INTERPRETABILITY: INTERPRETABILITY CLAIMS LIVE AT DIFFERENT LEVELS OF IDENTIFIABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretability methods such as sparse autoencoders (SAEs), probes, and steering vectors name units inside a network and claim that those units are useful abstractions. Every such claim presupposes that the unit is identifiable from the activation. We introduce a five-level identifiability framework: three structural levels defined on the code (coefficient, support, and subspace identifiability) and two functional levels defined on the downstream model (behavioral and causal identifiability). We test it on SAEs with one question: is the code the encoder produces the only good sparse code for that activation? Running classical sparse coders on the same dictionary, we find alternatives that overlap with the encoder's code at Jaccard ≈ 0.5—roughly a third of each code's atoms are absent from the other—while reconstructing the activation as well and the model's predictions slightly better. Support identifiability fails while subspace and behavioral identifiability hold. The codes agree on the atoms with large coefficients and disagree on the many small ones, and the disagreement grows as the SAE is made less sparse, in a regime where the classical conditions for unique sparse recovery do not hold. Coefficient magnitude is the best predictor of how much the model's output changes when an atom is removed; for small atoms it loses significance, and agreement among sparse coders predicts the effect instead. Identifiability, on this view, is not a property an interpretability method has or lacks. It is a level at which a given claim holds—one that a method reaches on a given model and data—and the level at which that claim should be tested.

1 INTRODUCTION

Mechanistic interpretability seeks to reverse-engineer the internal computations of neural networks by identifying functional units within them and assigning meaning to those units: linear probes for decision boundaries (Alain & Bengio, 2016); steering vectors for behavior-modifying directions (Subramani et al., 2019; 2022); and feature-level decompositions via sparse autoencoders (SAEs) (Huben et al., 2024). These methods share a move from continuous activations to named units: the interpretability claim—"this unit means X"—propagates through the model to explain behavior. SAEs, in particular, have become a dominant feature-level method (Gao et al., 2024). The standard cataloging and naming workflow (identify which dictionary atom responds to a target concept, label it, use the label downstream) rests on an implicit assumption: that the sparse code an encoder produces is uniquely determined by the activations it processes. Distinct inputs almost surely induce distinct hidden activations (Nikolaou et al., 2025); whether a hidden activation uniquely determines its sparse code is a separate question. If multiple valid codes exist for the same activation, then "atom 7493 is active" describes the encoder's choice among alternatives, and not a property of the activation itself.

Sparse coding theory already provides us with the knowledge that this kind of uniqueness can indeed fail (Donoho & Elad, 2003). The underlying ℓ_0 problem is non-convex and admits multiple local minima in general (Lu & Zhang, 2013). Classical guarantees of unique recovery only hold under stringent dictionary conditions. On the dictionaries we study, these conditions fail on every activation, even when evaluated locally on each activation's active support. But it is also a recurring lesson in deep learning that worst-case theoretical obstructions often fail to "bite" in practice (Zhang et al.,

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.

2017). Whether SAE inference inherits classical sparse coding non-uniqueness, or escapes it on the dictionaries practitioners actually use, is therefore an empirical question.

A growing body of work has emerged investigating SAE identifiability from complementary angles: theoretical conditions for identifiable SAEs at training time (Cui et al., 2026), formal proofs that linear identifiability is insufficient under concept-shift frameworks (Joshi et al., 2026), and empirical studies of dictionary-learning failures in synthetic well-conditioned settings (Barin-Pacela et al., 2026). On the question of whether the encoder's inference on a fixed dictionary is unique, recent work has shown it is not (Carbo & Nalisnick, 2026): alternative sparse coding algorithms applied to the same activations and the same dictionary recover codes that reconstruct comparably or better while activating substantially different features, even though the codes are linearly equivalent. We build on this finding with a formal hierarchy of identifiability levels, a mechanism that locates the disagreement in the low-magnitude tail, an analysis of the local dictionary geometry that predicts it, causal tests of its consequences, and an extension to sparse probes. Related work is discussed further in App. H.

This dissociation points at something that the standard identifiability literature does not fully capture: identifiability not as a binary property, but as a hierarchy of levels. We therefore organize the paper around five levels of agreement between two codes for the same activation: three structural levels that form a chain (coefficient, support, and subspace identifiability), and two functional levels defined on the downstream model rather than on the code vector (behavioral and causal identifiability). Which one a method requires depends on the interpretability claim being made. The typical "catalog-and-name" workflow operates at the support level; classical representation learning theory delivers only subspace identifiability; steering and ablation operate at the causal level.

Contributions. We use the hierarchy as a lens on SAE inference on real language-model activations, and find that it does not merely locate the failure but gives it structure.

- **A hierarchy of identifiability levels** mapping each interpretability claim to the level it requires.
- **Subspace and behavioral identifiability hold (Section 3).** Alternative codes are linearly predictable from the encoder's code, more so than from the activation itself, and are at least as good for the model.
- **Support identifiability fails, graded by magnitude (Section 3).** Alternative codes agree with the encoder on the high-magnitude atoms and disagree on a low-magnitude tail that grows as the SAE is made less sparse.
- **Local geometry predicts where support fails (Section 4).** Support size and redundancy predict which activations are contested; coefficient magnitude predicts which atoms are retained.
- **Agreement carries causal information in the tail** (Section 5). Where feature magnitude stops predicting causal effect, agreement among principled sparse coders still does.
- **The hierarchy extends to sparse probes** (App. G), where the label supplies the external ground truth the SAE setting lacks: learners at matched sparsity overlap at Jaccard 0.20 yet agree on 96% of held-out predictions, and decision trees on disjoint features compute the same function.

2 IDENTIFIABILITY LEVELS FOR SAE INFERENCE

2.1 SETUP AND NOTATION

A trained SAE with frozen decoder $W_{\text{dec}} \in \mathbb{R}^{m \times d}$ (rows are unit-norm atoms $d_1, \ldots, d_m$) and bias b_{dec} associates each activation $x \in \mathbb{R}^d$ with a sparse code $z \in \mathbb{R}^m$ such that $x \approx W_{\text{dec}}^\top z + b_{\text{dec}}$. We use *code* throughout to refer both to the encoder's output and to codes produced by alternative sparse coding algorithms operating on the same dictionary and the same activation. The *support* of a code is the set $S(z) = \{i : z_i \neq 0\}$. Two inference maps $z(\cdot), z'(\cdot)$ are *identifiable up to a class $\mathcal{T}$ of transformations* if $z'(x) = T(z(x))$ for all x, for some $T \in \mathcal{T}$; smaller classes correspond to stronger identifiability. The transformation is fixed across activations, so per-activation statistics test

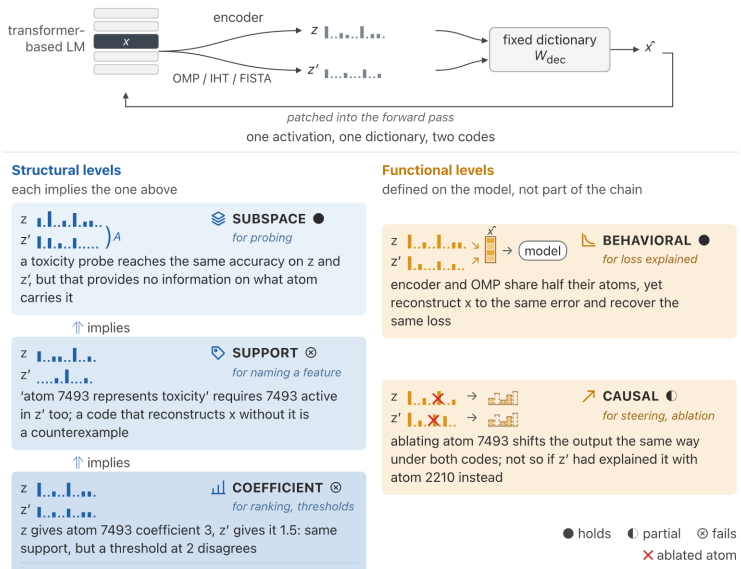

Figure 1: Five levels of identifiability for a sparse code. Top: one activation x, one fixed dictionary W_{dec}, two codes—the encoder's z and an alternative z'—each reconstructing x and patched back into the model. Left: three structural levels form a chain, each implying the one above (§2.2); right: two functional levels are defined on the downstream model (§2.3). Labels give the claim each level supports and its verdict for GemmaScope SAEs (§§3–5).

a necessary condition; failing it is a valid rejection, while passing it is supportive but not conclusive. And both maps share one fixed dictionary, so a permutation is admissible only between near-duplicate atoms; on a dictionary without duplicates "up to permutation" means equality. We therefore take permutations to be the identity, and test the near-duplicate case directly.

2.2 The structural chain

We define three levels of structural identifiability, in order of increasing strength. Structural levels compare codes as vectors without reference to the base model.

Subspace identifiability. $\mathcal{T} = \{A\}$, invertible linear maps: $z'(x) = A\,z(x)$. Codes at this level encode the same information content in possibly linearly transformed coordinates. This is the level addressed by classical representation-learning theory: Roeder et al. (2021) prove that learned representations across training runs are linearly identifiable, meaning two models trained on the same data converge to representations equal up to an invertible linear map. Probing cannot distinguish codes at this level—a linear probe trained to detect toxicity reaches approximately the same accuracy read off either code, because the information is present in both—and this is why probing accuracy cannot license a claim about which atom would carry that information. For general representation learning this is a positive result; for SAE-based mechanistic interpretability it is *insufficient*, since interpretability claims are not about information subspaces but about specific directions within them.

Support identifiability. Two codes are *support-identifiable* if equal up to a permutation and a fixed positive diagonal scaling: $z'(x) = DP\,z(x)$. A consequence of this is $S(z'(x)) = P\,S(z(x))$ for all x: the two maps activate the same set of dictionary atoms, with magnitudes that differ by a fixed factor per atom. This is the level at which the catalog-and-name workflow operates. For example, when a researcher reports that "feature 7493 represents toxicity," they claim that atom 7493 carries

part of the activation; a code that reconstructs the activation without it is a counterexample to that claim.

Coefficient identifiability. $\mathcal{T} = \{P\}$, permutation matrices: $z'(x) = P\,z(x)$, meaning the two maps assign the same coefficient to the same atom. The gap from the previous level is small but not empty, since anything that reads magnitudes rather than membership—rankings, thresholds, attribution weights, steering strengths—needs it. For example, if the encoder assigns atom 7493 a coefficient of 3 and another code assigns it 1.5, both agree the feature is present, but a threshold at 2 counts it as active under one code and not the other. This is the strongest structural level, but it is rarely achievable in overcomplete dictionaries.

