NON-GAUSSIANITY REVEALS INTERPRETABLE DIRECTIONS IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Finding human-interpretable directions in language-model activations is important for understanding and controlling model behavior. These directions are usually discovered through indirect learning objectives such as sparse reconstruction, but it remains unclear what existing statistical structure distinguishes interpretable directions and whether that structure can be used directly for feature discovery. We find that non-Gaussianity provides such a signature: projection distributions of natural-text residual-stream activations along sparse autoencoder (SAE) decoder directions are substantially more non-Gaussian than those along random directions, with most contexts forming a concentrated background and recurring patterns often occupying one tail. This raises a natural question: *if interpretable directions tend to have non-Gaussian projection distributions, can we discover them by searching for non-Gaussian directions directly in activation space, without training another sparse dictionary?* To test this hypothesis, we introduce ICA LENS, a practical workflow for **efficient** discovery, **reliable** annotation, and **faithful** evaluation of non-Gaussian directions in LLM residual-stream activations. Across GPT-2 Small, Gemma 2 2B, and Qwen 3.5 9B Base, we find that: (1) the recovered non-Gaussian directions capture coherent lexical, syntactic, and semantic patterns under both human and LLM-based evaluation; (2) non-Gaussianity is positively associated with both human-label confidence and automated interpretability; and (3) the resulting coordinates are competitive with SAEs across interpretability, sparse reconstruction, concept prediction, and targeted probe perturbation, while fitting in minutes per layer and using 7–43$\times$ less memory. These results show that non-Gaussianity is not merely a statistical property of learned features, but a direct unsupervised signal for discovering interpretable and useful structure in activations.

1 INTRODUCTION

Understanding the internal representations of large language models (LLMs) is a central goal of mechanistic interpretability (Olah et al., 2020; Elhage et al., 2021; Gurnee et al., 2023; Park et al., 2023). Sparse autoencoders (SAEs) have become a dominant approach to this problem by decomposing activations into sparse latents over an overcomplete dictionary of learned directions (Huben et al., 2024; Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2025). Their success raises a more basic question: what makes an interpretable direction discoverable in the first place? Current methods usually define features through a learning objective such as sparse reconstruction, but it remains unclear whether the resulting directions also share an intrinsic statistical signature in the underlying activation distribution. If such a signature exists, interpretable structure may be discoverable directly from activation geometry rather than only as the outcome of dictionary learning.

A simple intuition suggests what this signature might look like. Many interpretable directions are selective: they respond strongly to a recurring lexical, syntactic, semantic, or discourse pattern while remaining relatively inactive for most other contexts. Across natural-text activations, such a direction should therefore behave differently from a typical random direction. Most contexts may form a concentrated background, while the contexts sharing the relevant pattern occupy a sparse or asymmetric tail. This geometry naturally produces a non-Gaussian projection distribution.

We find this pattern in public SAE dictionaries (Section 2.1) and in directions constructed directly from coherent concepts (Appendix A.1). For public SAE directions, projection distributions are

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.

substantially more non-Gaussian than those of random directions, even though SAEs are trained for sparse reconstruction rather than explicit non-Gaussianity. Figure 1 shows a representative example: a random direction produces an approximately Gaussian background with semantically mixed extremes, whereas an interpretable direction separates a coherent group of related tokens into one tail. These observations motivate our central hypothesis: ***non-Gaussianity itself can serve as an unsupervised signal for discovering interpretable directions in language-model activations.***

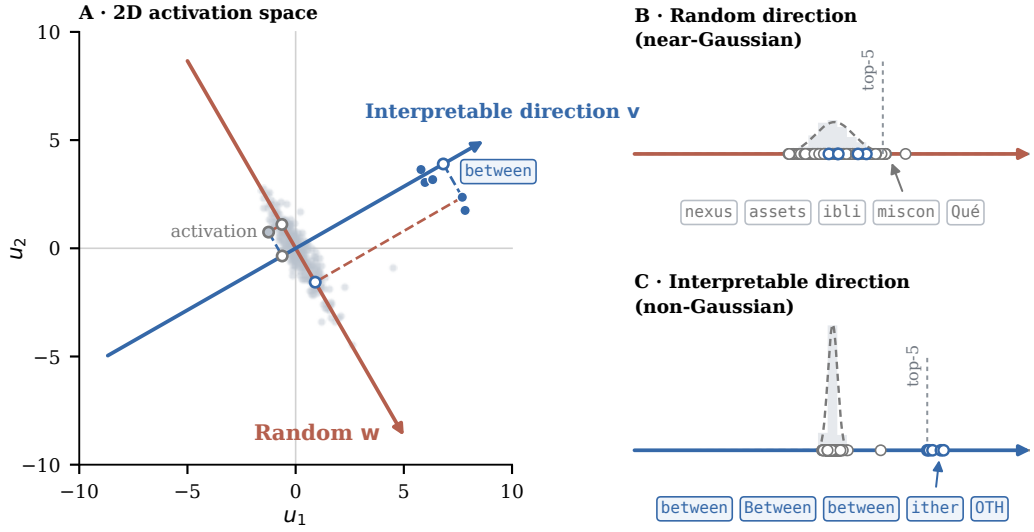


Figure 1: **Interpretable features leave a non-Gaussian signature.** (A) Real GPT-2 token activations projected onto a random direction $\mathbf{w}$ and an interpretable direction $\mathbf{v}$. (B) The random projection is close to Gaussian and its top-scoring tokens are semantically mixed. (C) The interpretable direction isolates a coherent set of related tokens in a non-Gaussian tail.

The hypothesis also suggests a practical opportunity. SAEs are powerful when a large, fine-grained feature dictionary is needed, but training a new dictionary for every model, layer, and activation site can be expensive (Lieberum et al., 2024). For early-stage analysis, however, a compact set of interpretable coordinates may already suffice to identify promising layers, inspect model states, probe concepts, and test lightweight interventions. Direct feature discovery from activation statistics could therefore provide a useful first lens before larger-scale dictionary learning.

