WHEN AI EXPLAINS AI: UNDERSTANDING AND MITIGATING THE INTERPRETABILITY LOTTERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpreting the internal representations of language models is crucial for understanding their behavior and guiding efforts to make them reliable and safe. Recent advances have enabled the discovery and interpretation of model features at scale. We study the *interpretability lottery*, in which a feature's attributed meaning remains contingent on the evidence and inferential choices of the interpretation procedure after evaluation and selection. A single published description leaves this semantic uncertainty implicit. We ask how the uncertainty relates to a feature's activating evidence and use, and how it can be mitigated. First, we probe an explain-then-score pipeline on Gemma Scope sparse autoencoder (SAE) features of Gemma-2-2B, rerunning it fifteen times on each of 150 features with the model, the SAE and the feature held fixed. No feature is near-unanimous across runs, the median feature receives about 9.5 effectively distinct labels, and automated judgments find substantial differences in meaning. Variation in held-out AUC does not track label instability, and label indices from different runs share less than a fifth of their top-ten retrieval results. Evidence selection produces the largest contrast in our factorial design, while context diversity correlates with instability ($\rho \approx 0.54$). Second, we use these findings to test mitigations. With examples drawn at random, greedy decoding does not reduce median instability, and aggregating runs reduces score SD more than medoid-label instability against matched controls. Selecting the explainer's top-k examples deterministically roughly halves instability at the same pipeline-run budget and holds on new features, yet even then most features remain above the near-unanimity threshold. These findings motivate treating a feature description as a semantic claim whose uncertainty and dependence on the selected evidence must be characterized alongside its evaluation score.

1 INTRODUCTION

Explaining how language models produce their behavior is important for identifying failure mechanisms and developing interventions that make their behavior more reliable and safe (Huben et al., 2024; Templeton et al., 2026). Recent advances in mechanistic interpretability have made it possible to isolate internal features and test their influence on model outputs (Bricken et al., 2023; Huben et al., 2024; Templeton et al., 2026). Sparse autoencoders (SAEs) have become a standard tool for decomposing the internal activations of large language models into a basis of sparsely active features (Bricken et al., 2023; Huben et al., 2024; Templeton et al., 2026). Open releases such as Gemma Scope (Lieberum et al., 2024) provide such decompositions for every layer of a model, with individual dictionaries containing tens of thousands of features.

Making these features usable at scale requires a semantic description of what each feature responds to. Automated interpretation (*auto-interp*) supplies a second system that assigns each feature a natural-language description (Bills et al., 2023; Paulo et al., 2025). In the standard design, a language model is shown a set of contexts on which the feature is active and asked to describe what they have in common; a second model then scores the description by how well it predicts the feature's activation on held-out text. The resulting labels are published and searched through platforms such as Neuronpedia (Lin, 2023), where they serve as the default description of each feature.

Assigning meaning to a feature requires generalizing from selected activation examples to a claim about what the model represents across inputs. We define the *interpretability lottery* as the depen-

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.

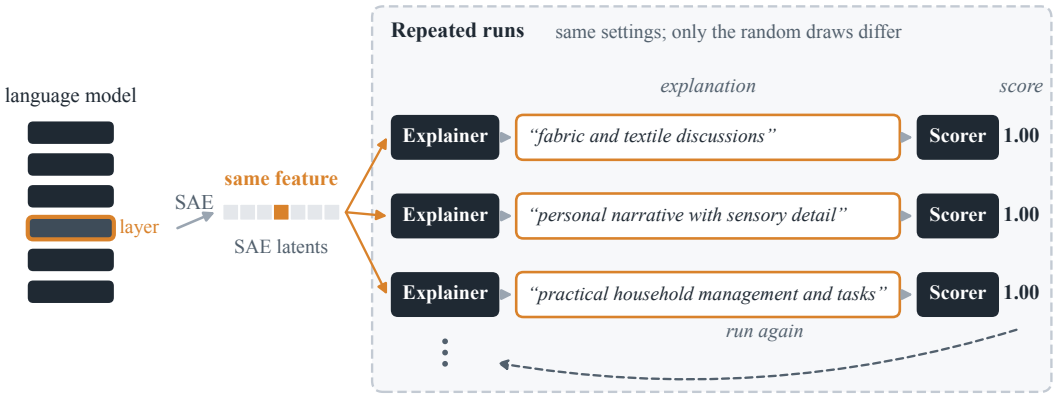

Figure 1: An example of the interpretability lottery for a fixed SAE feature. An SAE decomposes one layer of a language model into latents, one of which is the feature. The same feature is then run through the same explainer and scorer several times, with identical settings and only the random draws differing. Each illustrated run returns a different explanation, all with the same score, a held-out AUC of 1.00. The explanations are real runs of feature 16191 in our study (Neuronpedia: `gemma-2-2b/12-gemmascope-res-16k/16191`).

dence of that semantic claim on the evidence and inferential choices of the interpretation procedure, even after explanations have been evaluated and selected. A single published label can conceal substantively different accounts of the same feature that the workflow could also evaluate and select. These accounts may describe different valid facets or reflect limitations of the procedure. Our scientific question concerns the uncertainty hidden by a single selected description of the feature.

We probe this dependence by varying the interpretation process while holding the feature fixed. The explainer samples its description, the scorer samples its probe texts, and the contexts shown to the explainer are themselves drawn from a larger pool. A published label is therefore a single realization of a random process, yet it can be read as a property of the feature. Recent work has begun to ask what such choices do to auto-interp *scores* (van der Ben et al., 2026); the question we take up concerns the descriptions themselves, which is what a reader of a feature index actually consumes. A score evaluates a description on the tests used by the pipeline; it does not by itself establish that other evidence or explanatory choices would support the same semantic claim. Characterizing that dependence is part of assessing what an interpretation can support (Cronbach & Meehl, 1955; Messick, 1995), a framing others have brought to machine learning (Jacobs & Wallach, 2021; Méloux et al., 2025). Figure 1 illustrates the gap. One language model explains the internals of another, selecting different descriptions of the same feature with identical reported AUCs.

These labels decide what researchers inspect, steer and cite as evidence about what a model represents, including in safety work, where features are used to monitor and change model behavior (Templeton et al., 2026). The semantic claim selected by the pipeline can thus influence which internal component becomes the subject of further analysis. We treat the pipeline as a measuring instrument to connect the uncertainty in those claims to the evidence being interpreted, their use in feature retrieval, and practical controls on the interpretation process.

- *RQ1: What semantic uncertainty remains after explanation selection, and how does it relate to reported scores?* We ask whether selected labels express different meanings and whether variation in their evaluation scores tracks that disagreement.
- *RQ2: How do activating evidence and pipeline choices shape the selected interpretation?* We relate context diversity to label instability and compare variation in the examples, explainer and scorer.
- *RQ3: Which interventions mitigate the lottery, and at what cost?* This includes greedy decoding, evidence selection and aggregation.

The paper answers these questions in two parts (Figure 2). The first probes the pipeline. Rerunning it fifteen times on each of 150 features, we find that no feature is near-unanimous, that score variation

does not track label instability, and that example selection produces the largest contrast in the tested factorial design (Sections 4.2 and 4.3). The second turns this diagnosis into a mitigation. Fixing the examples shown to the explainer, as published pipelines already do, here to its top-k contexts, roughly halves instability at the same pipeline-run budget and holds on new features, although nearly every label still varies under it. With examples drawn at random, greedy decoding does not improve median label instability, and aggregating runs stabilizes the score but has a smaller relative effect on medoid-label instability (Section 4.4). Finally, analyses of the baseline runs connect semantic disagreement to how the labels are used. A search over the resulting descriptions returns different features depending on which interpretations populate the index (Section 4.5).

2 Related Work

Automated interpretability. Sparse autoencoders decompose a model's activations into sparsely active features (Bricken et al., 2023; Huben et al., 2024), addressing the polysemanticity that superposition produces (Elhage et al., 2022). They have been scaled to frontier models (Templeton et al., 2026; Gao et al., 2024) and released for every layer of open models (Lieberum et al., 2024; Rajamanoharan et al., 2024), making automated descriptions important for interpreting large feature sets. Bills et al. (2023) introduced the design that later work follows: show a model contexts on which a unit is active, ask for a description, and score the description by how well it predicts held-out activations. Delphi (Paulo et al., 2025) extends it to millions of SAE features, and Neuronpedia (Lin, 2023) publishes the resulting labels. Both fix the examples shown to the explainer: the OpenAI method behind Neuronpedia selects the top-activating records, and Delphi samples activation quantiles under a fixed seed. These policies motivate our comparison of how evidence selection affects the semantic content of labels (Section 4.4). Agentic explainers such as SAGE (Han et al., 2026) replace the single pass with a loop that proposes several hypotheses and tests them against activations, a natural candidate for reducing the variation we measure.