The chain. A permutation is a positive monomial matrix with $D = I$, so $\{P\} \subset \{DP\}$. A positive monomial matrix DP, the product of a positive diagonal matrix and a permutation matrix, is invertible, so the support-level transformation class is a subset of the invertible linear maps defining subspace identifiability. Since identifiability up to a class is witnessed by a single element of it, identifiability at any level implies identifiability at every weaker level: $\{P\} \subset \{DP\} \subset \{\text{invertible } A\}$, so coefficient $\implies$ support $\implies$ subspace. Each class is a group, so each level is an equivalence relation on inference maps and comparisons chain across methods. The reverse implications fail: on an overcomplete dictionary with $m > d + k$ one can construct an invertible A such that Az reconstructs identically but has strictly larger support, so the two are not support-identifiable (Prop. 1, App. A). The constructed code is dense, so the proposition establishes the logical gap; that sparse codes on real SAEs exhibit it is an empirical finding. This asymmetry is what Section 3 finds: cross-validated R^2 of 0.96–0.98 (subspace identifiable) alongside Jaccard of 0.42–0.58 (not support identifiable).

2.3 Two functional levels

Two further notions of agreement are defined on the downstream model rather than on the code vector. Both are implied by coefficient identifiability but by no weaker level, and can otherwise be treated as orthogonal to the chain above.

Behavioral identifiability. Two codes are *behaviorally identifiable* if the downstream model cannot tell their reconstructions apart: patching $\hat{x} = W_{\text{dec}}^{\top} z + b_{\text{dec}}$ or $\hat{x}' = W_{\text{dec}}^{\top} z' + b_{\text{dec}}$ into the forward pass yields the same output distribution, up to a tolerance set by the reconstruction-error floor. For a frozen base model, behaviorally identifiable codes are therefore interchangeable. Exact equality would be too strong: on a dictionary without exact linear dependencies, two codes with fewer than $(d + 1)/2$ active atoms each reconstruct identically only if they are equal (App. A). The non-uniqueness we study is therefore a property of approximate sparse coding. Codes with largely disjoint supports can reconstruct the same activation to comparable error, so behavioral identifiability is logically independent of support and subspace identifiability. This is the regime our results live in—on a typical layer-20 activation the encoder and OMP largely agree on the ten largest atoms (top-10 Jaccard ≈ 0.8) but far less on the remaining support (raw Jaccard ≈ 0.5), yet reconstruct the activation to the same error and recover the same downstream loss (Section 3). To a researcher asking how much of the model the SAE explains, the two codes are essentially interchangeable.

Causal identifiability. A fifth notion does not fit the chain because it is defined on a different object. Two codes are *causally identifiable with respect to an intervention class* $\mathcal{I}$ if applying corresponding interventions produces equivalent effects on model output. For an ablation that zeros coefficient j, two codes are causally identifiable at j if zeroing it produces the same change in the next-token distribution. Like behavioral identifiability, it is implied by coefficient identifiability with $P = I$, since identical codes respond identically to any intervention, but not by support identifiability: rescaling atom j changes how much ablating it removes. Suppose the encoder explains part of an activation with atom 7493 and OMP explains the same part with atom 2210, so that the two reconstructions are equally accurate. Ablating 7493 in the encoder code removes that component and shifts the next-token distribution; ablating 7493 in the OMP code does nothing, because the atom was never active. The codes were interchangeable until someone intervened.

Why sparse coding theory does not settle the question. The encoder approximates a sparse coding problem: find the sparsest z with $W_{\text{dec}}^{\top} z + b_{\text{dec}} \approx x$. Classical theory gives uniqueness only under

dictionary conditions real SAEs do not meet: at the coherence of every GemmaScope configuration we study, the Donoho–Elad bound (Donoho & Elad, 2003) guarantees uniqueness only for codes with a single active atom, and the local exact-recovery condition fails on every activation at every shipped sparsity level (App. B, Section 4). Whether the encoder's output is unique is therefore not settled by theory in either direction. A second consideration is the amortization gap: even where the sparse coding problem has a unique solution, the encoder is a single feed-forward map producing a code in one pass, while optimal sparse inference requires iterative search (Gregor & LeCun, 2010; O'Neill et al., 2025; Beck & Teboulle, 2009). Algorithms that directly optimize the objective on each activation—orthogonal matching pursuit, iterative hard thresholding, FISTA—operate without this constraint, so running them on the encoder's own dictionary isolates the contribution of the inference procedure (amortized versus iterative) to support disagreement. The gap need not manifest as a poor code: the encoder's output is itself a stationary point of the ℓ_0-constrained problem, and the iterative methods reach different stationary points of equal or better quality (App. D).

3 Is SAE inference unique?

3.1 Setup

Models and SAEs. Our main experiments use frozen GemmaScope JumpReLU SAEs (Lieberum et al., 2024) on Gemma-2-2B (Gemma Team, 2024) residual-stream activations in four configurations: layers 5, 12, and 20 at width $m = 16{,}384$ ($\sim 7\times$ overcomplete for $d = 2304$), and layer 20 at width $m = 65{,}536$ ($\sim 28\times$). Each configuration is pinned to a specific released sparsity level (nominal L_0 of 68, 82, 71, and 61 respectively). For the sparsity sweep we additionally use the layer-20, 16k SAEs released at nominal $L_0 \in \{22, 38, 139, 294\}$, and we replicate on a second model family using a Llama-Scope TopK SAE (He et al., 2024) at layer 24 of DeepSeek-R1-Distill-Llama-8B.

Activations. We collect activations from four sources, chosen to span the distributions on which SAEs are trained and used: PILE (primary), general web text (Gao et al., 2020); FACTUAL, short factual-completion prompts (e.g. "The capital of France is") at the final token; MMLU, MMLU prompts (Hendrycks et al., 2021) at the final prompt token, whose next-token prediction is the answer letter; and CODE, permissively licensed Python files. FACTUAL is the set used by Carbo & Nalisnick (2026), and its target tokens make it the natural set for the causal experiments of Section 5. Sample sizes and sampling rules are in App. C. The encoder's mean sparsity varies across sources (56–95 atoms per activation) and is reported with every result. The base model runs in float32; all codes are stored in float32.

Comparison protocol. Each activation is processed by the encoder to obtain z_{enc} and by each method below to obtain an alternative code on the same fixed dictionary. Every comparison method is constrained to the encoder's per-activation sparsity, $\|z_{\text{alt}}\|_0 = \|z_{\text{enc}}\|_0 = k$, so support comparisons are between sets of equal size, and to non-negative coefficients, matching the JumpReLU encoder. Decoder rows are unit-norm as released. All randomness is seeded per activation, so no method's output depends on batch composition or on the order in which methods run.

3.2 Sparse coding methods

We compare the encoder against three sparse coding algorithms: greedy selection, ℓ_0-projected gradient descent, and convex ℓ_1 relaxation. Each targets the problem the encoder approximates, a non-negative k-sparse least-squares fit to x on the frozen dictionary, without the amortization constraint of Section 2. Orthogonal Matching Pursuit (OMP) is a greedy selection method that, at each iteration, adds to the active support the dictionary atom most correlated with the current residual, then re-fits coefficients on the active set by non-negative least squares (Pati et al., 1993). Iterative Hard Thresholding (IHT) alternates a gradient step on the reconstruction loss with projection onto the non-negative top-k entries (Blumensath & Davies, 2008); we run it from a random non-negative k-sparse vector and from the encoder's output, which tests whether z_{enc} is a stationary point of the ℓ_0-constrained dynamics. FISTA (Beck & Teboulle, 2009) solves the convex non-negative Lasso by accelerated proximal gradient. Its role is to rule out non-convex optimizer pathology as the source of disagreement: the Lasso has a unique optimum whenever the active atoms are in general position (Tibshirani, 2013). Because a single regularization strength cannot produce supports of size

Table 1: SAE inference is non-unique. Alternative sparse codes on the encoder's own dictionary reconstruct comparably or better (Reconstruction, MSE relative to the encoder) while selecting different atoms (Jaccard), yet agree on the high-magnitude atoms (top-10 Jaccard), on shared coefficients (Pearson on $S_\cap$), and at the subspace level (cross-validated bidirectional R^2 at $p = 50$; linear CKA). PILE, $n = 2000$; chance Jaccard ≈ 0.003. IHT-random's reconstruction ratio of 1.24 at 65k is a worse stationary point, not a convergence failure.

Config	Method	Recon	Jaccard	Top-10 J	Coef ρ	CV R^2	CKA
L5/16k	OMP	0.875	0.506	0.814	0.976	0.964	0.950
	IHT	0.985	0.449	0.810	0.970	0.969	0.954
	FISTA	0.961	0.541	0.828	0.974	0.969	0.955
L12/16k	OMP	0.867	0.474	0.804	0.972	0.978	0.999
	IHT	0.982	0.456	0.802	0.963	0.971	0.998
	FISTA	0.956	0.551	0.832	0.974	0.974	0.999
L20/16k	OMP	0.877	0.506	0.809	0.975	0.977	0.992
	IHT	0.986	0.497	0.801	0.968	0.970	0.992
	FISTA	0.946	0.576	0.822	0.975	0.973	0.993
L20/65k	OMP	0.831	0.440	0.772	0.967	0.972	0.992
	IHT	1.244	0.420	0.762	0.954	0.959	0.991
	FISTA	0.941	0.501	0.791	0.967	0.962	0.990

k across activations whose k varies severalfold, we calibrate λ per activation, truncate to the top k atoms, and re-fit coefficients on that support by non-negative least squares. Headline numbers use OMP-free, IHT-random, and FISTA-topk; variants, step sizes, stopping rules, and calibration are in App. C.

3.3 Support identifiability fails; subspace identifiability holds

Table 1 reports, for each configuration and method on the PILE set, the statistics of Section 2. The pattern is consistent across layers, widths, and methods. Alternative codes reconstruct as well as or better than the encoder—OMP by 12–17%—while overlapping with its support at Jaccard 0.42–0.58. Yet the codes are linearly predictable from one another (cross-validated R^2 of 0.96–0.98, CKA 0.95–0.999), and on the atoms two codes do share, their coefficients agree at correlation 0.95–0.98. Support identifiability fails; subspace identifiability holds, and on the shared support the coefficients agree.

The disagreement is in the low-magnitude tail. Raw Jaccard counts every atom equally. When we restrict to the ten largest coefficients, agreement rises to 0.76–0.83. The methods agree about which high-magnitude atoms are active and about how much, but they disagree about the long list of small ones.

The convex relaxation agrees most, and still misses roughly 30% of the support. FISTA is the method closest to the encoder on every configuration (0.50–0.58). Since its objective is convex and its support is well-defined, this disagreement cannot be attributed to non-convex optimizer pathology.