To test this hypothesis, we introduce ICA LENS, a practical workflow for discovering, annotating, evaluating, and using non-Gaussian directions in LLM residual-stream activations. After centering and whitening the activations, GPU-parallel symmetric FastICA searches for a non-Gaussian basis (Hyvarinen & Oja, 2000). We combine top-activating contexts, vocabulary readouts, and controlled counterfactuals to form and test component interpretations. Because FastICA assigns component signs arbitrarily while coherent semantics often concentrate in one tail, we use skewness to select the semantic tail before interpretability evaluation. This correction addresses a long-overlooked sign-sensitive bias that substantially understated ICA interpretability in the earliest direct ICA–SAE comparison (Huben et al., 2024).

Across GPT-2 Small, Gemma 2 2B, and Qwen 3.5 9B Base, we find that non-Gaussian directions are both interpretable and useful. First, our human audit shows that the recovered non-Gaussian directions capture coherent lexical, syntactic, semantic, positional, and contextual patterns (Section 2.2). At scale, automated evaluation also shows that skewness-selected ICA directions achieve interpretability comparable to public SAE features. We further analyze the relationship between non-Gaussianity and interpretability and find that mean automated interpretability rises as FastICA recovers increasingly non-Gaussian directions and then saturates; components with greater non-Gaussianity are also associated with higher human confidence and automated interpretability score.

Second, the recovered coordinates are useful beyond feature labeling (Section 2.3). Their component mixtures can be used to inspect new activations, and individual directions support generation-time steering of model behavior. Across held-out downstream datasets, they are competitive with

public SAEs in sparse reconstruction and sparse concept probing, and achieve higher targeted probe perturbation scores under small-to-medium feature budgets. Importantly, these results do not require a large learned dictionary. With one million activations per layer, our GPU implementation fits an ICA basis in under one minute for GPT-2 Small, about 5.5 minutes for Gemma 2 2B, and 12–13 minutes for Qwen 3.5 9B Base, while using 7–43× less parameter memory per layer than the matched SAEs.

Overall, these results suggest a different view of feature discovery in language models. **Interpretable directions have an intrinsic higher-order statistical signature. This signature predicts interpretability and can itself be optimized to recover interpretable, useful coordinates—without first learning a sparse dictionary.** This reframes non-Gaussianity from a descriptive property of learned features into a direct signal for discovering them.

2 From Non-Gaussianity to Feature Discovery

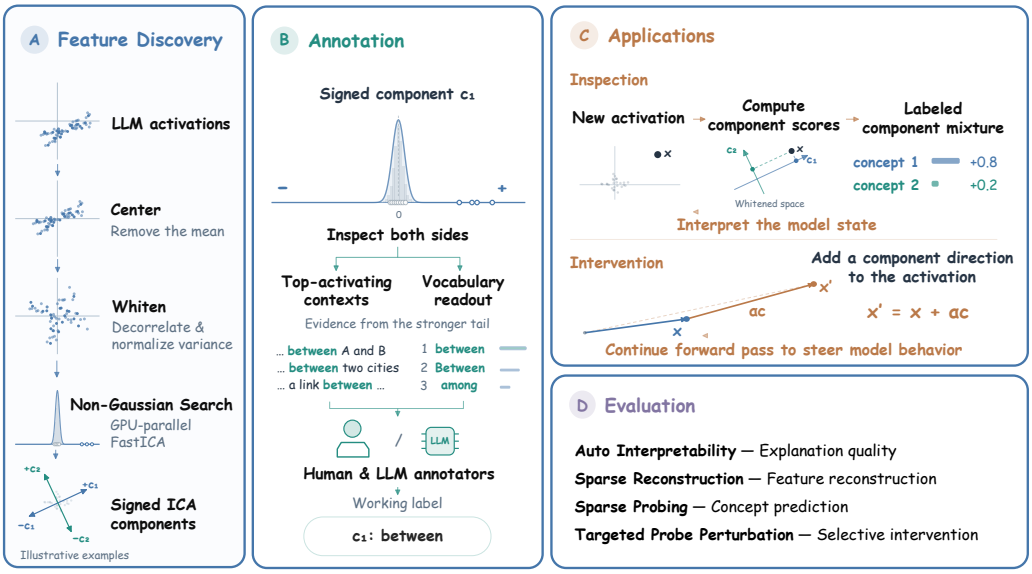


Figure 2: **ICA Lens workflow.** (A) GPU-accelerated symmetric FastICA discovers non-Gaussian directions from centered, whitened activations. (B) Top-activating contexts and vocabulary readouts support annotation. (C) Coordinates enable state inspection and editing. (D) Evaluation covers interpretability, sparse reconstruction, sparse probing, and targeted probe perturbation.

We now investigate our hypothesis using a concrete analysis pipeline, ICA Lens. Given residual-stream activations, it discovers signed non-Gaussian components, annotates them using activation contexts and vocabulary readouts, and exposes the resulting coordinates for inspection and intervention. Figure 2 summarizes the workflow.

Our analysis proceeds through three questions, moving from statistical discovery to semantic interpretation and practical utility: **(1) Discovery:** Can non-Gaussian directions be recovered directly from residual-stream activations? (Section 2.1) **(2) Interpretability:** Do the recovered directions correspond to coherent, recognizable patterns? (Section 2.2) **(3) Utility:** Do the resulting coordinates support useful readout and intervention? (Section 2.3)

2.1 Non-Gaussian Structure Can Be Recovered Directly

We first ask whether the statistical structure observed in SAE directions can be recovered directly from natural-text residual-stream activations, without learning another sparse dictionary. ICA provides a simple test of this question: after removing first- and second-order structure through centering and whitening, it searches for directions that differ maximally from a Gaussian distribution.