Evaluation and uncertainty in automated interpretability. Most evaluation asks whether an explanation is correct. High simulation scores coexist with causal failure (Huang et al., 2023), several metrics do not change when labels are corrupted (Oikarinen et al., 2025), and scores do not separate trained transformers from random ones (Heap et al., 2025). Other work characterizes the descriptions themselves (Puri et al., 2025; Gur-Arieh et al., 2025; McCann, 2026), and instability has been documented one level down, in SAEs trained under different seeds (Paulo & Belrose, 2025; Gerasimov et al., 2026) and in SAE benchmarks (Karvonen et al., 2025; Chanin, 2026), with feature consistency linked to explanation similarity (Song et al., 2026). At the pipeline level, our closest comparison, van der Ben et al. (2026), finds that workflow choices strongly affect auto-interp scores and feature rankings, with the example draw the largest source of score variation. We hold the trained feature and explainer fixed, vary the scorer separately, and examine the selected label's semantic content, its consequences for retrieval, and the distinct effects of mitigation on descriptions and scores. Both studies find evidence selection consequential on different models and outcomes.

3 Probe and Mitigation Framework

Interpreting a feature requires generalizing from activation evidence to a semantic claim that can guide subsequent analysis. Our framework asks how dependent that claim remains on the evidence and inferential choices of the procedure after evaluation and selection. We therefore hold the feature's activation function fixed and probe how the selected descriptions vary as those choices change. Semantic dispersion characterizes this dependence within a specified workflow, without assuming a single correct description. Mitigation then targets the choices implicated by the probe, with predictive fit and inference cost assessed alongside semantic stability (Figure 2).

3.1 The auto-interp pipeline under test

The workflow must both propose semantic hypotheses and decide which one will represent the feature. Auditing the selected output addresses the claim a downstream user receives, including the filtering imposed by evaluation. We study the explain-then-score design of Bills et al. (2023) as it is applied to SAE features, in four stages (Figure 2). *Sample*: a few of the feature's stored activating contexts are drawn. *Explain*: an explainer model reads them and proposes several candidate labels.

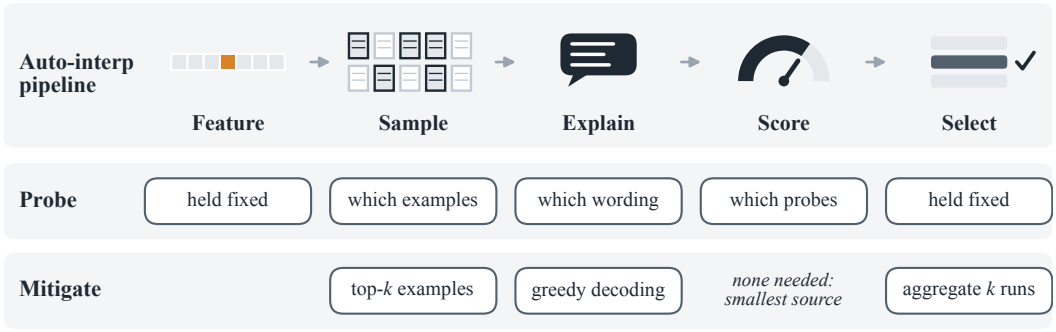

Figure 2: The probe and the mitigations, placed on the stages of one run of the auto-interp pipeline (top row). The probe reruns the pipeline on the same feature; the feature and the selection rule are held fixed, so every difference between runs comes from the three stages that draw randomly (Probe row). Each mitigation acts on the stage above it (Mitigate row); the scorer has none because it is the smallest source of label instability (Section 4.3).

Score: for each candidate, a scorer model writes probe texts, half meant to activate the feature and half not, and the candidate is scored on two channels. These test association with activation and sensitivity of model outputs to interventions, respectively. The first is the area under the ROC curve (AUC) of the feature's activation across the probes, where 0.5 is chance. The second is causal, run on a few of the candidate's positive probes: *steering* adds the feature's decoder direction to the residual stream while the model generates a short continuation, and a judge model rates whether the continuation matches the label; *ablation* projects the feature out and measures the change in logits, without comparing the output with the label. The two channels are combined into a confidence score, whose weights and pass threshold are fixed before collection to apply the same selection rule in each comparison,

$$\text{conf} = 0.60\,\text{AUC} + 0.40\,\text{causal}, \qquad \text{causal} = 0.60\,\text{steer} + 0.40\,\text{abl}. \tag{1}$$

Select: the candidate with the highest score is published. A fresh set of probes is then written for it, and every AUC we report is computed on these held-out probes, so its predictive fit is evaluated on tests not used to select the candidate. Three of the four stages draw randomly on every run: which contexts are shown, how the explainer words its candidates, and which probes the scorer writes.

3.2 Probe design for semantic uncertainty

To study dependence on the interpretation procedure, we rerun it while holding the language model, SAE, feature, context pool and workflow configuration fixed. Differences then arise from the procedure's own random draws. Random example draws expose dependence on partial views of the same context pool. Published pipelines fix the examples (Section 2), as our top-k intervention does, providing a policy that controls this source of variation.

A stage's apparent contribution can depend on which other stages vary, so attribution requires both conditional and joint comparisons. The first design frees one random stage at a time and pins the other two by replaying the same draw on every run. Because each source is then measured with the other two held at a single setting, the three measurements need not add up to the variation seen when all three are free. The second design therefore crosses the sources. The $2 \times 2 \times 2$ factorial compares all eight fixed/random workflow conditions, estimating average contrasts and interactions across the settings of the other stages. Here, pinning uses cached responses. Because the prompt enters the cache key, changing examples can also trigger fresh language-model calls. These contrasts measure workflow effects without fully isolating the underlying sampling mechanisms.

3.3 Measures of semantic variation and score reliability

Agreement in evaluation scores and agreement about a feature's meaning are distinct properties of the pipeline. We measure both. Our primary label-instability measure uses similarity between

descriptions to count effectively distinct labels. We embed the N labels a feature received as unit vectors, form their similarity matrix K, and take the Vendi score V (Friedman & Dieng, 2023), the exponentiated entropy of the eigenvalues of K/N, which reads as an effective number of distinct labels between 1 and N. Normalization expresses this diversity on a common scale,

$$\text{instability} = \frac{V - 1}{N - 1} \in [0, 1], \tag{2}$$

where 0 means every run produced the same label and 1 means the labels are as different as the embedding allows. Because paraphrases keep the index slightly above zero, we call a feature *near-unanimous* when its instability is below 0.05. The continuous index remains primary; Appendix A reports sensitivity to this cutoff. To check whether the conclusions depend on embedding geometry, we group a feature's labels into clusters in which every pair entails each other in both directions, using an entailment model, as semantic entropy does (Kuhn et al., 2023). The count provides a complementary, automated measure of dispersion among the selected descriptions.

Score reliability asks whether scores distinguish features consistently across runs. We report reliability for held-out AUC and for confidence as the intraclass correlation ICC(3,1) (Shrout & Fleiss, 1979), computed on the table of n features by k runs. A value near 1 indicates that between-feature differences dominate residual variation across runs of the same feature. Appendix A gives the formula, rating bands and absolute-agreement variant. A feature is floor-censored if its held-out AUC is at or below 0.5 in every run, leaving its score almost no room to vary; we report reliability with and without these features (Appendix D). A feature passes in a run when its confidence reaches $\tau = 0.65$. The *flip fraction* is the share of features whose 95% bootstrap interval for mean confidence contains τ, that is, features whose verdict could go either way depending on the sampled pipeline run.