The tail grows with sparsity level. If disagreement lives in the low-magnitude tail, it should scale with how much tail an SAE has. GemmaScope provides the layer-20, 16k dictionary at five sparsity levels, letting us vary k by a factor of 14 with everything else fixed. Figure 2 reports raw and top-10 Jaccard at each level. Raw Jaccard falls monotonically from 0.62 to 0.37 as k grows from 23 to 323; top-10 Jaccard *rises* from 0.73 to 0.85. A looser SAE does not lose agreement about the atoms that carry magnitude; it gains a larger ambiguous fringe.

Alternative codes are at least as good for the model. Behavioral identifiability asks whether the downstream model can tell the codes apart. We splice each method's reconstruction into the residual stream at layer 20 across all positions of 200 held-out sequences and measure next-token cross-entropy, reporting the fraction of loss recovered relative to zero-ablating the layer. Every alternative

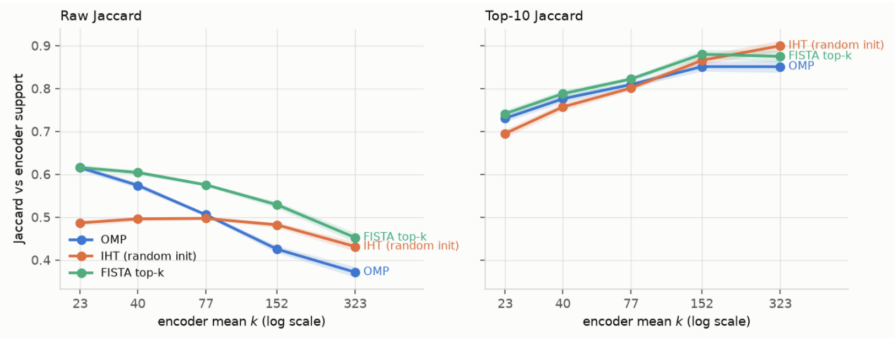

Figure 2: Sparsity sweep at layer 20, 16k, PILE. Raw (left) and top-10 (right) Jaccard between the encoder's support and each method's, across the five released sparsity levels (encoder mean k from 23 to 323, log scale). Raw agreement falls as the SAE is made less sparse while agreement on the ten largest atoms rises: a looser SAE gains an ambiguous low-magnitude fringe without losing the high-magnitude core. Per-level values and n are in App. D.

recovers *more* loss than the encoder (0.985–0.989 against the encoder's 0.984; App. D). A code that swaps out a third of the encoder's atoms is not a degraded substitute; for the model it is a slightly better one. Support non-uniqueness is, behaviorally, a genuine ambiguity.

Robustness. The dissociation has the same shape on every activation source and on a Llama-Scope TopK SAE at layer 24 of DeepSeek-R1-Distill-Llama-8B (raw Jaccard 0.43–0.52, top-10 0.74–0.76; App. D); it is not specific to one model, SAE architecture, or input distribution.

4 Geometry predicts non-uniqueness

Non-uniqueness concentrates in low-magnitude atoms. We now ask whether properties of the dictionary, evaluated on each activation's active support, predict where it occurs. No recovery guarantee from sparse coding theory holds on any activation of any shipped SAE; the active supports have a coherent core and a near-orthogonal fringe; and sparsity together with support redundancy explains 26–60% of per-activation disagreement, while per-feature retention is predicted almost entirely by coefficient magnitude.

The operating regime is past every recovery guarantee. The global coherence bound is vacuous because a single near-parallel pair anywhere in the dictionary defeats it (App. B). Tropp's exact recovery condition (Tropp, 2004) is its per-support analogue. For a support S with atom matrix D_S and Gram matrix $G_S = D_S D_S^\top$, the vector $G_S^{-1} D_S d_j$ is the least-squares recipe for building an inactive atom d_j out of the active ones, and

$$\mathrm{ERC}(S) = \max_{j \notin S} \left\| G_S^{-1} D_S d_j \right\|_1$$

is the largest coefficient mass any inactive atom needs. When $\mathrm{ERC}(S) < 1$ no inactive atom can masquerade as a combination of active ones, and OMP and basis pursuit recover S exactly. Since only S and its competitors enter, near-parallel pairs elsewhere in the dictionary do not affect it.

For the encoder's support, the condition fails everywhere. $\mathrm{ERC} \geq 1$ on 100% of activations in every configuration and input distribution (mean 2.7). It holds at GemmaScope's sparsest released level and grows as the SAE is made less sparse (App. E). The condition is sufficient rather than necessary: $\mathrm{ERC} \geq 1$ does not show that the sparse code is non-unique, it's simply showing us that there's no theoretical guarantee that it is unique. And this is the regime SAEs operate in, where the guarantee is absent.

The active support has a coherent core and an orthogonal fringe. The recovery condition fails because active atoms are not orthogonal, and random matrix theory says by how much. The eigenvalue variance of the active-support Gram matrix G_S equals the mean squared pairwise cosine among

co-firing atoms (up to the factor $k-1$), and the Marchenko–Pastur law gives its value for k random directions in $\mathbb{R}^d$, so the observed variance is measured in parameter-free units of chance (App. E). Across configurations it is 3.7–7.5$\times$ the Marchenko–Pastur value, with 24–38% of eigenvalues outside the random bulk (App. E, Table 4). A second null, random k-subsets of the released dictionary, splits this excess into two parts: the dictionary's own non-orthogonality, which dominates at layer 5, and the *selection* of which atoms co-fire, which dominates at layer 20 and at 65k width. The supports have a coherent core and a near-orthogonal fringe, and the fringe is the low-magnitude tail where the disagreement of Section 3 lives.

Predicting per-activation disagreement. We ask whether the geometry of an activation's encoder support can predict how much the alternative methods will disagree with it. For this, per activation we regress the agreement between the encoder's support and each headline method's—Jaccard $J(S_{\text{enc}}, S_M)$ and the mass-weighted Ruzicka index (Jaccard with atoms weighted by coefficient mass)—on quantities of the encoder's support alone: k and the redundancy ratio $r_{\text{eff}}/k = 1/(1+\text{Var}(\lambda))$, which separate how many atoms are active from how non-orthogonal they are and are nearly independent ($R^2 = 0.01$), while nothing about the alternative code enters. Together they explain 26–60% of per-activation disagreement (5-fold cross-validated R^2, per configuration, dataset and method); ERC, $\lambda_{\min}$, effective rank and cross-support coherence are nearly collinear with these two and add less than 0.02 (App. E).

Predicting per-feature retention. We ask, given that atom j is in the encoder's support for x_i, will method M also select it? For this we fit logistic regression with GroupKFold by activation on three predictor sets: coefficient magnitude $\log|z_{\text{enc},j}|$ alone; geometry alone (the atom's nearest-inactive similarity, ERC, $\lambda_{\min}$); and both. Magnitude alone reaches AUROC 0.83–0.87; geometry alone 0.52–0.65; both 0.85–0.88. Geometry's increment over magnitude is 0.002–0.033 (mean 0.013) and shrinks with depth—0.018–0.033 at layer 5 against 0.002–0.010 at layer 20.

5 Causal consequences

Previously we located support non-uniqueness in the low-magnitude tail, and showed that magnitude, not geometry, predicts which features an alternative method retains. We now ask what non-uniqueness costs: do features that several sparse coders agree on have different effects under intervention from features only the encoder selects? The answer is yes, but mostly because agreed features are larger. Once magnitude is accounted for, agreement carries a small increment of causal information—and that increment remains where magnitude stops being informative, in the tail.

Per-feature ablation protocol. For each (activation, feature) pair (x_i, j) with $j \in S_{\text{enc}}(x_i)$ we zero $z_{\text{enc},j}$, reconstruct, patch the reconstruction into the residual stream at the SAE layer at the last token, and measure $\text{KL}(P_{\text{enc-patch}} \,\|\, P_{\text{abl}})$ against the unablated encoder reconstruction patched at the same position. Both compared states pass through the encoder reconstruction, so the reconstruction-error floor cancels; the only difference is one zeroed coefficient. We run at layers 20 and 12 (16k), on two prompt sets: FACTUAL (the 379 prompts the model completes correctly) and MMLU (final prompt token, answer letter as target). For each, we sample 100 activations and within each 20 features: the ten largest by $|z_{\text{enc},j}|$ and ten drawn uniformly from the rest of the support, giving 2,000 pairs per (layer, prompt set). Each pair is labeled with membership in the OMP, IHT, and FISTA supports; $n_{\text{agree}} \in \{0,1,2,3\}$ counts them. Every feature in the top-10 stratum is retained by all three methods, so in that stratum agreement and magnitude are the same variable; the random stratum has all four agreement classes but does not equalize magnitude across them, so it carries the test with magnitude as a covariate (App. F).

Agreement predicts causal effect, mostly through magnitude—except in the tail. Within the random stratum, median ablation effect rises monotonically with n_{agree} on both layers: features retained by all three methods have a median KL about 5$\times$ that of features retained by none at layer 20 (3.4$\times$ at layer 12; App. F). But coefficient magnitude alone explains 24% of the variance in logKL at layer 20, and adding n_{agree} raises this by 0.013; the partial Spearman correlation between agreement and effect, controlling for magnitude, is positive and significant on both layers (App. F, Table 7). The raw contrast is real; most of it is magnitude. The exception is where the disagreement lives: restricting to the lowest magnitude tertile of the random stratum, n_{agree} remains significant on both layers, while

$\log |z_{\text{enc},j}|$ loses significance at layer 12 (tail ρ in App. F, Table 7). For high-magnitude features, magnitude tells you everything and agreement adds a little. For low-magnitude features, magnitude tells you little and agreement carries the signal. This is the practical content of the paper: the features whose identity is ambiguous are the ones for which cross-algorithm agreement is informative about causal effect. App. F reports the magnitude-adjusted results across both layers and both prompt sets: layer 20 replicates on MMLU, where the tail correlation exceeds the pooled one; layer 12 replicates on FACTUAL and not on MMLU. The layer dependence is a result: the causal information in agreement is present at layer 20 across prompt distributions and present but weaker and less consistent at layer 12.

Causal identifiability on the shared and exclusive supports. The analyses above measure the causal weight of agreement: whether agreed features have larger effects. Causal identifiability as defined in our chain asks something different: whether ablating the same atom produces the same effect under two different codes. For each atom j in both the encoder's and an alternative method's support, we ablate j under each code, patching each through its own reconstruction, and compare the two effects across all shared pairs at layer 20 on FACTUAL ($4{,}968$ pairs for OMP, $5{,}803$ for FISTA). The effects agree: Spearman $+0.93$ and $+0.92$. Because large features move the output distribution substantially under either code and so agree almost trivially, the informative test is the lowest magnitude tertile, and it holds there: $+0.88$ and $+0.80$. Shared atoms are causally interchangeable, including in the tail.