Given cached activations $X^\ell \in \mathbb{R}^{n\times d}$ from layer ℓ, we subtract the empirical mean to obtain X_c. Writing the covariance eigendecomposition as $\frac{1}{n}X_c^\top X_c = U\Lambda U^\top$, we whiten the activations using $Z = X_c U\Lambda^{-1/2}$, so that Z has zero empirical mean and identity empirical covariance. We then fit a

full-dimensional basis with symmetric FastICA (Hyvarinen, 1999), which updates all components in parallel and jointly orthogonalizes them after each iteration.[1] The resulting orthogonal rotation $W = [w_1, \ldots, w_d] \in \mathbb{R}^{d \times d}$ defines component scores $y_j = Zw_j$ across tokens. We optimize the logcosh contrast, which increases as the projection distribution departs from a standard Gaussian. Here, *non-Gaussianity* refers to the distribution of scores across natural-text activations along a direction, including heavy-tailed or asymmetric distributions. Because we retain all d components, ICA produces a complete basis rather than an overcomplete dictionary. Appendix B.1 gives the full activation-collection and symmetric FastICA pipeline.

LLM residual streams contain strong, recoverable non-Gaussian structure. To compare this structure across ICA, SAE, PCA, and random directions, we measure the excess kurtosis of their projection scores as follows[2]. A Gaussian distribution has population excess kurtosis zero, while positive values indicate heavier tails.

$$\kappa_j = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_{ij} - \bar{y}_j)^4}{\left[\frac{1}{n}\sum_{i=1}^{n}(y_{ij} - \bar{y}_j)^2\right]^2} - 3, \qquad \bar{y}_j = \frac{1}{n}\sum_{i=1}^{n} y_{ij}. \tag{1}$$

Figure 3 reveals a consistent pattern across models and layers. Random projections remain close to the Gaussian reference, while PCA stays much closer to the random baseline and captures only weak heavy-tailed structure. In contrast, SAE decoder directions are substantially more heavy-tailed despite being trained for sparse reconstruction rather than non-Gaussianity. ICA, which searches for non-Gaussianity directly, recovers the strongest heavy-tailed projections across all three models. These results show that modern LLM residual streams contain strong higher-order structure that is directly recoverable. Appendix A.1 shows another method to construct interpretable non-Gaussian directions and compares them with ICA directions; Appendix A.2 further compares ICA and SAE structure through directional overlap and tokenwise responses.

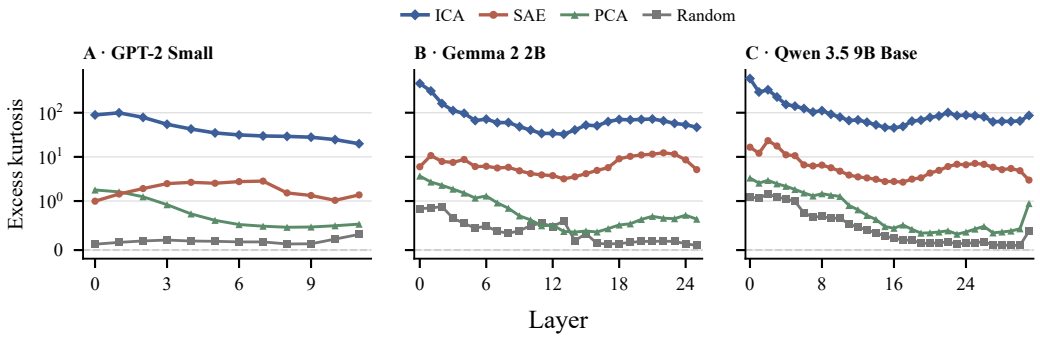


Figure 3: **ICA recovers the most heavy-tailed directions across models and layers.** Mean excess kurtosis of projections onto unit-norm ICA, SAE, PCA, and random directions, measured on one million held-out residual-stream activations per layer. Random projections remain near the Gaussian reference, while ICA consistently produces the largest excess kurtosis.

This structure can be recovered efficiently at LLM scale. We implement symmetric FastICA in PyTorch to update all components in parallel on GPUs. We use 50 iterations as the default operating point, where population-level non-Gaussianity has largely saturated; extending to 200 iterations further improves exact cross-seed component matching when identity stability is the priority (Appendix B.4). With one million activations per layer on a single NVIDIA GB10, fitting takes under one minute for GPT-2 Small, about 5.5 minutes for Gemma 2 2B, and 12–13 minutes for Qwen 3.5 9B Base, while the resulting bases use 7–43× less parameter memory per layer than the matched public SAEs. Appendix B.2 gives the full memory, latency, and end-to-end fitting comparison. Appendices B.3–B.5 analyze layer-wise fitting dynamics, cross-seed and iteration-

[1]The resulting axes are orthogonal in whitened space; their corresponding directions in the original residual-stream space need not be orthogonal.

[2]Logcosh contrast and excess kurtosis follow closely aligned optimization trajectories and show similar convergence behavior (Appendix B.5). We use kurtosis in the following analyses because its Gaussian reference is simple and its larger visual dynamic range makes differences across directions easier to interpret.

budget stability, and the distinction between population-level and component-level convergence, while Appendix F lists the exact matched public SAE checkpoints.

These results answer the discovery question: strongly non-Gaussian directions can be recovered directly and efficiently from natural-text residual-stream activations. We next ask whether these directions correspond to coherent, human-recognizable activation patterns.

2.2 Many Non-Gaussian Directions Are Interpretable

We evaluate the interpretability of these directions in two complementary ways. Human annotation provides a direct assessment, with human confidence as a quantitative measure. Automated annotation allows us to evaluate more components efficiently, with automated interpretability scores as a second quantitative measure. We then use both measures to examine the relationship between interpretability and non-Gaussianity.