3.4 Mitigation through pipeline interventions

The probe motivates controls on the evidence supplied to the explainer and on hypothesis generation. Aggregation tests whether pooling repeated inference can stabilize both descriptions and scores. *Top-k examples* always shows the explainer the feature's most strongly activating contexts, testing whether a consistent view of its strongest responses reduces dispersion. This changes both evidence content and determinism. *More examples* shows the explainer more contexts per run, testing whether broader coverage of the pool reduces sensitivity to the draw. *Greedy decoding* sets the explainer's temperature to zero, targeting explainer sampling while leaving the evidence-selection policy unchanged. *Aggregating k runs* splits a feature's runs into disjoint blocks of k; each block publishes its medoid label, the label most similar on average to the others in the block, and its mean held-out AUC. We compare the stability of these block outputs with a control that takes one random run from each block, so that both are measured over the same number of items.

We do not target the scorer, because it is the smallest of the three sources of label instability (Section 4.3). The three single-run interventions cost one run per feature, as the unmodified pipeline does; aggregation costs k runs. Mitigation selection constrains predictive fit because lower dispersion alone need not yield more informative labels. We compare the single-run interventions on one set of features, disqualify any that lowers median held-out AUC by more than 0.05, and promote the one with the largest reduction in median label instability. The promoted intervention is then rerun on a disjoint set of features, testing whether the benefit extends beyond those used to choose it. When an intervention supplies more contexts, token use and monetary cost can increase even if the pipeline-run count stays fixed.

4 Experiments

4.1 Experimental setup

We study SAE features of Gemma-2-2B (Gemma Team, 2024), using the canonical Gemma Scope residual-stream SAE (Lieberum et al., 2024) at layer 12 with 2^{14} latents (`average_l0_82`), the SAE behind Neuronpedia's published labels for this model. Each feature stores up to forty activating contexts, of which each run shows the explainer five. The explainer is Claude Haiku at temperature 1.0, proposing five candidate labels; the scorer is GPT-4o-mini at temperature 0.2, writing twenty

Table 1: Selected labels exhibit substantial semantic dispersion, while score reliability is moderate. 150 features × 15 runs. Metrics are defined in Section 3.3. An ICC is the share of score variation attributable to differences between features; *moderate* (0.5 to 0.75; Koo & Li, 2016) means a quarter to a half of it is run-to-run noise.

	Metric	Value
Label	Instability, mean	0.581
	Instability, median	0.604
	Near-unanimous features	0 of 150
Score	ICC(3,1), held-out AUC	0.599 *(moderate)*
	without the 9 floor-censored features	0.572 *(moderate)*
	ICC(2,1), held-out AUC	0.598
	ICC(3,1), confidence	0.719 *(moderate)*
	ICC(2,1), confidence	0.719
Verdict	Flip fraction	0.127

probes per candidate. The causal channel uses two probes per candidate, steering adds the decoder direction at unit scale, and the steering judge is Claude Haiku at temperature 0. Top-k shows the five strongest contexts and more examples shows twenty; Appendix C gives exact settings.

The main experiment runs the pipeline fifteen times on each of 150 features, stratified by activation frequency with equal numbers from each decile; nine of them are flagged as floor-censored. A pilot of 50 features × 10 runs, on a disjoint list, set the sample size and was used to compare the single-run mitigations, and the promoted mitigation was rerun on the 150 main features. Hypotheses, thresholds and exclusion rules were fixed before the main collection. The one-at-a-time attribution uses 50 features × 10 runs per condition and the factorial 50 features × 5 runs per cell; in total the study comprises 10,000 pipeline runs. Label embeddings use the Sentence-BERT encoder `all-MiniLM-L6-v2` (Reimers & Gurevych, 2019), and every conclusion that depends on them is recomputed under `all-mpnet-base-v2` and reported only if it holds under both; entailment clusters use DeBERTa-large-MNLI (He et al., 2020). Appendix B.9 gives uncertainty intervals for the main quantitative findings.

4.2 RQ1: What semantic uncertainty remains after explanation selection?

Finding 1

Rerunning the pipeline on the same feature leaves substantial semantic variation in the selected label. None of the 150 features is near-unanimous across fifteen runs. Score reliability is moderate, and features whose labels change more do not have scores that change more.

No feature is near-unanimous across fifteen runs. With the model, the SAE and the feature held fixed, not one of the 150 features is near-unanimous, and the median feature receives about 9.5 effectively distinct labels in fifteen runs (Table 1). Score reliability, by contrast, is moderate for both held-out AUC and confidence, and it barely changes when floor-censored features are excluded or shifts shared within a run are counted. The instability is supported by complementary measures. The entailment measure ranks features similarly and finds more separate clusters, not fewer (Figure 3a), and a second encoder gives the same result (Appendix B.2). These differences concern the claims selected for a feature. In the baseline analyses of Section 4.5, two judges classify only 2.8% to 5.1% of sampled non-identical label pairs as equivalent, and indices built from different runs share less than a fifth of their top-ten retrieval results. Thus the lottery affects both descriptions and the features retrieved by a fixed semantic query.

Score variation does not track label instability. Across the 141 features outside the floor-censored subset, a feature's label instability shows no detectable correlation with the run-to-run variation of its held-out AUC (Figure 3b); the interval in Appendix B.9 rules out a correlation stronger than about 0.18 in either direction. Nor is this a ceiling effect, since only 2 of the 150 features have a mean held-out AUC of 0.99 or more. The pipeline thus reports scores whose variation does not indicate the semantic dispersion of the selected labels. Average score remains a separate property of these

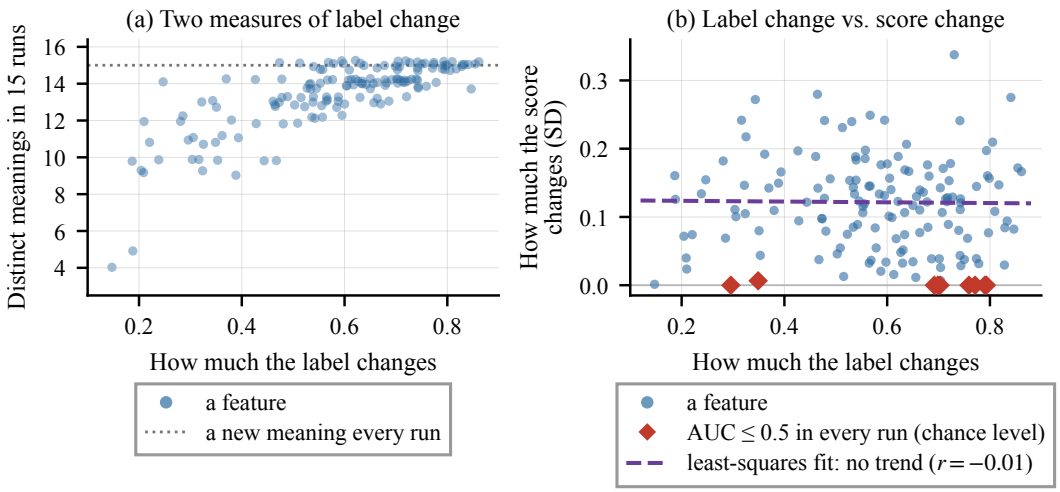

Figure 3: Labels are unstable under both measures, and how unstable a feature's label is shows no detectable linear association with how much its score varies. One point per feature (150 features $\times$ 15 runs). (a) The embedding measure and the entailment measure rank features alike, but entailment separates descriptions more finely. The 44 features on the dotted line have one entailment cluster per run. Counts are jittered vertically. (b) The dashed line is a least-squares fit over the 141 features outside the floor-censored subset; red diamonds mark the nine excluded features.

interpretations. Auditing the score components separately confirms a similarly weak association for both label-dependent AUCs (Appendix B.3).

4.3 RQ2: How do evidence and pipeline choices relate to label instability?

Finding 2

The largest factorial contrast concerns the examples shown to the explainer. Once the examples vary, letting the explainer or the scorer vary as well adds almost nothing in these conditions, so a source measured on its own can look much larger than what it adds to the full pipeline.

The largest contrast is associated with the example draw. Features with more diverse activating contexts have more variable selected descriptions ($\rho \approx 0.54$ under both encoders; Section 4.5). This connects workflow sensitivity to the evidence available for a feature and motivates comparing evidence-selection policies. To examine the workflow choices associated with this variation, we crossed the three random steps in a $2 \times 2 \times 2$ factorial, so that every combination of random and fixed steps was run (Section 3.2). Table 2 gives label instability in each of the eight combinations. With all stages pinned through replay, the labels repeat. Varying examples while pinning the other stages under the cache protocol produces nearly all the instability of the full pipeline, the explainer's sampling alone produces less, and the scorer's little. A separate experiment that frees one step at a time gives the same ranking (Appendix B.4). The feature in Figure 1 illustrates the role of the draw. Its stored contexts come from unrelated settings, from sewing projects and personal stories to chip manufacturing and a software component named Fabric, so different draws of five show the explainer different settings, and its labels differ accordingly. With the examples fixed, the explainer's sampling alone still leaves the labels far from stable.