The exclusive atoms are different: the encoder's exclusive atom and OMP's nearest exclusive atom are nearly orthogonal (median cosine 0.08) and their ablation effects correlate at only $+0.57$ (App. F). Agreement on an atom implies agreement on its effect; disagreement means different atoms with partially overlapping effects.

The hierarchy beyond SAEs. The SAE setting has no external ground truth: the encoder's output and the alternatives are all codes for the same activation. A sparse probe is a sparse code for a labeled concept, so the hierarchy applies with the feature basis as the dictionary and the label as an external check that every alternative code is a good one. On MLP neurons at layer 20, across 37 concepts with text-computable labels, the dissociation of Section 3 reproduces: at $k = 32$, ℓ_1-logistic and greedy forward selection overlap at Jaccard 0.20 yet agree on 96% of held-out predictions ($\kappa = 0.96$), and each learner's logit is predicted from the other's support at cross-validated $R^2 = 0.95$ against 0.27 from a random support of the same size. Decision trees from one forest with no shared features agree at $\kappa = 0.93$–0.95: behavioral identifiability with the whole structural chain failing, which no pair of linear codes can exhibit. The full setup, eight learners, three feature bases, and two preregistered predictions that fail are in App. G.

6 CONCLUSION

Identifiability for SAE inference is not a binary property but a chain of structural levels with a functional axis. On GemmaScope SAEs, subspace and behavioral identifiability hold, support identifiability fails, and causal identifiability holds on shared atoms; the disagreement lives in a low-magnitude tail that grows as the SAE is made less sparse, and in that tail cross-algorithm agreement predicts causal effect where coefficient magnitude no longer does. Sparse probes with an external label reproduce the dissociation, and decision trees on disjoint features compute the same function, so behavioral identifiability can hold while the entire structural chain fails.

Limitations. Results are on GemmaScope, with one replication on a Llama-Scope TopK SAE; the magnitudes of reported quantities may shift with model architecture, SAE objective, and prompt distribution. Causal effects are measured under single-token, last-position ablation. And ours is an empirical characterization: Prop. 1 establishes the logical gap between levels, but we measure the non-uniqueness of sparse codes on real dictionaries rather than prove it. A natural next step is to test whether features that agreement flags as reliable give more controllable, less drift-prone steering than features it flags as unreliable.

Identifiability, on this view, is not a property an interpretability method has or lacks. It is a level at which a given claim holds—one that a method reaches on a given model and data—and the level at which that claim should be tested.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Vitória Barin-Pacela, Shruti Joshi, Isabela Camacho, Simon Lacoste-Julien, and David Klindt. Stop probing, start coding: Why linear probes and sparse autoencoders fail at compositional generalization. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 364–412. PMLR, 2026.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. doi: 10.1137/080716542.

Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Inverse Problems*, 24(3):035016, 2008.

Alessa Carbo and Eric Nalisnick. Which sparse code? identifiability failures in sae inference. In *ICLR 2026 Workshop on Scientific Methods for Understanding Deep Learning (Sci4DL)*, 2026. URL https://openreview.net/forum?id=JiyytGKbA9. Poster.

Jingyi Cui, Qi Zhang, Yifei Wang, and Yisen Wang. On the limits of sparse autoencoders: A theoretical framework and reweighted remedy. In *International Conference on Learning Representations*, 2026.

David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5): 2197–2202, 2003. doi: 10.1073/pnas.0437847100.

David L. Donoho, Michael Elad, and Vladimir N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling, 2020. URL https://arxiv.org/abs/2101.00027.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024. doi: 10.48550/arXiv.2406.04093. URL https://arxiv.org/abs/2406.04093.

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

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 399–406, 2010.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders, 2024. URL https://arxiv.org/abs/2410.20526.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, 2024.

Shruti Joshi, Andrea Dittadi, Sébastien Lachapelle, and Dhanya Sridhar. Sparse shift autoencoders for identifying concepts from large language model activations, 2026. URL https://arxiv.org/abs/2502.12179.

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, 2024. URL https://arxiv.org/abs/2408.05147.

Zhaosong Lu and Yong Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization*, 23(4):2448–2478, 2013. doi: 10.1137/100808071.

Maxime Méloux, Giada Dirupo, François Portet, and Maxime Peyrard. The dead salmons of ai interpretability, 2025. URL https://arxiv.org/abs/2512.18792.

Giorgos Nikolaou, Tommaso Mencattini, Donato Crisostomi, Andrea Santilli, Yannis Panagakis, and Emanuele Rodolà. Language models are injective and hence invertible, 2025. URL https://arxiv.org/abs/2510.15511.

Charles O'Neill, Alim Gumran, and David Klindt. Compute optimal inference and provable amortisation gap in sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46877–46896. PMLR, 2025.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. Technical Report TR 93-41, University of Maryland, College Park, MD, 1993.

Geoffrey Roeder, Luke Metz, and Durk Kingma. On linear identifiability of learned representations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9030–9039. PMLR, 2021.

Nishant Subramani, Samuel R. Bowman, and Kyunghyun Cho. Can unconditional language models recover arbitrary sentences? In *Advances in Neural Information Processing Systems*, volume 32, pp. 15232–15242, 2019.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.48.

Ryan J. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7:1456–1490, 2013. doi: 10.1214/13-EJS815.

Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

Joel A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

Zeyu Yun, Yubei Chen, Bruno A. Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: Contextualized embedding as a linear superposition of transformer factors. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 1–10, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

LLM USAGE STATEMENT

We used large language models in three ways. Coding agents (Claude Code, running Claude Fable 5.0 and Claude Opus 5) implemented and executed the code for our experimental pipeline; the authors designed the experiments, preregistered the probe predictions, and verified the code and results. Claude Fable was used to proofread the manuscript for any inconsistencies, including mathematical statements and proofs. The main figure (Figure 1) was produced with Claude Design from a brief written by the authors. All scientific claims, analyses, and text are the authors' responsibility.

A PROOF OF THE CHAIN AND DEGENERATE REGIMES

In this section, we prove the chain of Section 2, show that its reverse implications fail, and work out the regimes in which some levels coincide.

For the structural statements, we allow all permutations and consider real-valued inference maps on a common nonempty input domain. The counterexample below need not preserve nonnegativity or a fixed sparsity budget. For the functional statements, following our experimental protocol: both maps use the same fixed decoder, and coefficient identifiability is taken with $P = I$, as in the main text. We first consider exact functional agreement.

Proposition 1. *Let $\mathcal{T}_{\mathrm{coef}} = \{P\}$ be the permutation matrices, $\mathcal{T}_{\mathrm{supp}} = \{DP\}$ with D positive diagonal, and $\mathcal{T}_{\mathrm{sub}} = \mathrm{GL}(m)$. Then:*

(i) $\mathcal{T}_{\mathrm{coef}} \subset \mathcal{T}_{\mathrm{supp}} \subset \mathcal{T}_{\mathrm{sub}}$, strictly for $m \geq 2$, and each class is a group under matrix multiplication;

(ii) identifiability at any structural level implies identifiability at every weaker structural level, and each structural level is an equivalence relation on inference maps;

(iii) neither reverse structural implication holds in general. In particular, let $W_{\mathrm{dec}}^{\top} \in \mathbb{R}^{d\times m}$. Whenever $m > d + k$ and $k \geq 1$, there exist inference maps z and $z' = Az$, with A invertible and z exactly k-sparse at some input, whose reconstructions coincide at every input but which are not support-identifiable. They are therefore subspace- and behaviorally identifiable;

(iv) coefficient identifiability with $P = I$ implies behavioral and causal identifiability for corresponding interventions; support identifiability alone guarantees neither.

Proof. (i) A permutation is DP with $D = I$, so $\mathcal{T}_{\mathrm{coef}} \subseteq \mathcal{T}_{\mathrm{supp}}$. A positive monomial matrix has determinant $\pm\prod_i D_{ii} \neq 0$, so $\mathcal{T}_{\mathrm{supp}} \subseteq \mathrm{GL}(m)$. Both inclusions are strict for $m \geq 2$:

$$2I \in \mathcal{T}_{\mathrm{supp}} \setminus \mathcal{T}_{\mathrm{coef}}, \qquad I + e_1 e_2^{\top} \in \mathrm{GL}(m) \setminus \mathcal{T}_{\mathrm{supp}}.$$

The latter matrix is invertible, with inverse $I - e_1 e_2^{\top}$, but has two nonzero entries in its second column and hence is not monomial. Permutations, positive monomial matrices and $\mathrm{GL}(m)$ each contain the identity and are closed under products and inverses, so each is a group.

(ii) If $z'(x) = Tz(x)$ for all x with $T \in \mathcal{T}_1 \subseteq \mathcal{T}_2$, then the same T witnesses $\mathcal{T}_2$-identifiability. Reflexivity follows from $I \in \mathcal{T}$. For symmetry, $z' = Tz$ implies $z = T^{-1}z'$, with $T^{-1} \in \mathcal{T}$. For transitivity, if $z' = T_1 z$ and $z'' = T_2 z'$, then $z'' = T_2 T_1 z$, with $T_2 T_1 \in \mathcal{T}$. In each case, the witnessing transformation is fixed across inputs.

(iii) *Support does not imply coefficient.* Take any inference map that is nonzero at some input and set $z'(x) = 2z(x)$. The two maps are support-identifiable with $D = 2I$ and $P = I$, but are not coefficient-identifiable: a permutation preserves the Euclidean norm, whereas $\|2z(x)\| = 2\|z(x)\| \neq \|z(x)\|$ whenever $z(x) \neq 0$.

Subspace does not imply support. Write $B = W_{\mathrm{dec}}^{\top}$. Fix an input x_0 and a nonzero, exactly k-sparse code $z_0 = z(x_0)$, with support $S = S(z_0)$. Such an inference map can be chosen, for example, by taking $z(x) = z_0$ at every input.

By rank–nullity,

$$\dim\ker B = m - \mathrm{rank}(B) \geq m - d.$$

Requiring a vector in $\ker B$ to vanish on the k coordinates in S imposes at most k additional linear constraints. Therefore the subspace

$$U = \{u \in \ker B : u_i = 0 \text{ for every } i \in S\}$$

has dimension at least $m - d - k > 0$. There is consequently a nonzero $u \in U$, so that $Bu = 0$ and $\operatorname{supp}(u) \cap S = \emptyset$.