2.2.1 From Components to Working Interpretations

A few coherent top examples are not sufficient to establish interpretability, since a label can be fit post hoc to a small set of extreme activations. We therefore treat component labels as hypotheses that must be supported by multiple sources of evidence and survive controlled counterexamples.

Component profiles combine contextual and vocabulary evidence. We use two sources of evidence to annotate each component. (1) Activation contexts. For component j, FastICA returns an axis w_j in whitened activation space, with score $y_{ij} = w_j^\top z_i$ for token activation z_i. The highest-scoring contexts show *when* a component is active. (2) Vocabulary readouts. Mapping the component back to the residual stream gives a writing direction d_j, which we decode with Logit Lens (nostalgebraist, 2020) and R-Lens (Blank et al., 2026) to read *what* it tends to write. Table 1 shows some examples.

Table 1: **Representative ICA components.** Top-activating contexts and vocabulary readouts support the same working label. Underlining marks the scored token.

Component	Working label	Top-activating contexts	Logit Lens	R-Lens
GPT-2 L8/C396	Programming/API specification language	#1 in Java interface that returns class or subclass type #2 require all modules ending in ".spec" from #3 "categories" is not an array, but object	#1 "JSON" #2 "syntax" #3 "initialization"	#1 "JSON" #2 "syntax" #3 "tuple"
Gemma 2 2B L20/C1154	1960s–1970s American culture/history	#1 76 by then President Gerald Ford, and #2 at the height of The Summer of Love. #3 started for the Yankees against the Reds, whose	#1 "Vietnam" #2 "hippie" #3 "seventies"	#1 "Vietnam" #2 "hippie" #3 "Vietnam"
Qwen 3.5 9B Base L20/C352	Both	#1 3 weeks. Both the rats' behavior #2 creative nonfiction. Both graduate and or flagship #3 copies, and that both that copyright notice	#1 "sides" #2 "sexes" #3 "/all"	#1 "sides" #2 "sexes" #3 "这两种"

Iterative counterfactual tests refine working labels. A human or LLM annotator first summarizes the evidence in a working label, then tests the label on controlled prompt pairs that differ in the proposed concept while preserving other aspects of the input. The annotator revises the label when the tests reveal a mismatch and repeats this process until they can assign a label with confidence. For example, for a component labeled "children's educational content," we compare a picture *book* for preschoolers with a reference *book* for engineers. If the label is informative, the component should respond much more strongly to the first prompt despite the shared target word. Appendix C.2 gives the complete annotation workflow, sentence-level traces, controlled counterfactual checks, confidence rubric, and all 150 randomly sampled component annotations.

2.2.2 Human Audits Reveal Interpretable, Often One-Sided Semantics

ICA recovers many semantically interpretable directions. We randomly sample 150 components from the 50-iteration fits, 50 from each model. Of these, 123 receive high-confidence human labels: 82% for GPT-2 Small, 84% for Gemma 2 2B, and 80% for Qwen 3.5 9B Base. Human-label confidence also rises with excess kurtosis, a relationship we analyze quantitatively in Section 2.2.4.

ICA semantics are often one-sided. Although every ICA component has two signed tails, the human audit reveals that a coherent semantic pattern is often concentrated in one tail, while the

opposite tail is diffuse rather than an opposing concept. Consistent with this observation, strongly non-Gaussian components also tend to have more asymmetric score distributions. Across the 150 audited components, excess kurtosis is strongly correlated with absolute skewness ($\rho_s = 0.92$). This suggests that much of the heavy-tailed structure recovered by ICA is asymmetric rather than equally distributed across both tails. We therefore use skewness computed on the fitting activations to identify the tail that drives the distributional asymmetry, and use that tail for subsequent analysis and evaluation. A positive skew selects the positive tail, while a negative skew selects the negative tail. Appendix C.1 details the skewness-based orientation and tie-breaking rules.

2.2.3 Automated Interpretability Evaluation

LLMs annotate components and score interpretability at scale. Following the method and evaluation protocol of Huben et al. (2024), we use an explanation–prediction pipeline to evaluate interpretability at scale. An LLM annotator receives text fragments with their component scores and writes a short explanation of the activation pattern. A separate LLM predictor then uses this explanation to predict token-level component scores on held-out fragments. The automated interpretability score is the Pearson correlation between predicted and observed scores, so a high value indicates that the explanation generalizes to unseen examples.

A long-overlooked sign ambiguity systematically biases ICA evaluation downward. We uncover a consequential failure mode in the earliest ICA–SAE comparison of Huben et al. (2024): the evaluation considers only the positive tail of each ICA component. Because FastICA assigns signs arbitrarily, this can evaluate the diffuse tail even when the opposite tail contains a clear semantic pattern, systematically underestimating ICA and making it substantially less interpretable than it is.

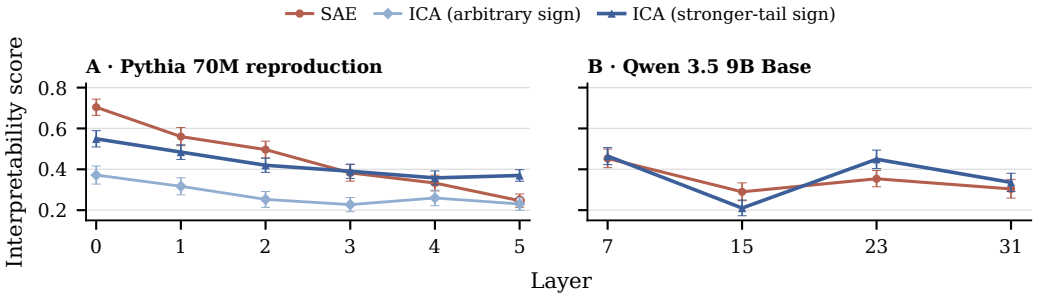


Figure 4: **ICA was underestimated by a sign-sensitive evaluation.** (A) On Pythia 70M, arbitrary-sign evaluation reproduces the reported gap to SAE; stronger-tail orientation recovers much of it. (B) On Qwen 3.5 9B Base, sign-corrected ICA is competitive with SAE. Points show means over 150 components with 95% intervals.