Workflow contrasts depend on which other stages vary. The right column of Table 2 is nearly flat. This pattern is consistent with overlapping effects of evidence selection and generation, but the bounded index and cache design prevent us from identifying a unique mechanism. Conditional effects are not additive. The one-at-a-time measurements add up to more than the instability seen when all three sources vary. On held-out AUC variance, each source measured alone is two to three times its average effect in the factorial (Appendix B.4). For the score, however, the example draw and the explainer matter about equally, so the relative importance of workflow choices depends on whether the outcome is semantic dispersion or score variance.

Table 2: Once the examples vary, the labels are about as unstable as they get, across the tested cache conditions. Label instability in each cell of the $2 \times 2 \times 2$ factorial, 50 features $\times$ 5 runs per cell. With five runs per cell, values are not comparable with the fifteen-run values of Table 1.

Also random	Examples fixed	Examples random
Neither explainer nor scorer	0.01	0.78
Scorer	0.18	0.76
Explainer	0.53	0.77
Explainer and scorer	0.51	0.79

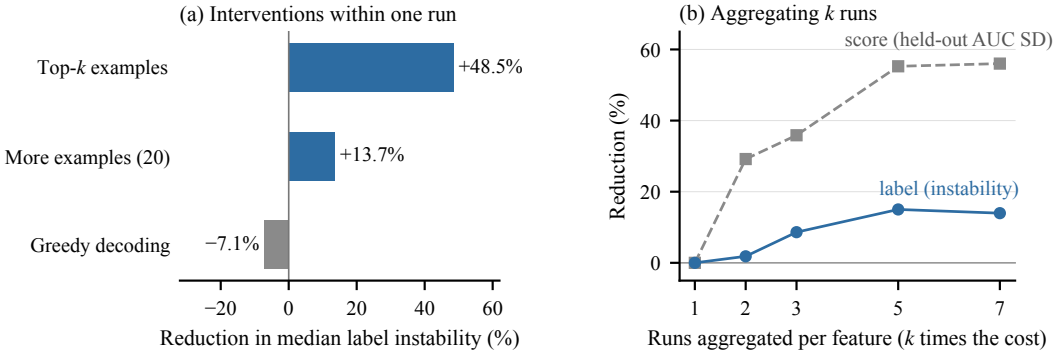

Figure 4: Top-k example selection reduces label instability far more than the other interventions, and aggregating runs helps the score far more than the label. (a) Single-run interventions, each costing one run per feature, as the reduction in median label instability on the pilot features. (b) Aggregating k runs, costing k runs per feature, as the reduction in mean variation against a matched control on the main features. The panels use different samples and statistics and are not on a common scale (Appendix B.5).

4.4 RQ3: Which interventions mitigate the lottery, and at what cost?

Finding 3

With examples drawn at random, greedy decoding does not reduce median label instability. Selecting the explainer's top-k activating examples roughly halves label instability at the same pipeline-run budget, and the gain holds on features it was not chosen on.

Greedy decoding does not reduce median label instability when examples are drawn at random. Setting the explainer's temperature to zero changes median label instability from 0.645 to 0.690, while mean instability changes from 0.652 to 0.632. The primary median criterion therefore shows no improvement (Figure 4a; Table 9 in Appendix B.5). The explainer still proposes about as many distinct candidates, leaving the selection stage multiple alternatives. This test drew examples at random, where the draw dominates; with fixed examples the explainer is the main remaining source (Table 2). The effect of greedy decoding with fixed examples remains untested in this study.

Top-k example selection halves instability, and the gain holds on new features. On the pilot, top-k examples removes about half of median label instability and more examples removes much less, both using one pipeline run per feature and improving the measured AUC proxy on the pilot (Figure 4a). Rerun on the 150 main features, none of which it was chosen on, top-k reproduces its pilot result almost exactly, with improvements across the distribution of per-feature instability (Appendix B.5). Median instability falls from 0.604 to 0.328 (45.7%), with a change of -0.012 in median feature-mean held-out AUC. This agrees with both attribution designs ranking the example draw first. More examples also help, at a cost that grows with their number; we did not test more than twenty. Even with top-k examples, 144 of the 150 features remain above the near-unanimity threshold. The result establishes partial control over semantic dispersion, while correctness and coverage still require independent assessment.

Aggregation has different effects on score and label variation. Averaging five runs reduces held-out AUC SD by 55.3%, while medoid-label instability falls by 15.0% against its matched control, with no further gain at $k = 7$ in this experiment (Figure 4b). Score noise averages out, but a medoid can only return a label that some run produced, and choosing one representative need not resolve differences in semantic content. Agreement in numerical outcomes and agreement in semantic conclusions are therefore distinct objectives for pipeline design.

4.5 Semantic consequences and evidence dependence

Finding 4

Runs usually differ in what they say a feature means, not only in how they phrase it, and this changes which features a search over labels returns.

Most cross-run disagreements express different meanings. We examine the semantic content and consequences of the selected descriptions using the baseline confirmatory runs. Two judges from different providers classified $1{,}500$ non-identical cross-run label pairs (Appendix B.6). Under either judge, rewording accounts for one pair in twenty or fewer, while roughly a quarter of pairs name unrelated concepts and most of the rest overlap without being equivalent; very few pairs in the full set of cross-run comparisons are identical strings. We claim nothing from the intermediate categories, on which the judges disagree.

Instability carries over to search. An index built from one run's labels and queried with reference descriptions shares under a fifth of its top-ten results with an index built from another run (overlap@10 $= 0.186$, versus a chance reference of 0.067). The queried feature enters the top ten in some runs but not others for 89.3% of queries across fifteen runs. The index covers the 150 studied features, so these figures describe it and do not extrapolate to a full 2^{14}-feature index (Appendix B.7). Descriptive collision, where many features share one label (McCann, 2026), presents a complementary challenge. Descriptions can vary within a feature and overlap across features. A change in the selected description can therefore change which internal component a fixed query brings to the researcher's attention.

Features whose contexts are more varied get less stable labels. A feature's stored contexts can be alike or very different, as for the feature in Figure 1. We measure *context diversity* by applying the instability index of equation 2 to embeddings of a feature's stored contexts. The more varied a feature's contexts, the less stable its label (Spearman $\rho \approx 0.54$ under both encoders; Appendix B.8). A possible explanation follows from Section 4.3. Each run shows the explainer five of the contexts, and when the contexts differ, different draws support different labels. A diversity measure based only on word overlap shows the same association, and the number of stored contexts shows little marginal association with instability. Of five feature properties we tested, all computable from the SAE without a language-model call, only context diversity is supported under both encoders. It is therefore a candidate signal for prioritizing features for further interpretation; its predictive value over the full SAE remains untested. Even the least varied tercile has mean instability 0.459, so low diversity does not certify a stable label.

5 Conclusion

The interpretability lottery concerns how an interpretation procedure turns activation evidence into a semantic claim about a model feature. Our study shows that evaluation and selection leave substantial uncertainty in that claim. For the tested explain-then-score pipeline, no feature is near-unanimous across fifteen runs, score variation does not track label instability, and the largest contrast in our factorial design concerns the examples shown to the explainer. The association with context diversity and changes in feature retrieval link this uncertainty to activating evidence and downstream use. Acting on that diagnosis, selecting top-k examples roughly halves the instability at the same pipeline-run budget, whereas greedy decoding does not improve the primary median measure when the examples are drawn at random, and aggregating runs mainly stabilizes the score. Even with the fix, nearly every feature remains above the near-unanimity threshold. We recommend publishing labels with their semantic variation across runs and the example-selection rule, while separately assessing its effect on label coverage.

REPRODUCIBILITY STATEMENT

The supplementary material contains the source code: the pipeline, the experiment configurations, the seeded script that draws the stratified feature samples, and the analysis and figure scripts that compute the reported statistics from the run outputs. Section 3.1 gives the pipeline and Section 4.1 the experimental setup; Appendix C records the exact model snapshots, decoding parameters, prompts, seed policy and per-condition run counts required to repeat the study. Because the pipeline queries commercial APIs whose behavior may change over time, a rerun may not reproduce our outputs exactly.