Let

$$v = \frac{z_0}{\|z_0\|^2}, \qquad A = I + uv^\top.$$

Then $v^\top z_0 = 1$ and $v^\top u = 0$. Thus A is invertible: indeed,

$$(I + uv^\top)(I - uv^\top) = I,$$

so $A^{-1} = I - uv^\top$. At the reference input,

$$Az_0 = z_0 + u.$$

Since z_0 and u have disjoint supports,

$$|S(Az_0)| = k + |\operatorname{supp}(u)| > k.$$

A permutation and positive diagonal scaling preserve the number of nonzero coordinates. Hence no positive monomial matrix maps z_0 to Az_0.

Now define $z'(x) = Az(x)$ using this same fixed A at every input. The support-cardinality mismatch at x_0 suffices to rule out support identifiability of the inference maps. Yet

$$BA = B + Buv^\top = B,$$

so

$$W_{\text{dec}}^\top z'(x) + b_{\text{dec}} = W_{\text{dec}}^\top z(x) + b_{\text{dec}}$$

for every x. The maps are therefore behaviorally identifiable, because their patched activations coincide, and subspace-identifiable by construction.

At the reference input, the support overlap is exactly

$$J\big(S(z_0), S(Az_0)\big) = \frac{k}{k + |\operatorname{supp}(u)|}.$$

Without further assumptions on the dictionary, the dimension argument does not guarantee that u is nonzero at every coordinate outside S.

The constructed alternative is not k-sparse at x_0 and need not be nonnegative. More specifically, every nonzero $u \in \ker B$ satisfies

$$|\operatorname{supp}(u)| \geq \operatorname{spark}(B).$$

For a full-spark dictionary, meaning that every set of d columns of B is linearly independent, $\operatorname{spark}(B) = d + 1$. Part (iii) therefore establishes the logical gap for unrestricted real-valued inference maps; that sparse, nonnegative codes on real dictionaries exhibit this gap approximately is the empirical finding of Section 3.

(iv) If $z'(x) = z(x)$ for all x, their reconstructions coincide. Applying the same intervention to identical codes also produces identical intervened codes and reconstructions. Their downstream output distributions and intervention effects therefore agree, establishing behavioral and causal identifiability.

To show that support identifiability alone guarantees neither, it suffices to give a decoder and downstream map for which both fail. Take a unit-norm decoder atom d_1, set $b_{\text{dec}} = 0$, and consider the constant maps

$$z(x) = e_1, \qquad z'(x) = 2e_1.$$

These maps are support-identifiable, but their reconstructions are d_1 and $2d_1$. Let the downstream output distribution be

$$F(h) = \big(\sigma(d_1^\top h),\, 1 - \sigma(d_1^\top h)\big), \qquad \sigma(t) = \frac{1}{1 + e^{-t}}.$$

Since $\sigma(1) \neq \sigma(2)$, $F(d_1) \neq F(2d_1)$, so behavioral identifiability fails.

Ablating atom 1 sends both reconstructions to 0. The respective changes in output distribution are

$$F(0) - F(d_1) \quad \text{and} \quad F(0) - F(2d_1),$$

which differ. Thus causal identifiability also fails for this ablation. The example establishes a failure of implication: support agreement does not determine whether a particular downstream model distinguishes the codes. □

Degenerate regimes. The distinctions between levels depend on the structure and values of the codes. Two cases show which distinctions disappear when only one atom is active or each coefficient carries only one bit.

One active atom. Let $z(x) = a(x)\, e_{i(x)}$ and $z'(x) = b(x)\, e_{j(x)}$ with $a, b > 0$. First consider the structural levels with permutations allowed. Suppose $z'(x) = Az(x)$ for all x with A invertible. For any atom i that fires on some x,

$$Ae_i = \frac{b(x)}{a(x)}\, e_{j(x)}.$$

The left-hand side does not depend on x, so neither does the right: $j(x)$ and $b(x)/a(x)$ are determined by i alone, and we write them as $j(i)$ and $c_i > 0$.

The map $i \mapsto j(i)$ is injective on the atoms that ever fire. Indeed, suppose for a contradiction that $j(i) = j(i')$ for distinct used atoms $i \neq i'$. Then

$$A\big(c_{i'} e_i - c_i e_{i'}\big) = c_{i'} c_i e_{j(i)} - c_i c_{i'} e_{j(i')} = 0.$$

The vector in parentheses is nonzero, contradicting invertibility of A. Thus $i \mapsto j(i)$ extends to a permutation P of $[m]$, with $Pe_i = e_{j(i)}$ on the used atoms. Taking $D_{j(i),j(i)} = c_i$ on their images and 1 elsewhere gives

$$z'(x) = DPz(x)$$

for every x.

Subspace identifiability therefore implies support identifiability in this regime, and the two structural rungs coincide when permutations are allowed. Coefficient identifiability remains stronger: the example $z' = 2z$ is still support-identifiable but not coefficient-identifiable.

For the functional levels, retain the fixed dictionary and the convention $P = I$ used in the main text. Exact equality of reconstructions requires

$$a(x)\, d_{i(x)} = b(x)\, d_{j(x)}.$$

With unit-norm, non-parallel atoms and positive coefficients, this forces $i(x) = j(x)$ and $a(x) = b(x)$. Thus exact reconstruction equality coincides with coefficient identifiability under these assumptions. Behavioral identifiability, however, only requires agreement of downstream output distributions and need not imply equality of the reconstructions.

Ablating the single active atom leaves b_{dec} under either code. Let $F_x(h)$ denote the downstream output distribution when activation h is patched into the forward pass for input x, and write

$$\hat{x} = W_{\mathrm{dec}}^{\top} z(x) + b_{\mathrm{dec}}, \qquad \hat{x}' = W_{\mathrm{dec}}^{\top} z'(x) + b_{\mathrm{dec}}.$$

For the corresponding intervention that removes each code's sole active coefficient, the changes in output are

$$\Delta(x) = F_x(b_{\mathrm{dec}}) - F_x(\hat{x}), \qquad \Delta'(x) = F_x(b_{\mathrm{dec}}) - F_x(\hat{x}').$$

Consequently,

$$\Delta(x) - \Delta'(x) = F_x(\hat{x}') - F_x(\hat{x}).$$

The two changes agree exactly if and only if the pre-intervention output distributions agree. The same equivalence holds approximately when both comparisons use the same norm and tolerance.

With one active atom, then, support and subspace identifiability coincide when permutations are allowed. Behavioral and causal identifiability coincide for the specific intervention that removes the sole active coefficient, and both are implied by coefficient identifiability with $P = I$. They do not in

general imply coefficient identifiability, because the downstream model need not distinguish different reconstructed activations.

One bit per coefficient. Suppose both inference maps take values in $\{0, c\}^m$ for the same fixed $c > 0$. If $z'(x) = DPz(x)$, then at every coordinate i where $(Pz(x))_i = c$,

$$c = z'_i(x) = D_{ii}c,$$

so $D_{ii} = 1$. Thus D acts as the identity on every coordinate that is active in $Pz(x)$ at any input. Its values on coordinates that are always zero have no effect. It follows that

$$z'(x) = Pz(x)$$

for all x, so support identifiability implies coefficient identifiability.

The structural chain therefore reduces to subspace and support = coefficient. The distinction of Section 2 arising from unequal nonzero magnitudes—rankings, thresholds, and coefficient weights—disappears within this restricted code class.

B Mutual coherence and classical uniqueness bounds

The uniqueness of a sparse decomposition under a fixed dictionary is governed by *mutual coherence*:

$$\mu(W_{\text{dec}}) = \max_{i \neq j} \frac{|d_i^\top d_j|}{\|d_i\|_2 \|d_j\|_2}.$$

The Donoho–Elad uniqueness theorem (Donoho & Elad, 2003) states that any code z satisfying

$$\|z\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu}\right)$$

is the unique sparsest decomposition of x under W_{dec}. For low-coherence dictionaries this bound permits dense codes; for high-coherence dictionaries it permits only nearly-trivial ones.

The bound is vacuous on real SAEs. For all GemmaScope configurations we study, the mutual coherence is $\mu \in [0.992, 0.997]$, computed exactly over all atom pairs (Figure 3). At $\mu = 0.992$ the bound is $\frac{1}{2}(1 + 1/0.992) \approx 1.004$, so at this coherence the Donoho–Elad bound guarantees uniqueness only for codes with a single active atom; the encoders we study activate 56–95 atoms per activation. The coherence-based uniqueness theorem therefore says nothing about real SAEs. Nor is this a matter of a few outliers on an otherwise incoherent dictionary: the typical atom pair is as orthogonal as a random pair, but the coherence distribution is heavy-tailed, and hundreds of thousands of pairs exceed the maximum coherence a random dictionary of the same shape would contain. A worst-case statistic over such a tail is the wrong instrument. The natural repair is to evaluate uniqueness conditions locally, on each activation's active support; we do so in Section 4 and find that the local sufficient condition fails as well, on every activation, at every sparsity level the SAEs are shipped at.

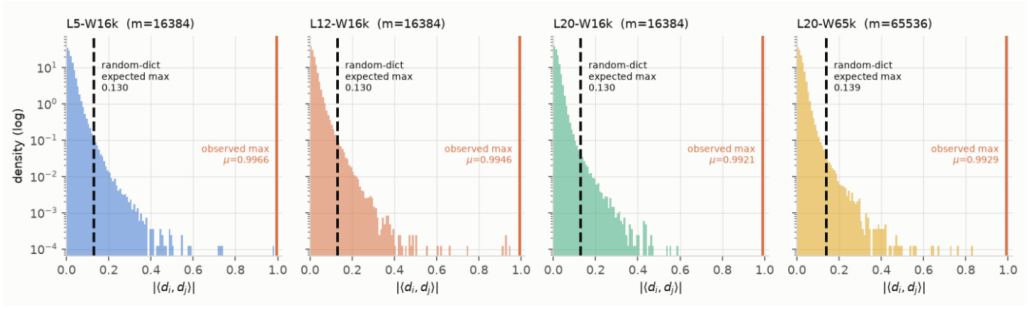

Figure 3: Distribution of pairwise $|\langle d_i, d_j \rangle|$ over all atom pairs, log density. Dashed: expected maximum for a random dictionary of the same shape. Solid: observed maximum μ.

Algorithm 1 Non-negative Orthogonal Matching Pursuit (OMP)

Require: activation x, decoder W_{dec}, bias b_{dec}, sparsity k, forbidden set F
Ensure: sparse code $z \in \mathbb{R}^m_{\geq 0}$

$\tilde{x} \leftarrow x - b_{\text{dec}}$
$r \leftarrow \tilde{x}, \quad S \leftarrow \emptyset, \quad z \leftarrow 0$
while $|S| < k$ **do**
  $i^\star \leftarrow \arg\max_{i \notin S \cup F} d_i^\top r$ ▷ Greedy selection, signed correlation
  $S \leftarrow S \cup \{i^\star\}$
  $z_S \leftarrow \arg\min_u \left\| W_{\text{dec},S}^\top u - \tilde{x} \right\|_2^2$ ▷ Least-squares refit
  if any entry of z_S is negative **then**
    $z_S \leftarrow \arg\min_{u \geq 0} \left\| W_{\text{dec},S}^\top u - \tilde{x} \right\|_2^2$ ▷ NNLS fallback, float64
  end if
  $r \leftarrow \tilde{x} - W_{\text{dec},S}^\top z_S$
end while
return z

Coherence must be computed exactly. A random sample of 10^6 pairs returns a maximum as low as 0.59, so any bound keyed to μ must not be estimated from a sample.