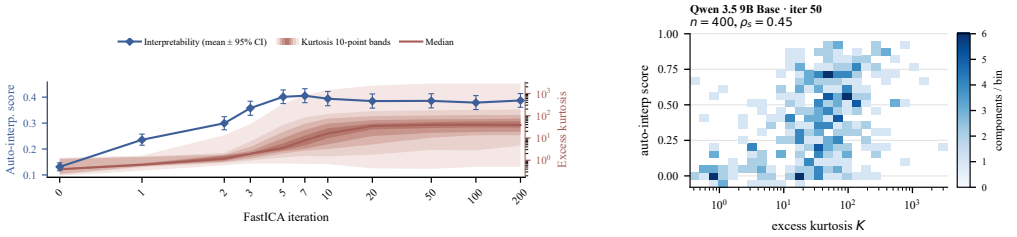
Figure 4 isolates this effect. We first reproduce the original Pythia 70M comparison using the released setup and implementation,[3] recovering the previously reported gap between SAE and ICA (*ICA, arbitrary sign*). We then evaluate each component using the tail selected by the procedure described above (*ICA, stronger-tail sign*). With this correction, much of the gap disappears and ICA becomes competitive with SAE. The same sign-corrected evaluation on Qwen 3.5 9B Base also finds ICA competitive with the public SAE baseline. Appendix C.3 specifies the explanation–prediction pipeline and matched feature-sampling rules, while Appendix C.4 tests the comparison across ICA fitting budgets, training corpora, and evaluator models.

2.2.4 Revisiting the Link Between Interpretability and Non-Gaussianity

The results above show that many ICA components are interpretable. We next ask whether interpretability is systematically related to non-Gaussianity itself. We track the same 400 sampled Qwen 3.5 9B Base components (100 each from layers 7, 15, 23, 31) throughout FastICA fitting, recording each component's sign-corrected automated interpretability score and excess kurtosis. Figure 5 shows the relationship at two levels. Across fitting iterations, mean automated interpretability rises as FastICA finds increasingly non-Gaussian directions and then saturates. Across components at

[3]We follow the released ICA baseline at https://github.com/HoagyC/sparse_coding/blob/main/autoencoders/.

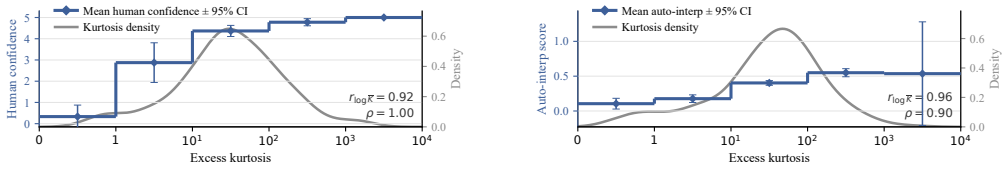
iteration 50, excess kurtosis also remains positively associated with automated interpretability, with a Spearman correlation of $\rho_s = 0.45$. Thus, both the fitting dynamics and the component-level comparison show that stronger non-Gaussianity is associated with greater interpretability. Appendix C.5 examines apparent exceptions and identifies two evaluator failure modes, suggesting that some outliers reflect limitations of automated evaluation rather than genuine violations of the broader trend.



(a) Across FastICA fitting iterations.

(b) Across components at iteration 50.

Figure 5: **Interpretability rises with non-Gaussianity.** (A) Mean automated interpretability rises during FastICA fitting and then saturates as non-Gaussianity increases. (B) At iteration 50, excess kurtosis remains positively associated with automated interpretability across individual components.



(a) Human-label confidence.

(b) Automated interpretability.

Figure 6: **Interpretability increases across excess-kurtosis bins.** Blue lines show bin means; gray curves show component density. (A) Human-label confidence for 150 audited components across three models. (B) Sign-corrected automated interpretability for 400 Qwen 3.5 9B Base components.

The relationship becomes clearer when components are compared across broad ranges of non-Gaussianity. In practice, components with excess kurtosis above 1 are usually interpretable for human annotators, and those above 10 can generally be labeled with high confidence. This suggests that real interpretability changes across broad kurtosis ranges rather than point by point. We therefore group components into excess-kurtosis bins. Figure 6a reports the mean human-label confidence in each bin for the 150 audited components from all three models. Figure 6b reports the mean sign-corrected automated interpretability score for the 400 sampled Qwen 3.5 9B Base components at iteration 50. Despite using different component sets and evaluation methods, human confidence and automated interpretability score show the same threshold-like trend. Most ICA components lie in the 10–100 range, where both measures are substantially higher than in the low-kurtosis bins.

Together, these results answer the interpretability question: many non-Gaussian directions exhibit coherent, human-recognizable activation patterns. We next ask whether these directions are also useful coordinates for representing and intervening on model activations.

2.3 Non-Gaussian Directions Are Useful Representation Coordinates

We evaluate three increasingly demanding properties. **Sparse reconstruction** asks whether a few coordinates preserve the activation. **Sparse probing** asks whether a few coordinates predict concept labels. **Targeted probe perturbation (TPP)** asks whether ablating coordinates associated with one concept disrupts its probe more than unrelated probes. For sparse probing and TPP, methods are compared at the same top-k coordinate budget. Matching k controls the number of directions that must be selected and manipulated, measuring utility per selected direction. Reconstruction is evaluated across all model layers and six datasets; probing and TPP use two representative layers per model across eight and two datasets, respectively. All datasets are different from the fitting data, testing generalization to downstream tasks. Appendix E.1 specifies the datasets, layer coverage, aggregation, feature budgets, and evaluation definitions used throughout these comparisons.