AI USE STATEMENT

Generative AI is the object of study in this work: the pipeline under test uses commercial language models as its explainer and scorer, and two further models are used as judges in the disagreement analysis (Section 4.5). All such uses are documented in Section 4.1 and Appendix C.

Generative AI was additionally used as an assistant in preparing this work, for four purposes. *Research execution and ideation:* it implemented and tested analysis code, and proposed several of the follow-up analyses that the authors then chose to run (the retrieval, judge-disagreement and feature-level predictor analyses, and the factorial attribution design). *Drafting:* it produced first drafts of the Introduction, Related Work, Experimental Setup and Results sections and of the figures, which the authors revised. *Writing aid:* it edited prose for clarity and tone. *Retrieval:* it helped locate related work. All AI-assisted code was reviewed by the authors and is covered by an automated test suite, and all statistics reported in this paper were produced by that code from the recorded run outputs. The hypotheses, the decision to run each analysis and the interpretation of results are the authors'. Every reference was checked against a primary source. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models, 2023. OpenAI. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`.

Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. SAELens. `https://github.com/decoderesearch/SAELens`, 2024.

Trenton Bricken, Adly Templeton, Joshua Batson, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

David Chanin. Are sparse autoencoder benchmarks reliable? *arXiv preprint arXiv:2605.18229*, 2026.

Lee J. Cronbach and Paul E. Meehl. Construct validity in psychological tests. *Psychological Bulletin*, 52(4):281–302, 1955.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023. arXiv:2210.02410.

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.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Gleb Gerasimov, Timofei Rusalev, Nikita Balagansky, Daniil Laptev, Vadim Kurochkin, and Daniil Gavrilov. Unstable features, reproducible subspaces: Understanding seed dependence in sparse autoencoders. *arXiv preprint arXiv:2606.12138*, 2026.

Yoav Gur-Arieh, Roy Mayan, Chen Agassy, Atticus Geiger, and Mor Geva. Enhancing automated interpretability with output-centric feature descriptions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. arXiv:2501.08319.

Jiaojiao Han, Wujiang Xu, Mingyu Jin, and Mengnan Du. SAGE: An agentic explainer framework for interpreting SAE features in language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 483–495, Rabat, Morocco, 2026. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Automated interpretability metrics do not distinguish trained and random transformers. *arXiv preprint arXiv:2501.17727*, 2025. Version 2, January 2026.

Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously assessing natural language explanations of neurons. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 317–331, 2023.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Abigail Z. Jacobs and Hanna Wallach. Measurement and fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pp. 375–385, 2021. doi: 10.1145/3442188.3445901.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum 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 *Proceedings of the 42nd International Conference on Machine Learning*, 2025. arXiv:2503.09532.

Terry K. Koo and Mae Y. Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2):155–163, 2016.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2302.09664.

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.

Johnny Lin. Neuronpedia: Interactive reference and tooling for analyzing neural networks, 2023. Software available from `https://www.neuronpedia.org`.

Jordan F. McCann. Descriptive collision in sparse autoencoder auto-interpretability: When one explanation describes many features. *arXiv preprint arXiv:2605.12874*, 2026.

Maxime Méloux, François Portet, and Maxime Peyrard. Mechanistic interpretability as statistical estimation: A variance analysis. *arXiv preprint arXiv:2510.00845*, 2025.

Samuel Messick. Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning. *American Psychologist*, 50 (9):741–749, 1995.

Tuomas Oikarinen, Ge Yan, and Tsui-Wei Weng. Evaluating neuron explanations: A unified framework with sanity checks. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. arXiv:2506.05774.

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

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48393–48421, 2025.

Bruno Puri, Aakriti Jain, Elena Golimblevskaia, Patrick Kahardipraja, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. FADE: Why bad descriptions happen to good features. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17138–17160, Vienna, Austria, 2025.

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*, 2024.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Patrick E. Shrout and Joseph L. Fleiss. Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428, 1979.

Xiangchen Song, Aashiq Muhamed, Yujia Zheng, Lingjing Kong, Zeyu Tang, Mona T. Diab, Virginia Smith, and Kun Zhang. Mechanistic interpretability should prioritize feature consistency in sparse autoencoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2172–2210, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.99.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *arXiv preprint arXiv:2605.29358*, 2026.

Sinie van der Ben, Neele Roch, Anna Hedström, and Mennatallah El-Assady. Building fast, evaluating slow: Pipeline choices dominate autointerpretability score variance. *arXiv preprint arXiv:2607.19386*, 2026.

A Measure definitions

These are the standard definitions behind the measures used in Section 3.3, collected here so that the main text can state what each quantity is for.

Probe AUC. For positive probes P and negative probes N, with a_x the feature's peak activation on probe x, ties counted as one half,

$$\mathrm{AUC} = \frac{1}{|P|\,|N|} \sum_{p \in P} \sum_{n \in N} \Big[\mathbf{1}(a_p > a_n) + \tfrac{1}{2}\mathbf{1}(a_p = a_n) \Big]. \tag{3}$$

Vendi score. With K the similarity matrix of the N unit label embeddings and λ_i the eigenvalues of K/N,

$$V = \exp\Big(-\sum_{i=1}^{N} \lambda_i \log \lambda_i \Big), \qquad \text{instability} = \frac{V-1}{N-1}. \tag{4}$$

A feature is near-unanimous when its instability is below 0.05, or, under the entailment measure, when all its runs fall in one cluster. No result on the main sample depends on this cutoff, since its least variable feature has instability 0.147. The near-unanimous count under top-k examples does depend on it, with 3, 6 and 13 of 150 features at cutoffs of 0.02, 0.05 and 0.10.

Intraclass correlation. Let y_{ij} be the score of feature i in run j, for n features and k runs, with feature means $\bar{y}_{i\cdot}$, run means $\bar{y}_{\cdot j}$ and grand mean $\bar{y}$. With $\mathrm{MS_R}$ the between-feature mean square, $\mathrm{MS_C}$ the between-run mean square and $\mathrm{MS_E}$ the residual,

$$\mathrm{MS_R} = \frac{k}{n-1}\sum_{i=1}^{n}(\bar{y}_{i\cdot}-\bar{y})^2, \qquad \mathrm{MS_E} = \frac{1}{(n-1)(k-1)}\sum_{i=1}^{n}\sum_{j=1}^{k}(y_{ij}-\bar{y}_{i\cdot}-\bar{y}_{\cdot j}+\bar{y})^2, \tag{5}$$

$$\mathrm{ICC}(3,1) = \frac{\mathrm{MS_R}-\mathrm{MS_E}}{\mathrm{MS_R}+(k-1)\,\mathrm{MS_E}}, \qquad \mathrm{ICC}(2,1) = \frac{\mathrm{MS_R}-\mathrm{MS_E}}{\mathrm{MS_R}+(k-1)\,\mathrm{MS_E}+\frac{k}{n}(\mathrm{MS_C}-\mathrm{MS_E})}. \tag{6}$$

ICC(3,1) omits the run term, which is why it is invariant to a shift shared by every feature within a run and ICC(2,1) is not. We interpret the values using the bands of Koo & Li (2016), with values below 0.5 considered poor, 0.5 to 0.75 moderate, 0.75 to 0.9 good, and above 0.9 excellent.

B Supporting tables and figures

These support Section 4; each is cited from the subsection of the main text whose numbers it carries.

B.1 The pipeline

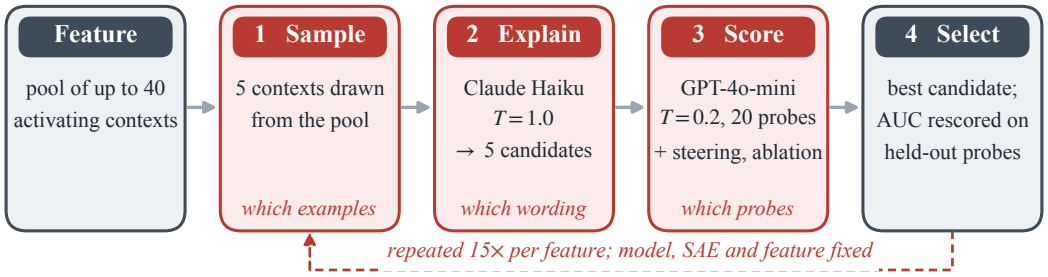

Figure 5: One run of the pipeline, with the settings used throughout. Stages in red draw randomly on every run: which examples the explainer sees, how it words its candidates, and which probes the scorer writes. These are the three sources attributed in Section 4.3; stages in gray are deterministic. The dashed path marks the repetition: each feature is run 15 times with the model, the SAE and the feature held fixed.