Exact versus approximate uniqueness. The coherence bound is a lower bound on the *spark* of the dictionary, the smallest number of linearly dependent atoms: $\text{spark}(W_{\text{dec}}^\top) \geq 1 + 1/\mu$ (Donoho & Elad, 2003). The sparsest exact representation of x is unique whenever $\|z\|_0 < \text{spark}/2$, and on a dictionary without exact linear dependencies the spark is $d + 1$, so exact representations with fewer than $(d+1)/2$ atoms are unique. This guarantee is not the relevant one: SAE codes reconstruct x only approximately, and uniqueness under approximation requires stronger conditions of the coherence or exact-recovery type (Donoho et al., 2006; Tropp, 2006), globally and locally.

C SPARSE CODING METHODS: DETAILS

Activation sources. PILE: 2,000 activations, one uniformly sampled non-BOS position per 128-token window. FACTUAL: 583 prompts at the final token; the causal experiments of Section 5 use the 379 prompts the base model completes correctly. MMLU: 500 five-shot prompts at the final prompt token; correctness labels are recorded. CODE: 500 activations, sampled as for PILE.

Each method is constrained to the encoder's per-activation sparsity and to non-negative coefficients, as described in Section 3. All three run in float32; the only exception is the fallback NNLS solver described under OMP, which runs in float64. OMP and the FISTA re-fit work with the centred activation $\tilde{x} = x - b_{\text{dec}}$, while IHT and FISTA keep the bias inside the objective; the two formulations are equivalent. The Lipschitz constant $L = \lambda_{\max}(W_{\text{dec}} W_{\text{dec}}^\top)$ used by IHT and FISTA is computed once per dictionary by power iteration on the $d \times d$ matrix $W_{\text{dec}}^\top W_{\text{dec}}$, which has the same largest eigenvalue (at most 500 iterations, tolerance 10^{-9}). All randomness is seeded per activation.

Orthogonal Matching Pursuit (OMP). A greedy selection method that, at each iteration, adds to the active support the dictionary atom most correlated with the current residual, then re-fits coefficients on the active set by non-negative least squares (Pati et al., 1993) (Algorithm 1). Selection uses the signed correlation $d_i^\top r$, since under the non-negativity constraint only atoms positively correlated with the residual can carry a positive coefficient. The re-fit first solves the unconstrained least-squares problem on the active set and falls back to an exact NNLS solver (in float64) only if a coefficient is negative; on our data the fallback is rarely taken, and we record how often per activation. Iteration terminates when $|S| = k$. OMP is deterministic. We report OMP-free ($F = \emptyset$) and three variants OMP-exclude-top-$\{1, 2, 3\}$, where F contains the encoder's top-j atoms by $|z_{\text{enc},i}|$, removed from the candidate set for the whole run; these test whether OMP's agreement with the encoder is carried by its dominant atoms. Headline numbers use OMP-free.

Algorithm 2 Non-negative Iterative Hard Thresholding (IHT)

Require: activation x, decoder W_{dec}, bias b_{dec}, sparsity k, initialization mode
Ensure: sparse code $z \in \mathbb{R}^m_{\geq 0}$
if init = encoder **then**
  $z \leftarrow z_{\text{enc}}$ ▷ Encoder initialization
else
  $z \leftarrow 0$; sample k indices uniformly without replacement; $z_j \sim \text{Uniform}(0, 0.1)$ on sampled indices ▷ Random initialization
end if
$\eta \leftarrow 5/L$
for $t = 1, \ldots, 20{,}000$ **do**
  $g \leftarrow W_{\text{dec}}\left(W_{\text{dec}}^\top z + b_{\text{dec}} - x\right)$ ▷ Gradient
  $z \leftarrow [z - \eta g]_+$ ▷ Gradient step, non-negative projection
  $z \leftarrow H_k(z)$ ▷ Keep top-k entries
  if $f(z)$ is not finite **then**
    flag as diverged; **break**
  end if
  if support unchanged for 50 steps **and** relative change in $f < 10^{-8}$ **then**
    break
  end if
end for
return z

Iterative Hard Thresholding (IHT). Alternates a gradient step on the reconstruction loss with projection onto the non-negative top-k entries (Blumensath & Davies, 2008) (Algorithm 2). The projection clamps negative entries to zero and then keeps the k largest. We use step size $5/L$, with L the largest eigenvalue of $W_{\text{dec}}W_{\text{dec}}^\top$ computed per dictionary; a fixed absolute step is not one method across configurations, and this choice minimizes the ℓ_0 objective across all four while staying well below the divergence threshold. Iteration stops when the support has been unchanged for 50 steps and the relative change in objective over those steps is below 10^{-8} in absolute value, or after $20{,}000$ steps; a non-finite objective is flagged as divergence rather than convergence. We run IHT from two initializations. From the encoder's output (IHT-encoder), which tests whether z_{enc} is a stationary point of the ℓ_0-constrained dynamics; and from a random non-negative k-sparse vector (IHT-random), seeded per activation, as a non-encoder baseline. The random start places k nonzeros at positions drawn uniformly without replacement, with values uniform on $[0, 0.1)$. Headline numbers use IHT-random.

Here $f(z) = \frac{1}{2}\|W_{\text{dec}}^\top z + b_{\text{dec}} - x\|_2^2$, and H_k retains the k largest entries and zeros the rest.

FISTA: convex ℓ_1 relaxation. FISTA (Beck & Teboulle, 2009) solves the convex non-negative Lasso

$$\min_{z \geq 0} \tfrac{1}{2}\|W_{\text{dec}}^\top z + b_{\text{dec}} - x\|_2^2 + \lambda\|z\|_1$$

by accelerated proximal gradient with step size $1/L$ (Algorithm 3). Because a single regularization strength cannot produce supports of size k across activations whose k varies severalfold, we calibrate λ per activation as the largest value at which the Lasso solution retains at least k atoms, then truncate to the top k by magnitude and re-fit coefficients on that support by non-negative least squares; the re-fit corrects the shrinkage of Lasso coefficients and improves reconstruction by 23–52%. The calibration is a bisection on $\log \lambda$ over $[10^{-2}, 10^4]$ with exactly 24 steps, each solving the Lasso from a zero start, without warm starts. Convergence is assessed on the iterate rather than the objective, which flattens much earlier: a solve stops when the relative ℓ_2 change of the iterate falls below 10^{-6}. We report FISTA-zero (raw Lasso solution from a zero start, at most 2,000 iterations), FISTA-topk (truncated and re-fit from the FISTA-zero solution; the headline variant, for support-size parity), and FISTA-enc (encoder-initialized, at most 20,000 iterations). If the Lasso solution has fewer than k nonzeros, FISTA-topk keeps all of them, so its support can fall short of k; we record the shortfall per activation. FISTA-zero and FISTA-enc agree on support at Jaccard 0.94–1.00, as convexity requires, and under per-activation calibration all three variants nearly coincide.

Algorithm 3 FISTA calibration and top-k refit

Require: activation x, decoder W_{dec}, bias b_{dec}, sparsity k

$\ell_{\mathrm{lo}} \leftarrow \log 10^{-2}, \quad \ell_{\mathrm{hi}} \leftarrow \log 10^{4}$
for $s = 1, \ldots, 24$ **do**
  $\ell \leftarrow (\ell_{\mathrm{lo}} + \ell_{\mathrm{hi}})/2$
  $z \leftarrow \textsc{Fista}(e^{\ell}, 0, 2000)$ ▷ Zero start
  if $\|z\|_0 \geq k$ **then**
    $\ell_{\mathrm{lo}} \leftarrow \ell$
  else
    $\ell_{\mathrm{hi}} \leftarrow \ell$
  end if
end for
$\lambda^{\star} \leftarrow e^{\ell_{\mathrm{lo}}}$
$z^{\mathrm{zero}} \leftarrow \textsc{Fista}(\lambda^{\star}, 0, 2000)$
$S \leftarrow$ indices of the $\min(k, \|z^{\mathrm{zero}}\|_0)$ largest entries of z^{zero}
$z_S^{\mathrm{topk}} \leftarrow \arg\min_{u \geq 0} \left\| W_{\mathrm{dec},S}^{\top} u - (x - b_{\mathrm{dec}}) \right\|_2^2, \quad z_{\bar{S}}^{\mathrm{topk}} \leftarrow 0$ ▷ NNLS refit
$z^{\mathrm{enc}} \leftarrow \textsc{Fista}(\lambda^{\star}, z_{\mathrm{enc}}, 20{,}000)$ ▷ Encoder start
return $z^{\mathrm{zero}}, z^{\mathrm{topk}}, z^{\mathrm{enc}}$

FISTA solver. For regularization λ, initialization $z^{(0)}$, and step size $\eta = 1/L$, we optimize

$$\frac{1}{2} \left\| W_{\mathrm{dec}}^{\top} z + b_{\mathrm{dec}} - x \right\|_2^2 + \lambda \|z\|_1, \qquad z \geq 0.$$

Initialize $y^{(1)} = z^{(0)}$, $\tau_1 = 1$. At iteration t,

$$g^{(t)} = W_{\mathrm{dec}} \left(W_{\mathrm{dec}}^{\top} y^{(t)} + b_{\mathrm{dec}} - x \right), \qquad z^{(t)} = \left[y^{(t)} - \eta g^{(t)} - \eta \lambda \right]_+ .$$

We terminate when

$$\frac{\|z^{(t)} - z^{(t-1)}\|_2}{\|z^{(t)}\|_2 + 10^{-12}} < 10^{-6},$$

or when the maximum iteration count is reached. Otherwise,

$$\tau_{t+1} = \frac{1 + \sqrt{1 + 4\tau_t^2}}{2}, \qquad y^{(t+1)} = z^{(t)} + \frac{\tau_t - 1}{\tau_{t+1}} \left(z^{(t)} - z^{(t-1)} \right).$$

Calibration uses $T_{\max} = 2000$, whereas FISTA-enc uses $T_{\max} = 20{,}000$.