2.3.1 SPARSE RECONSTRUCTION

A few non-Gaussian coordinates preserve substantial activation information. Although ICA is not fitted with a reconstruction objective, retaining its largest coordinates reconstructs held-out activations better than PCA and random bases at sparse budgets across all three models. The resulting cosine similarity also approaches that of the public SAEs using only a small number of coordinates (Figure 7). Appendix E.2 disaggregates the comparison across six evaluation domains and model depth and reproduces the same broad ordering under normalized MSE as an alternative metric.

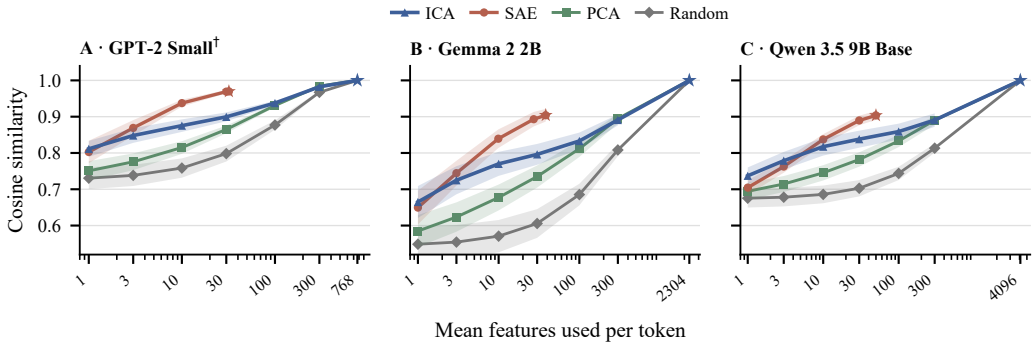


Figure 7: **Non-Gaussian directions support sparse reconstruction.** Top-k reconstruction cosine on held-out activations. Lines show means ± 1 std across six datasets and all model layers.

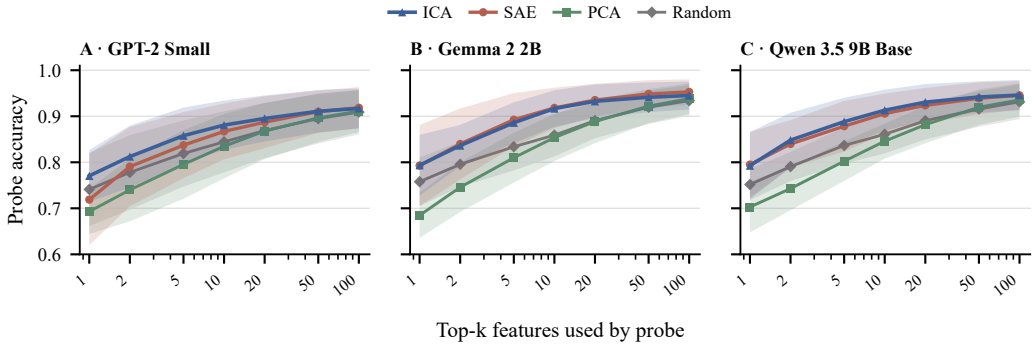


Figure 8: **Non-Gaussian directions support sparse concept readout.** SAEBench sparse-probing accuracy using the top-k features.

2.3.2 SPARSE CONCEPT READOUT

Non-Gaussian coordinates concentrate concept-relevant information. On SAEBench sparse probing, probes using only a few ICA coordinates achieve strong concept-prediction accuracy (Figure 8). ICA outperforms the public SAEs on GPT-2 Small and remains competitive on Gemma 2 2B and Qwen 3.5 9B Base. This shows that the recovered directions do more than admit human-readable labels. They also provide sparse access to task-relevant information. On Gemma 2 2B layer 12, an additional comparison with compact Matryoshka SAE variants (Bussmann et al., 2025) and the training-light ITDA baseline (Leask et al., 2025) shows ICA outperforming both across the tested feature budgets. Appendix E.3 provides this comparison, the full SAEBench protocol, and dataset-level results across all eight tasks.

2.3.3 SELECTIVE PROBE-LEVEL INTERVENTION

Sparse probing establishes that concept information can be read from these coordinates; TPP asks whether it can also be selectively disrupted. We ablate the top-k coordinates associated with a target probe and compare the resulting accuracy drop with the average drop on non-target probes. Across all three models and evaluated budgets, ICA produces consistently higher TPP scores than the public SAE baselines (Figure 9), meaning that its ablations affect the target probe more selectively. Appendix E.4 gives the full TPP definition and separates intended from unintended effects at the

dataset level. Across models, ICA's advantage is driven primarily by larger target-probe effects, while unintended effects remain comparatively small.

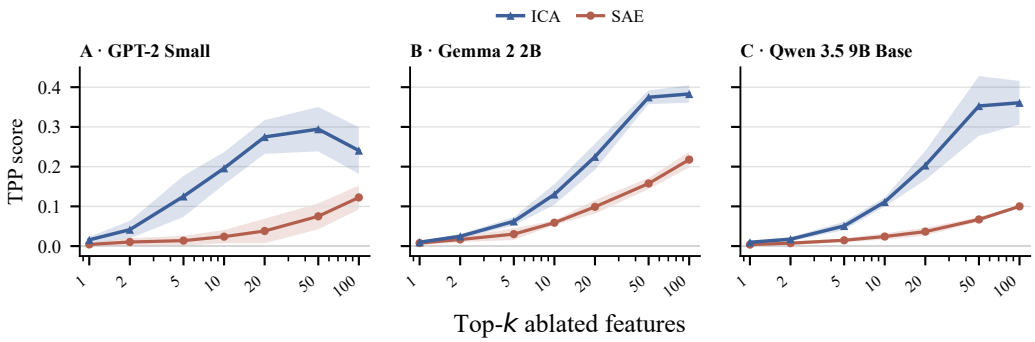


Figure 9: **ICA components support selective intervention.** TPP scores under top-k feature ablation.