B.2 Label variability and the dissociation

The pilot and the main sample agree. Table 5 sets the pilot, a disjoint list of 50 features × 10 runs, beside the main sample. Score reliability is lower on the main sample and label instability slightly lower, so the pilot understated the score problem and slightly overstated the label one.

The result does not depend on the estimator. Three checks leave it intact. The primary ICC(3,1) and the absolute-agreement ICC(2,1) agree to within 0.001, so there is no run-level offset to charge for. A second sentence encoder moves mean instability from 0.581 to 0.577. The entailment measure, which counts mutual-entailment clusters, finds *more* disagreement than embeddings do, by a factor of 1.46, and places each run in a separate entailment cluster for 44 features; the two rank features alike (Spearman $\rho = 0.742$). We report the embedding measure throughout as the smaller of the two, without claiming it bounds instability from below. *Neither finds one unanimous feature.*

Table 3: The two measures of label variability, over 150 features $\times$ 15 runs. The embedding measure counts effectively distinct vectors; the entailment measure counts mutual-entailment clusters. They provide complementary automated views of dispersion, with different sensitivities to semantic similarity and entailment.

	Embedding (Vendi)	Entailment clusters
Distinct interpretations per feature	9.13	13.37
As a share of the 15 runs	60.9%	89.1%
Near-unanimous features	0 / 150	0 / 150
Ratio, entailment to embedding	1.46×	

Table 4: Stability of the pass/fail verdict, 150 features $\times$ 15 runs. The flip fraction (Section 3.3) asks whether a feature's underlying status is ambiguous. "Both sides" is the share of features with runs on either side of the threshold, which grows with the number of runs. The last column is the chance that two runs of a feature fall on opposite sides, which does not.

Verdict	Flip fraction	Both sides	Two runs disagree (mean)
Confidence vs. $\tau = 0.65$	0.127	0.547	0.174
Held-out AUC vs. chance	0.207	0.620	0.215

Table 5: Pilot and main sample under the full-stochasticity configuration. The pilot uses a disjoint feature list.

Metric	Pilot (50 $\times$ 10)	Main (150 $\times$ 15)
Instability, mean	0.652	0.581
Instability, median	0.645	0.604
Near-unanimous features	0 of 50	0 of 150
ICC(3,1), held-out AUC	0.671	0.599
ICC(3,1), confidence	0.778	0.719
Flip fraction	0.120	0.127

The pass/fail verdict often depends on the run. The dissociation of Section 4.2 reaches the verdict a reader acts on. Over fifteen runs, 54.7% of features fall on both sides of the $\tau = 0.65$ threshold, but that count grows with the number of runs and overstates what one reader risks. The run-count-free quantity is the chance that two runs of a feature disagree, amounting to about one pair in six on average (Table 4), and one in eight for the median feature.

B.3 SCORE-COMPONENT RELIABILITY

The component audit provides a further check on how the score is built. One component, ablation, never compares the model's output with the label, so a score could look stable because of it alone; auditing the components separately shows that the weak association also occurs for label-dependent AUCs (Table 6). Held-out AUC, which is computed entirely from probes written for the selected label and carries no weight in confidence, is moderately reliable on its own and its variation is likewise uncorrelated with label instability. The most reliable component is ablation, which does not compare anything with the label, and the least reliable is the steering judge, which checks the label most directly, although the steering intervention itself is weak (Appendix D).

Table 6: The part of the score that checks the label is the least reliable. Reliability of each score component, 150 features $\times$ 15 runs. Weights are those of equation 1; held-out AUC is computed after selection and carries no weight. The last column is the rank correlation between a feature's label instability and the variation of that component across its runs. *Uses label* marks whether a component's value depends on the label; ablation depends on it only through the probes it is run on.

Component	Weight	Uses label	ICC(3,1)	Median SD	ρ with instability
Selection AUC	0.60	yes	0.691	0.091	-0.050
Held-out AUC	—	yes	0.599	0.118	-0.097
Steering match	0.24	yes	0.173	0.187	$+0.235$
Ablation	0.16	via probes	0.836	0.025	-0.392
Causal (combined)	0.40	partly	0.221	0.110	$+0.260$
Full confidence	1.00	partly	0.719	0.056	-0.009

B.4 FACTORIAL ATTRIBUTION

The main text reports the factorial on label instability (Table 2). Table 7 gives the separate one-source-at-a-time experiment, and Table 8 the factorial on held-out AUC variance, which, unlike the bounded instability index, is far from its ceiling and can be decomposed into main effects and interactions. There, each source freed alone is two to three times its main effect, and about half of the gap between the additive prediction and the observed total is one negative interaction, between the example draw and the explainer. The additive prediction covers a small part of the range AUC variance can take, and single features exceed it more than threefold. On label instability, the example draw's main effect is about twice the explainer's.

Table 7: Variation with one source freed at a time. Each source is freed with the other two pinned, 50 features $\times$ 10 runs per condition.

Source allowed to vary	Label instability	Held-out AUC variance
Example draw	0.617	0.0129
Explainer	0.400	0.0134
Scorer	0.096	0.0063

Table 8: The factorial on held-out AUC variance, 50 features $\times$ 5 runs per cell. A main effect averages a source's contribution over the settings of the other two; the ratio divides the freed-alone cell by the main effect.

Source	Main effect	Freed alone	Ratio
Example draw	0.0056	0.0124	2.2×
Explainer	0.0071	0.0141	2.0×
Scorer	0.0018	0.0058	3.2×
Example draw × explainer	−0.0072	distinguishable from zero	
Example draw × scorer	−0.0012	not distinguishable	
Explainer × scorer	−0.0017	not distinguishable	
All three free, observed	0.0171	freed-alone sum 0.0323	

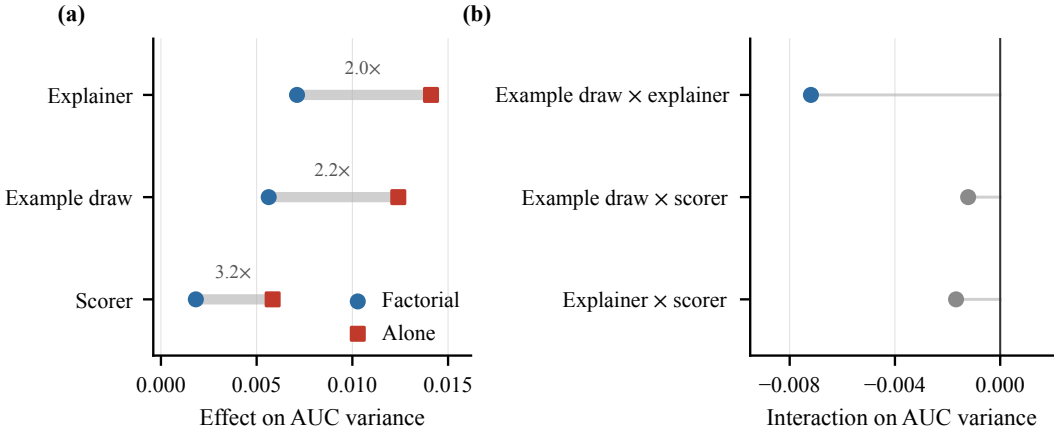

Figure 6: Factorial attribution on held-out AUC variance (Table 8). (a) Each source's factorial main effect (circles) against its effect when freed alone (squares), the quantity the isolation conditions measure; bars and ratios show the gap. (b) Two-way interactions, with blue marking the one distinguishable from zero.

B.5 Mitigations, aggregation and replication

Table 9: Mitigation conditions, 50 features $\times$ 10 runs, compared with the pilot baseline on the same features. Quality is the change in median held-out AUC; a condition is disqualified if this falls below -0.05.