D Additional results for Section 3

Subspace agreement is specific to the code. Because both codes are functions of the same activation, a possible interpretation might be that linear predictability between them might merely reflect that both encode x. We show it does not. Predicting the alternative code from the raw activation instead of the encoder code gives R^2 of only 0.34–0.48 at $p = 50$, the gap widening with p (to 0.66–0.89 at $p = 320$); by CKA the raw activation reaches 0.70–0.86 against the encoder code's 0.95–0.999. In-sample R^2 exceeds the cross-validated value by exactly p/n.

Near-duplicate atoms do not explain the disagreement. Trained decoders contain a small number of near-parallel atom pairs (34–102 pairs at $|\cos| > 0.9$ among 16,384 atoms, none forming clusters larger than two). Merging these before computing Jaccard changes it negligibly. The methods are not choosing between synonyms.

The encoder's code is a local optimum, not a poor approximation. IHT started from the encoder's own code stays there: Jaccard 0.99–0.996 across all configurations and datasets under a contractive step. IHT started from a random point lands elsewhere at Jaccard ≈ 0.45 with equal or better reconstruction. The ℓ_0-constrained problem on these dictionaries has many stationary points of comparable quality, and the encoder simply occupies one.

Non-uniqueness is not optimization noise. A competing explanation is that the sparse coding problem on these activations is ill-conditioned, so that every solver—the encoder included—returns a code that small perturbations would change. We perturb each activation with Gaussian noise, $x + \sigma\epsilon$, and measure the Jaccard between each method's clean and noisy supports. For this, the comparison methods are solved at the clean encoder's k, otherwise support size and support identity would change together. And the encoder is reported both raw and with its noisy code truncated to the clean k, because the encoder's sparsity turns out to be far less stable than its support: at $\sigma = 1$ it fires $7.6\times$ as many atoms as on the clean activation ($14\times$ at 65k), so raw Jaccard understates its support stability by ≈ 0.12 at every configuration. All methods are stable. Defining σ_{match} as the noise level at which the encoder's k-matched self-agreement falls to its clean agreement with OMP, we find $\sigma_{\text{match}} = 0.61$–$0.73$ across configurations and datasets. Swapping the inference algorithm perturbs the support as much as substantial input noise does. Each solver reliably returns its own code; the solvers reliably return different codes.

Robustness across activation sources and model families. The disagreement is not explained by near-duplicate atoms, by input noise, or by both codes encoding the same activation, and the encoder's code is a stationary point among many of comparable quality (above). The dissociation has the same shape on every activation source (raw Jaccard 0.44–0.58, top-10 0.74–0.83 across FACTUAL, MMLU, and CODE at layer 20) and on a second model family: a Llama-Scope TopK SAE at layer 24 of DeepSeek-R1-Distill-Llama-8B gives raw Jaccard 0.43–0.52, top-10 0.74–0.76, and shared-coefficient correlation 0.96–0.97, inside the Gemma range on every statistic. The phenomenon is not specific to one model, one SAE architecture, one trainer, or one input distribution.

Table 2: Loss recovered when each method's reconstruction replaces the residual stream at layer 20 (16k), 200 sequences, 25,400 positions. Clean CE 2.479; zero-ablated 15.312.

Method	CE	Loss recovered
Encoder	2.689	0.984
IHT	2.670	0.985
FISTA	2.649	0.987
OMP	2.625	0.989

Sparsity sweep: per-level values. Table 3 reports the values plotted in Figure 2. n tapers with k because OMP's cost scales as k^3; CIs are ± 0.008 at $n = 1000$ and ± 0.03 at $n = 250$. IHT-random's raw Jaccard is flat until the largest rung.

Table 3: Sparsity sweep at layer 20, 16k. Raw and top-10 Jaccard between the encoder and each method, PILE.

			Raw Jaccard			Top-10 Jaccard		
L_0	n	$\bar{k}$	OMP	IHT	FISTA	OMP	IHT	FISTA
22	1000	23	0.616	0.487	0.616	0.731	0.695	0.742
38	1000	40	0.574	0.496	0.604	0.776	0.757	0.788
71	2000	77	0.506	0.497	0.576	0.809	0.801	0.822
139	500	152	0.425	0.482	0.529	0.851	0.866	0.880
294	250	323	0.372	0.431	0.452	0.851	0.900	0.875

E LOCAL GEOMETRY OF THE ACTIVE SUPPORT

Random-matrix structure of the active support: derivation. To measure how far from orthogonal the active atoms are, we use the Gram matrix $G_S = D_S D_S^\top$ of the atoms an activation turns on: its entries are the pairwise cosines among active atoms, with ones on the diagonal. The eigenvalues summarize how non-orthogonal the set is. If the k active atoms were exactly orthogonal, every eigenvalue of G_S would equal 1, and the more that the atoms lean on each other, the more the eigenvalues spread away from 1. Since the eigenvalues always sum to k, the variance is the mean

squared cosine between pairs of active atoms, $\mathrm{Var}(\lambda) = (k-1)\,\overline{C_{ij}^2}$: eigenvalue variance and root-mean-square pairwise cosine are the same number. (This is the average overlap, distinct from the mutual coherence $\max_{i\neq j}|C_{ij}|$ that enters the ERC bound.)

Some of the spread is by chance, since in high-dimensional spaces, random unit vectors are nearly but not exactly orthogonal. The Marchenko–Pastur law tells us precisely how much spread k random directions in $\mathbb{R}^d$ produce: the eigenvalues fill an interval $[(1-\sqrt{\gamma})^2, (1+\sqrt{\gamma})^2]$ with $\gamma = k/d$, and their variance is $\gamma \approx (k-1)/d$, which is simply the statement that a random pair has cosine about $1/\sqrt{d}$. While we wouldn't expect the atoms a trained SAE fires together to look like random directions, a variance of c times the Marchenko–Pastur value says the co-firing atoms have $\sqrt{c}$ times the root-mean-square cosine of chance, and a top eigenvalue beyond the upper edge $(1+\sqrt{\gamma})^2$ says at least one direction in the active span is more crowded than k random vectors could produce, both in units with no fitted parameters.

Since chance is not the right baseline for a trained dictionary, we also compare against random k-subsets of the released dictionary, matched to each activation's k and estimated from 2000 draws per configuration. The baseline also has a closed form for the variance: the expected mean squared cosine over a random subset equals the mean squared cosine over all dictionary pairs, so the random-subset variance is $(k-1)$ times that dictionary-wide average. The excess over chance therefore has two parts: how non-orthogonal the dictionary's atoms are in general (MP $\rightarrow$ dictionary null), and how much more non-orthogonal the particular atoms that fire are together (dictionary null $\rightarrow$ observed). Figure 4 shows the pooled eigenvalue spectra against both nulls, and Table 4 splits the excess: the dictionary's own geometry carries most of it at layer 5, and co-firing selection carries most of it at layer 20 and at 65k width. Values are in Table 4.

Table 4: Eigenvalue variance of the active-support Gram matrix, PILE, against two nulls. "Dictionary" is random k-subsets of the released decoder; "selection" is the remaining excess attributable to which atoms co-fire. rms coherence is $\sqrt{\mathrm{Var}(\lambda)/(k-1)}$; the random-pair value is $1/\sqrt{d} = 0.021$.

Config	$\bar{k}$	$\mathrm{Var}(\lambda)$	$\times$MP	dictionary	selection	outside bulk	rms coh.
L5/16k	78	0.125	3.7	2.9	1.3	25%	0.040
L12/16k	86	0.140	3.8	2.1	1.8	24%	0.041
L20/16k	77	0.133	4.0	1.4	2.9	27%	0.042
L20/65k	64	0.209	7.5	1.6	4.7	38%	0.057

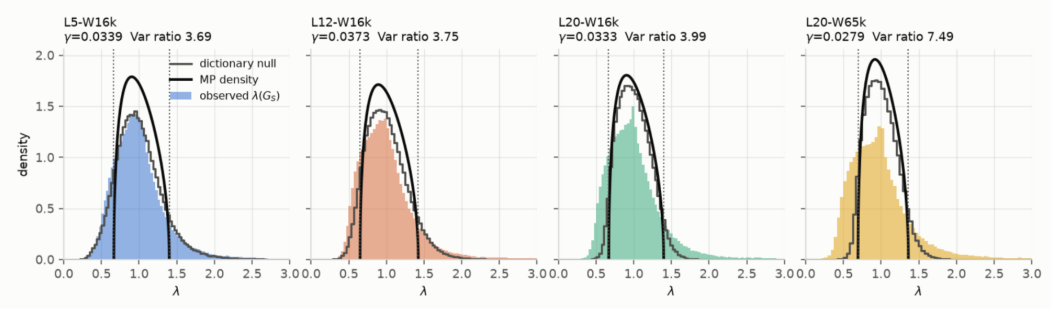

Figure 4: Eigenvalues of the active-support Gram matrix G_S, PILE, pooled over activations, against two nulls: the Marchenko–Pastur density for k random directions in $\mathbb{R}^d$ (thick; dotted lines mark its edges) and random k-subsets of the released decoder (thin). Panel headers give the observed variance as a multiple of the MP value, split into the dictionary's own geometry (MP $\rightarrow$ dictionary null) and co-firing selection (dictionary null $\rightarrow$ observed). 24–38% of eigenvalues fall outside the MP bulk.

The exact recovery condition across sparsity levels. ERC ≥ 1 on 100% of activations in every configuration and input distribution (mean 2.7). This is the case for GemmaScope's sparsest released version (1.61 at $k \approx 23$) and grows as the SAE is made less sparse ($1.61 \rightarrow 1.97 \rightarrow 2.65 \rightarrow 3.91 \rightarrow 6.72$ for $k = 23 \rightarrow 323$; Table 5). Random k-subsets of the same dictionary satisfy the condition below $k \approx 24$; the encoder's supports do not, even at the smallest k.

Table 5: Exact recovery condition of the encoder's support across the five released sparsity levels of the layer-20, 16k dictionary, PILE. ERC ≥ 1 on 100% of activations at every level; random k-subsets of the same dictionary satisfy ERC < 1 below $k \approx 24$.

L_0	$\bar{k}$	ERC
22	23	1.61
38	40	1.97
71	77	2.65
139	152	3.91
294	323	6.72

Support geometry across sparsity levels. Across the five released sparsity levels of the layer-20, 16k dictionary, the observed eigenvalue variance falls as k grows, while both nulls rise with $\gamma = k/d$. At the loosest level, the observed variance falls below the MP value. Atoms added at higher L_0 are less coherent with the support, not more.

The failure is not internal to the support. On the layer-20 dictionary the median $\lambda_{\min}(G_S)$ is 0.45 at 16k width and 0.39 at 65k, with no activation below 10^{-6}, so the active atoms are well-conditioned among themselves; the obstruction is between them and the inactive pool, which is what ERC measures and $\lambda_{\min}$ cannot see.