Together, these evaluations answer the utility question: a small number of non-Gaussian directions preserve activation information, support concept readout, and enable selective probe-level intervention. Appendix D applies the labeled non-Gaussian directions directly, using them to decompose contextual polysemy and token-embedding structure and to perform generation-time steering and clamping.

Sparse Feature Learning for LLM Interpretability. Sparse autoencoders (SAEs) have become a standard tool for decomposing language-model activations into interpretable feature coordinates, with work spanning monosemantic feature discovery, improved architectures and objectives, automated labeling, standardized evaluation, and model-wide SAE releases (Huben et al., 2024; Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2025; Paulo et al., 2024; Karvonen et al., 2025; Makelov et al., 2025; Rajamanoharan et al., 2024a;b; Bussmann et al., 2024; Mudide et al., 2025; Bussmann et al., 2025; Lieberum et al., 2024; Deng et al., 2026). Despite these advances, feature discovery is still driven primarily by sparse reconstruction, while interpretability is assessed after the dictionary has been learned. Our work asks an inverse scientific question: whether interpretable directions share a measurable property of their activation distributions that can itself serve as the discovery signal. Our results identify non-Gaussianity of projection distributions as such a signal.

Non-Gaussian Directions in Representation Space. Independent component analysis (ICA) is a classical method for recovering statistically independent, non-Gaussian directions in linear representation spaces (Hyvarinen, 1999; Hyvarinen & Oja, 2000), and prior work has found interpretable ICA components in several representation spaces (Yamagiwa et al., 2023; Musil & Mareček, 2024a;b). Our work contributes a broader scientific framing for non-Gaussian feature discovery, a corrected evaluation protocol for ICA interpretability, a practical workflow for modern LLMs, and a systematic comparison with public SAEs. First, we shift the focus from ICA itself to the broader relationship between non-Gaussianity and interpretability. We find interpretable non-Gaussian structure not only in ICA components, but also in SAE decoder directions and concept-constructed directions, and quantify this relationship using both human-label confidence and automated interpretability. Second, we identify and correct a long-overlooked and consequential evaluation failure mode that systematically biases ICA interpretability downward. Because FastICA assigns signs arbitrarily while coherent semantics often occupy only one tail, evaluating a fixed positive tail can substantially underestimate ICA interpretability and make ICA appear weaker than it is. Our skewness-based semantic-tail selection recovers much of this gap across fitting budgets, corpora, and evaluator pipelines. Third, we provide a practical and reusable ICA workflow for LLM interpretability, combining GPU-parallel fitting, component profiling, a dedicated annotation interface, systematic human auditing, convergence and stability analysis, and tools for reading and intervening on model activations. Finally, we fill an important empirical gap by comparing ICA and public SAEs across interpretability, reconstruction, sparse probing, targeted perturbation, efficiency, geometric overlap, and tokenwise response structure.

3 Conclusion

Our results show that higher-order activation geometry can itself serve as a feature-discovery signal, opening a lightweight route to repeated analysis across model layers and training checkpoints.

Reproducibility Statement

The supplementary materials provide the code needed to reproduce ICA Lens and the reported analyses. Appendix B documents the full path from activation collection and whitening to GPU symmetric FastICA, residual-space transforms, computational measurements, and seed- and iteration-level convergence analyses. Appendix C reports all 150 manual annotations and specifies the tail-orientation, counterfactual-testing, automated-evaluation, matched-sampling, robustness, and error-analysis protocols. Appendix D gives the component-selection and intervention settings for the reading and generation case studies. Appendix E defines every dataset, layer, aggregation rule, coordinate budget, metric, and intervention used in reconstruction, sparse probing, and targeted probe perturbation, with full disaggregated results. Finally, Appendix F records the exact public SAE releases, checkpoint revisions, and feature-filtering rules used in all comparisons.

AI Use Statement

Generative AI tools were used for literature search and writing polish. LLMs also assisted with generating candidate positive and counterfactual examples during component annotation and served as explainer and predictor models in the automated interpretability evaluations described in Appendix C.3. The authors designed the experimental protocols, reviewed and refined generated materials where appropriate, and verified all reported analyses, results, and claims.

References

Pierre Ablin, Jean-François Cardoso, and Alexandre Gramfort. Faster independent component analysis by preconditioning with hessian approximations. *IEEE Transactions on Signal Processing*, 66(15):4040–4049, 2018. doi: 10.1109/TSP.2018.2844203.

Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995. doi: 10.1162/neco.1995.7.6.1129.

Camila Blank, Agam Bhatia, and Neel Nanda. R-Lens: Making J-Lens more faithful on early layers. *AI Alignment Forum*, August 2026. URL https://www.alignmentforum.org/posts/nv8oedrnLXKRzNEL9/rlens-making-j-lens-more-faithful-on-early-layers.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

Jean-François Cardoso and Antoine Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proceedings F: Radar and Signal Processing*, 140(6):362–370, 1993. doi: 10.1049/ip-f-2.1993.0054.

Boyi Deng, Xu Wang, Yaoning Wang, Yu Wan, Yubo Ma, Baosong Yang, Haoran Wei, Jialong Tang, Huan Lin, Ruize Gao, et al. Qwen-scope: Turning sparse features into development tools for large language models. *arXiv preprint arXiv:2605.11887*, 2026.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer

circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

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. In *International Conference on Learning Representations*, 2025.