Condition	Median instability	Reduction	Quality
Baseline	0.645	—	—
Top-k (deterministic examples)	0.332	+0.313	+0.013
More examples (20, not 5)	0.556	+0.088	+0.058
Greedy (explainer temperature 0)	0.690	−0.046	+0.016

Table 10: Aggregation outcomes against a matched control at equal item count. The *Cost* column gives the number of pipeline runs per feature. At the measured rate of \$0.011 per run, $k = 5$ is approximately \$0.055 per feature.

k	Cost	Label instability	Held-out AUC SD
1	1×	—	—
2	2×	1.8% better	29.2% better
3	3×	8.6% better	35.9% better
5	5×	**15.0%** better	**55.3%** better
7	7×	14.0% better	56.0% better

Table 11: The selected condition at pilot and confirmatory scale. The two feature lists are disjoint, so the confirmatory column is an out-of-sample test of a condition chosen on the pilot. Score reliability is reported although the selection rule did not require it.

	Pilot (50×10)	Confirmatory (150×15)	Baseline (150×15)
Median label instability	0.332	**0.328**	0.604
Near-unanimous features	2 / 50	6 / 150	0 / 150
Change in median AUC	+0.013	−0.012	—
ICC(3,1), held-out AUC	—	0.666	0.599
ICC(3,1), confidence	—	0.757	0.719

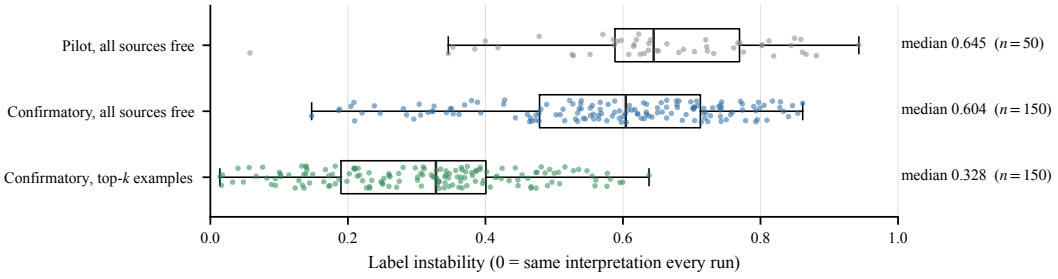

Figure 7: Per-feature label instability under three conditions, one point per feature, with boxes marking the quartiles. The pilot (50 features × 10 runs) and the confirmatory sample (150×15) are disjoint; the top-k condition was selected on the pilot and re-run on the confirmatory features. Fixing the examples moves the whole distribution, although no condition brings a typical feature near zero.

B.6 SEMANTIC VERSUS LEXICAL DISAGREEMENT

Table 12: How cross-run label pairs relate, under two judges from different providers, 1,500 non-identical pairs sampled at most ten per feature. Bold rows are the ones the judges agree on and from which we draw conclusions. The intermediate categories are not comparable. GPT-4o-mini answered both *subsumes* calibration pairs correctly and then never assigned that category to a real pair, so the two judges are not estimating the same four-way partition.

Relation	Claude Sonnet	GPT-4o-mini
Equivalent *(same concept, reworded)*	**5.1%**	**2.8%**
Subsumes *(one strictly narrower)*	14.9%	0.0%
Overlapping *(neither contains the other)*	56.3%	67.2%
Unrelated *(different concepts)*	**23.6%**	**30.0%**

B.7 EFFECT ON RETRIEVAL

Table 13: Retrieval from label indices built on different runs, 150 features, 105 run pairs. Every quantity is reported under both sentence encoders, since all are embedding-dependent (§4.1). Unchanged target inclusion means that the queried feature is always in the top ten or always absent; it does not require the same retrieved set.

	MiniLM	MPNet	Reference
Overlap@10 between two indices	0.186	0.193	0.067
Same top-ranked feature	0.112	0.118	—
Recall@10 of the queried feature	0.394	0.408	0.067
Recall flip across 15 runs	0.893	0.867	0.9995
Unchanged target inclusion	0.107	0.133	0.0005

B.8 FEATURE-LEVEL PREDICTORS

All five predictors are computed from the SAE and the feature's stored contexts, without a language-model call. *Context diversity* is the rescaled Vendi score of equation 2 over the embeddings of the feature's stored contexts, so a feature that fires on one kind of text scores near 0 and one that fires on unrelated kinds of text scores near 1. *Token diversity* is the same quantity over the token at which each context peaks. *Activation density* is the fraction of tokens on which the feature is active, and *peak activation* its approximate maximum activation, both as reported by Neuronpedia. *Live contexts* is the number of stored contexts on which the feature fires; it is included as a mechanical control, since a feature with few contexts beyond the five shown to the explainer has little room to vary. As a check that does not use embeddings, *lexical diversity* is one minus the mean pairwise Jaccard overlap between the contexts' sets of words; it correlates with instability at $\rho = 0.362$ and 0.376 under the two encoders.

Table 14: Spearman correlation of each predictor with per-feature instability, 150 features. Holm-adjusted across the five tests; a predictor counts as supported only if it holds under both encoders (§4.1). Token diversity crosses $p = 0.05$ between encoders, so we claim nothing from it. Crossing the encoders, so that diversity and instability are measured in different spaces, leaves context diversity at $\rho = 0.533$ and 0.547 against 0.539 and 0.540 uncrossed.

Predictor	ρ (MiniLM)	p (Holm)	ρ (MPNet)	p (Holm)
Context diversity	**0.539**	**<0.0001**	**0.540**	**<0.0001**
Token diversity	0.170	0.133	0.214	0.035
Activation density	0.172	0.133	0.160	0.101
Peak activation	−0.174	0.133	−0.189	0.063
Live contexts *(mechanical)*	0.071	0.391	0.057	0.490

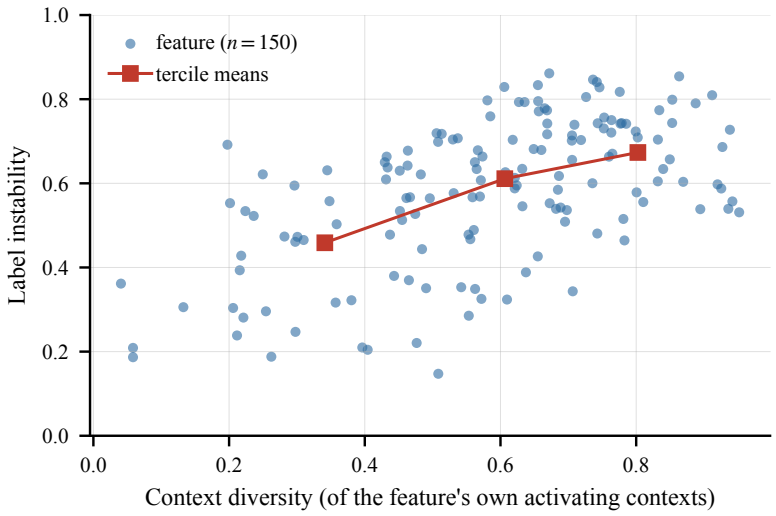

Figure 8: Diversity of a feature's own activating contexts against its label instability, one point per feature (150 features $\times$ 15 runs; Spearman $\rho = 0.539$, and 0.540 under the second encoder). Squares are tercile means. The predictor requires no language-model call, but even the most homogeneous tercile averages an instability of 0.459.

B.9 Uncertainty of headline numbers

Table 15 gives 95% intervals for the numbers the main text reports as point estimates. All but one resample features, the unit of analysis throughout, with 10,000 draws; they are computed from the exported per-feature results by `analysis/headline_intervals.py` and the analysis scripts. The null correlation between label instability and held-out AUC variation rules out an association stronger than about 0.18 in either direction. Top-k reduces instability for 141 of these 150 features.

Table 15: Headline numbers with 95% intervals. Intervals are percentile bootstraps over features, except †, a Fisher-z interval. Pilot rows use 50 features × 10 runs; one-at-a-time and factorial rows use the designs of Section 3.2.