Per-activation regression: details. We regress agreement on k, effective rank $(\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$, ERC and $\lambda_{\min}$, all functions of $G_{S_{\text{enc}}}$. We fit ordinary least squares on standardized predictors, separately per configuration, dataset and method, and report 5-fold cross-validated R^2. These four candidate predictors cannot be used in conjunction since they are nearly the same variable. Effective rank is $0.885\,k$ with standard deviation 0.025, so k alone predicts it with $R^2 = 0.997$, and ERC correlates with k at 0.90. The fit is unchanged by trading one for another, and the individual coefficients are arbitrary. We therefore use k and the redundancy ratio, and ERC and cross-support coherence add less than 0.02 once these are present.

F ADDITIONAL CAUSAL RESULTS

The reconstruction-error floor. We measure the floor directly—the KL between the encoder-patched and clean runs—and it is 0.13 at layer 20 and 0.04 at layer 12, two to three orders of magnitude above the per-feature effects we measure, which is why the cancellation matters.

Why the random stratum carries the test. Every feature in the top-10 stratum is retained by all three methods ($2000/2000$ at layer 20, $1997/2000$ at layer 12). In that stratum agreement and magnitude are the same variable. The random stratum has all four agreement classes represented—but it does *not* equalize magnitude across them: agreed features are larger there too. Restricting to it is necessary for the test and not sufficient for the inference; magnitude must be a covariate.

Table 6: Median logKL by number of methods retaining the feature, random stratum, FACTUAL. The unadjusted contrast (bottom vs. top row) is $\approx 5\times$ at layer 20; the magnitude-adjusted increment is $\Delta R^2 = 0.013$.

	Layer 20		Layer 12	
n_{agree}	median logKL	n	median logKL	n
0	-7.77	266	-8.89	266
1	-7.28	129	-8.60	129
2	-7.01	172	-8.38	200
3	-6.09	433	-7.67	405

Full regression detail. The n_{agree} coefficient falls from 0.74 to 0.27 when $\log|z_{\text{enc},j}|$ enters. Within magnitude tertiles, the binary agreed-versus-none contrast ($n_{\text{agree}} = 3$ vs. 0, on fewer pairs than the

ordinal test of Section 5) is significant in the low and middle tertiles at layer 20 and in none at layer 12.

Table 7: Magnitude-adjusted agreement effects across layers and prompt sets. ΔR^2: gain from adding n_{agree} to $\log|z|$. Partial Spearman of n_{agree} with logKL controlling for $\log|z|$, pooled and in the lowest magnitude tertile.

Layer	Prompt set	ΔR^2	pooled ρ	tail ρ
20	FACTUAL	$+0.013$	$+0.27$	$+0.21$
20	MMLU	$+0.008$	$+0.22$	$+0.31$
12	FACTUAL	$+0.006$	$+0.14$	$+0.18$
12	MMLU	$+0.000$	n.s.	n.s.

Exclusive atoms. The exclusive atoms are a different story. For each atom the encoder selects and OMP does not, we find OMP's most similar exclusive atom and compare their ablation effects. The nearest substitute is nearly orthogonal (median cosine 0.08), the effects correlate at $+0.57$, and how similar the substitute is does not predict how similar its effect is ($\rho = 0.00$). Where the codes disagree, they are not choosing between synonyms; they are fitting the same residual with genuinely different directions whose downstream effects only partly coincide. This is the causal face of the orthogonal fringe of Section 4, and it gives the empirical relationship between support and causal identifiability a concrete form: agreement on an atom implies agreement on its effect, while disagreement on an atom means different atoms with partially overlapping effects.

The methods are not interchangeable. With separate membership indicators, membership in the IHT and FISTA supports predicts larger effects ($+0.21$ and $+0.17$ respectively), while membership in OMP's support predicts slightly *smaller* effects (-0.04 at layer 20, -0.09 at layer 12, confidence intervals excluding zero). The count n_{agree} averages over this. We report it because it is the natural operational proxy, and note that which method agrees matters.

A composite diagnostic. We combine n_{agree} with three local-geometry indicators of the feature's support—ERC below its per-configuration median, $\lambda_{\min}(G_S)$ above its median, and the feature's nearest inactive atom at similarity below 0.5—into a composite score from 0 to 6. Median logKL rises monotonically with the composite on both layers (-8.9 at 0 to -5.8 at 5 at layer 20); partial Spearman with effect, controlling for magnitude, is $+0.21$ at layer 20 and $+0.16$ at layer 12.

Alternative outcome. Using the log-probability of the gold answer letter as the outcome instead of KL yields no signal on either layer; magnitude alone explains $R^2 \leq 0.04$ of it, so this outcome is underpowered rather than contradictory.

G BEYOND SAES: SPARSE PROBES AND FORESTS

Setup. Three feature bases at layer 20 of Gemma-2-2B: MLP neurons ($d = 9{,}216$), the GemmaScope SAE code ($m = 16{,}384$), and the raw residual stream ($d = 2{,}304$). 37 concepts with labels computable from text (part of speech, code, quotation, numerals, sentence boundaries, and language identity on parallel Europarl text), each with at least 20,000 positives and document-level splits; six small-n sequence concepts on MMLU and FACTUAL are reported separately. ℓ_1-logistic and seven comparison learners fit each concept at budget $k \in \{1, \ldots, 256\}$ on identical standardized features: ℓ_1-logistic tuned to exactly k nonzeros then refit (the analogue of FISTA-top-k), greedy forward selection (the analogue of OMP), iterative hard thresholding from random and from ℓ_1 initializations (the latter a control that reproduces ℓ_1's support), the mean-difference ranking of Gurnee et al. (2023), a best-first decision tree with k splits, and 500-tree forests in two variants—every tree sees every feature, or a random $\sqrt{d}$ of them. Because a tree may re-split a feature, its realized support can be smaller than k; comparisons use the realized size. Resampling uses stratified half-samples of the balanced training set: 50 for the linear learners and tree on the MLP and residual bases, 25 on the SAE basis, and 2 for the forests (5 on one concept, 3 on three), a deadline-driven reduction that limits the precision of forest statistics. For each concept, k^* is the smallest budget within one point of a dense probe; results are stratified by k/k^*.

Scope. Trees and forests were fit on the MLP basis only; forests at $k = 256$ on one concept only. A layer-12 replication was planned but not run: the token-level layer-12 activations were removed for disk space before the probe phases and not recollected. All results in this section are therefore layer 20. The per-feature causal phase of the preregistration was not run.

Headline results. On MLP neurons at layer 20 (37 concepts, $B = 50$), the dissociation of Section 3 reproduces with an external label. At $k = 32$, ℓ_1 and greedy selection overlap at Jaccard 0.20 yet agree on 96% of held-out predictions ($\kappa = 0.96$), their logits correlate at 0.96, weights on shared features at 0.99, and each learner's logit is predicted from the other's support at cross-validated $R^2 = 0.95$ against 0.27 from a random support of the same size. The one-active-atom regime of App. A is reflected exactly: on all eight concepts with $k^* = 1$, every learner selects the same feature. Trees from one forest with no shared features agree at $\kappa = 0.93$–0.95: behavioral identifiability with the whole structural chain failing.

Headline dissociation, per learner. At $k = 32$ on MLP neurons, ℓ_1 against each comparison learner shows low support overlap with high functional agreement for six of the seven learners; the exception is IHT-ℓ_1, the control initialized at the ℓ_1 solution, which reproduces its support exactly (Jaccard 1.00). On the raw residual basis the same pattern holds for the linear learners. On the SAE basis it is present but weaker: greedy overlaps with ℓ_1 at Jaccard 0.52, and IHT-random agrees at $\kappa = 0.91$ with logit correlation 0.83 and cross-support $R^2 = 0.71$, against 0.89–0.96 on the other bases. Concepts are markedly sparser in the SAE basis (median k^* of 4 against 16 and 32 on MLP and residual), and non-uniqueness at k^* among the linear learners is similar across bases (mean Jaccard distance 0.33 MLP, 0.26 residual, 0.33 SAE).

Non-uniqueness and underdetermination past k^*. For ℓ_1 at the budget grid point nearest $2k^*$, support overlap with greedy across resamples is 0.30 (Jaccard, median over 37 concepts), while ℓ_1's agreement with itself across half-samples is 0.60. Restricting to the 30 concepts whose $2k^*$ lies on the budget grid gives 0.32 and 0.61. Pooling across all learner pairs gives cross-learner overlap 0.24–0.33 and within-learner stability 0.33–0.48. In every view, algorithm choice moves the support more than resampling, and resampling moves it substantially: the contested features are underdetermined by the data as well as non-unique across algorithms.

Forest usage frequencies. With only two forest resamples, forest statistics are indicative rather than precise. The frequency with which a forest's ten most-used features recur across resamples is 0.60 at $k = 32$ and 0.59 at $k = 128$ (medians over concepts), with the count of low-frequency fringe features growing from 44 to 51. The aggregate stability masks two regimes: on the 16 token-level concepts the top-10 recurrence rises from 0.86 to 0.96 as k grows, while on the 21 language-identity concepts it stays near 0.45. A stable core with a growing fringe describes the token concepts; the language concepts are contested throughout.

Preregistered predictions that fail. Two preregistered predictions fail. First, support geometry does not predict per-concept disagreement: sparsity and support redundancy reach cross-validated $R^2 = 0.05$ over 37 concepts, underpowered relative to the per-activation analysis of Section 4 (which has thousands of rows). Second, a dense probe on the union of forest supports loses to an ℓ_1 probe of the same size by 0.66 percentage points: contested supports are not complementary information.

H Related work

Antecedents. Yun et al. (2021) propose dictionary learning as a transformer interpretation tool, training a single overcomplete dictionary across BERT layers and using FISTA for inference. Their work predates the SAE literature and uses convex ℓ_1 relaxation as the inference algorithm rather than an amortized encoder. We compare the encoder's support against FISTA's on a fixed SAE dictionary, finding the convex relaxation does not recover the encoder's support—a result complementing rather than conflicting with their methodological choice.

Concurrent work on SAE identifiability. Cui et al. (2026) give theoretical conditions for identifiable SAEs at training time; we ask the orthogonal question of whether a fixed SAE's inference is

unique on a given activation. Joshi et al. (2026) prove linear identifiability is insufficient to determine the SAE decoder under their concept-shift framework—a parallel finding to our empirical result. Méloux et al. (2025) issue a programmatic call for uncertainty quantification across mechanistic interpretability methods; we instantiate that call concretely. Barin-Pacela et al. (2026) diagnose dictionary-learning failures in synthetic well-conditioned settings; their work and ours are orthogonal, addressing learnability of the dictionary and uniqueness of inference on a fixed dictionary respectively.