Quantity	Estimate	95% interval
Magnitude (Section 4.2)		
Label instability, mean	0.581	[0.553, 0.608]
Label instability, median	0.604	[0.567, 0.634]
Correlation with held-out AUC SD (141 features)	−0.014	[−0.179, 0.152]†
Flip fraction, confidence	0.127	[0.073, 0.180]
Two runs disagree on the verdict, mean	0.174	[0.144, 0.205]
Attribution (Section 4.3)		
One at a time: example draw	0.617	[0.565, 0.666]
One at a time: explainer	0.400	[0.349, 0.453]
One at a time: scorer	0.096	[0.074, 0.120]
Factorial main effect: example draw	0.467	[0.420, 0.514]
Factorial main effect: explainer	0.220	[0.183, 0.256]
Factorial main effect: scorer	0.038	[0.018, 0.058]
Factorial interaction: example draw × explainer	−0.417	[−0.486, −0.349]
Mitigation (Section 4.4)		
Pilot mean instability: baseline	0.652	[0.605, 0.695]
Pilot mean instability: greedy decoding	0.632	[0.577, 0.681]
Pilot mean instability: more examples (20, not 5)	0.512	[0.460, 0.563]
Pilot mean instability: top-k examples	0.329	[0.285, 0.374]
Top-k on the 150 main features, mean instability	0.308	[0.284, 0.332]
Top-k paired reduction in mean instability	47.0%	[43.3%, 50.8%]
Consequences (Section 4.5)		
Retrieval: overlap@10 between two runs' indices	0.186	[0.179, 0.195]
Retrieval: queries whose recall flips across runs	0.893	[0.840, 0.940]

C Reproducibility details

Model, SAE and features. Gemma-2-2B (`google/gemma-2-2b`) at layer 12, with the Gemma Scope canonical residual-stream SAE (`gemma-scope-2b-pt-res-canonical`, `layer_12/width_16k/canonical`, average L_0 of 82), served through SAELens (Bloom et al., 2024). Feature indices correspond to Neuronpedia's `gemma-2-2b/12-gemmascope-res-16k`. The model runs in bfloat16 on GPU and float32 on CPU; SAE weights are float32 and projections are computed in float32. Each feature stores at most 40 activating contexts, ordered by activation strength in the Neuronpedia release used for this study.

Model snapshots and decoding. Explainer: Claude Haiku, snapshot `claude-haiku-4-5-20251001`, temperature 1.0, up to 5 candidates. Scorer: `gpt-4o-mini` through the OpenAI Responses API, temperature 0.2, 20 probes per candidate. Steering judge: Claude Haiku at temperature 0, the same snapshot as the explainer. Disagreement judges (Section 4.5): `claude-sonnet-4-6` and `gpt-4o-mini`. Output budgets depend on the request size, with $\max(768, \min(4096, 180c))$ tokens for c candidates, so 900 at $c = 5$; $\max(768, \min(4096, 120p))$ for p probes, so 2,400 at $p = 20$; and 512 for the judges. Top-p is left at the provider default and never set. *The scorer and the Sonnet judge use undated aliases, so a rerun may not reach the same model weights.*

Seeds. A seed is drawn per run and enters the cache key alongside provider, prompt, model, temperature and `max_tokens`, but is *not* forwarded to either provider. Anthropic exposes no seed parameter and neither does the OpenAI Responses API. A seed is therefore a nonce that distinguishes cached draws. Two consequences follow. A configuration whose seeds are all pinned replays cached responses exactly, which is why the all-pinned factorial cell is near zero by construction (Appendix D); and temperature-zero decoding is not guaranteed reproducible, which is what the greedy arm measures under the experimental configuration reported here.

Retries and refusals. Provider calls retry up to three times with exponential backoff (2^{attempt} seconds). Authentication and access failures are fatal and abort the sweep. A content refusal does not raise an exception. When the scorer declines, nothing parses, and probe generation falls back to templates, whose scores would reflect the fallback text; such runs are detected and excluded. One run of $10{,}000$ was excluded on this rule.

Causal channel. Steering adds magnitude $\times$ direction to the layer-12 residual stream through a forward hook, with magnitude 1.0, and generates 32 new tokens greedily (`do_sample=False`); Claude Haiku at temperature 0 then rates the continuation against the label in $[0, 1]$. Ablation projects the feature out and reports $1 - \exp(-|\overline{\Delta \text{logits}}|)$, which does not compare the output with the description. Both use the candidate's first 2 positive probes as intervention prompts, so both depend on the label through those probes. A magnitude of 1.0 adds the decoder direction as if the feature were active at 1, whereas stored activations reach the tens (Neuronpedia reports a peak near 27 for the feature of Figure 1), so steering is a weak intervention in this pipeline. The weights of the confidence score and the threshold τ were fixed before collection.

Analysis settings. Sentence encoders: `all-MiniLM-L6-v2` (primary) and `all-mpnet-base-v2`, both on CPU. The Vendi kernel is the cosine similarity of unit-normalized embeddings. Entailment clustering uses `microsoft/deberta-large-mnli` with an entailment threshold of 0.5 in both directions. Bootstrap intervals resample *features*, with $10{,}000$ draws for means and $1{,}000$ for rank correlations; every analysis seed is a fixed constant recorded in the script that uses it.

Run ledger. $10{,}000$ runs: pilot 500 (50×10); confirmatory $2{,}250$ (150×15); Experiment 2 isolation conditions $1{,}500$ ($3 \times 50 \times 10$); Experiment 3 mitigation arms $1{,}500$ ($3 \times 50 \times 10$); top-k confirmatory $2{,}250$ (150×15); factorial $2{,}000$ (8 cells $\times$ 50×5). At the measured rate of approximately \$0.011 per run, total API spend was about \$110.

Prompts. The scorer and judge prompts are reproduced in full; the explainer prompt is given in full in the supplementary repository (`src/auto_interp/pipeline/hypothesis.py`) and abridged here to its instruction and output contract. Its wording ("top activating contexts") is the same in every condition, including those in which the contexts are sampled at random.

```
Generate {count} positive and {count} negative probe
sentences for the semantic label below.  The positives
should clearly express the label.  The negatives should
avoid it while staying topically plausible.  Return one
sentence per line prefixed with POS: or NEG: Label:  {label}

You are checking whether a continuation reflects a semantic
label.  Answer with a single number between 0 and 1.  Label:
{label} Continuation:  {continuation}

You are an interpretability researcher analyzing a sparse
autoencoder feature.  Given the top activating contexts
below, propose up to {num_candidates} concise semantic
hypotheses about what this feature represents.  Favor
descriptions of the underlying concept over restating
incidental surface tokens.  [Nine bulleted constraints
follow, directing the model toward broad category-level
labels, requiring at least one such label among the
candidates, and naming exceptions for concrete surface
triggers.]  Return ONLY a JSON array ...each element
an object with keys "label" (a short noun phrase, 3--10
words), "confidence" (a float in [0,1]) and "rationale" (one
sentence, at most 16 words).  Contexts:  {contexts}
```

D LIMITATIONS

Scope. All results concern Gemma-2-2B at layer 12 under a single SAE width; we claim no generalization across layers, widths or model families.

Semantic uncertainty and correctness. We measure variation in the semantic claims selected by the pipeline. These measurements do not establish which claims correctly or fully describe a feature. A planned human validation was not run, so the gap between our two automated measures of label variability is unadjudicated, and claims from the judge taxonomy are restricted to categories for which the judges reach consistent conclusions.

Floor-censored features. Of the nine features flagged as floor-censored, eight have held-out AUC exactly 0.5 in every run; feature 2201 has mean AUC about 0.498 and SD about 0.006. Their restricted score variation should not be interpreted as evidence of a stable semantic description.

Attribution design. The factorial's all-pinned cell is near zero by construction, since a pinned source replays a stored response and the providers are never queried twice, so it says nothing about whether temperature-zero decoding is deterministic. Because prompts enter the cache key, changing the examples can trigger fresh explainer and scorer calls. The conditions therefore do not isolate every source of provider randomness.

Score composition. Steering adds the feature's decoder direction at unit scale, which is small next to the feature's own activations, so the steered continuation often barely changes; this likely explains the steering judge's low reliability, and the causal channel should be read with that in mind.

Instrument or feature. We ask for one label per feature, as published indices do. Different runs may describe genuinely different facets, or the procedure may fail to summarize the available evidence consistently. The context-diversity association alone does not separate these possibilities. Some features may benefit from multiple descriptions; this study does not evaluate that remedy.

Semantic coverage under top-k. Restricting the explainer to a feature's strongest contexts improves stability, but Gur-Arieh et al. (2025) report that descriptions from top-activating inputs miss a feature's causal effect on outputs, which held-out probe AUC would not detect.