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

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.

Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.

Aapo Hyvarinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, et al. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*, 2025.

Patrick Leask, Neel Nanda, and Noura Al Moubayed. Inference-time decomposition of activations (itda): A scalable approach to interpreting large language models. *arXiv preprint arXiv:2505.17769*, 2025.

Te-Won Lee, Mark Girolami, and Terrence J. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, 11(2):417–441, 1999. doi: 10.1162/089976699300016719.

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. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300, 2024.

Aleksandar Makelov, Georg Lange, and Neel Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control. In *International Conference on Learning Representations*, volume 2025, pp. 33588–33636, 2025.

Jari Miettinen, Klaus Nordhausen, and Sara Taskinen. fica: Fastica algorithms and their improved variants. *The R Journal*, 10(2):148–158, 2018. doi: 10.32614/RJ-2018-046.

Anish Mudide, Josh Engels, Eric Michaud, Max Tegmark, and Christian Schroeder de Witt. Efficient dictionary learning with switch sparse autoencoders. In *International Conference on Learning Representations*, volume 2025, pp. 101830–101844, 2025.

Tomáš Musil and David Mareček. Exploring interpretability of independent components of word embeddings with automated word intruder test. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 6922–6928, Torino, Italia, May 2024a. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.605/.

Tomáš Musil and David Mareček. Transforming hidden states into binary semantic features, 2024b.

nostalgebraist. Interpreting GPT: The logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024.

Anastasia Podosinnikova, Amelia Perry, Alexander S Wein, Francis Bach, Alexandre d'Aspremont, and David Sontag. Overcomplete independent component analysis via sdp. In *The 22nd international conference on artificial intelligence and statistics*, pp. 2583–2592. PMLR, 2019.

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

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

Przemyslaw Spurek, Pawel Rola, Jacek Tabor, and Adam Czechowski. Ica based on split generalized gaussian. *arXiv preprint arXiv:1802.05550*, 2018.

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, 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. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Hiroaki Yamagiwa, Momose Oyama, and Hidetoshi Shimodaira. Discovering universal geometry in embeddings with ica. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4647–4675, 2023.

CONTENTS

A Further Analysis and Discussion **15**

A.1 Constructing Interpretable Non-Gaussian Directions from Concepts 15

A.2 Relationship Between Non-Gaussian Directions and SAE Features 16

A.2.1 Directional Overlap and Complementarity 16

A.2.2 Tokenwise Response Structure 17

A.3 Scope, Limitations, Future Work, and Impacts 20

B ICA Lens Implementation, Efficiency, and Convergence **21**

B.1 Activation Collection and FastICA Fitting 21

B.2 Practical Efficiency 22

B.3 Fitting Dynamics 22

B.4 Seed Stability and Iteration Budgets 26

B.5 Convergence Diagnostics: Non-Gaussianity vs. LIM 27

C Labeling and Interpreting Non-Gaussian Directions **28**

C.1 Selecting the Semantic Tail by Skewness 28

C.2 Manual Audit Protocol and Full Annotations 29

C.3 Automated Interpretability Protocol 35

C.4 ICA–SAE Reproduction and Robustness Checks 35

C.5 Error Analysis of the Relationship Between Non-Gaussianity and Automated Interpretability 36

D Reading and Intervening with Non-Gaussian Directions **37**

D.1 Reading Component Mixtures 38

D.1.1 Contextual Decomposition of Polysemy 38

D.1.2 Lexical Structure in Token Embeddings 38

D.2 Illustrative Generation Interventions 40

E Full Quantitative Evaluation of Non-Gaussian Directions **41**

E.1 Evaluation Protocol and Aggregation 41

E.2 Sparse Reconstruction 42

E.2.1 Across Datasets 42

E.2.2 Across Layers 43

E.2.3 Alternative Metric 44

E.3 Sparse Probing 45

E.4 Targeted Probe Perturbation 47

F SAE Baselines and Checkpoints **49**

APPENDIX ROADMAP

The appendix extends the main results along six complementary directions, covering the scientific interpretation of non-Gaussianity, implementation and convergence, interpretability evaluation, practical use, full quantitative validation, and exact SAE baselines.

- **Appendix A: Further Analysis and Discussion.** We broaden the central finding beyond directions discovered directly by ICA and SAEs: concept-constructed directions also produce non-Gaussian tails, suggesting that non-Gaussianity is a more general signature of interpretable structure. We then compare ICA and SAE representations through directional overlap and tokenwise responses, and summarize the paper's scope, limitations, and future directions.
- **Appendix B: ICA Lens Implementation, Efficiency, and Convergence.** We provide the complete activation-collection, whitening, and symmetric FastICA procedure, together with practical memory, latency, and fitting-cost measurements. Layer-wise fitting trajectories, multi-seed experiments, and long-run convergence diagnostics reveal a useful distinction: population-level non-Gaussian structure is recovered quickly, while exact component identities continue to stabilize with longer optimization.
- **Appendix C: Labeling and Interpreting Non-Gaussian Directions.** We specify the skewness-based semantic-tail selection used throughout the paper and provide the full manual and automated interpretability protocols. This includes all 150 randomly sampled human annotations, sentence-level and counterfactual hypothesis tests, a dedicated ICA Lens annotation interface, reproduction of the historical ICA–SAE comparison, robustness checks across fitting budgets, corpora, and evaluator models, and an error analysis of automated interpretability. Together, these analyses characterize both when non-Gaussian directions admit stable interpretations and how reliably those interpretations can be measured.
- **Appendix D: Reading and Intervening with Non-Gaussian Directions.** We show how labeled components become a usable coordinate system rather than only a collection of feature descriptions. Case studies decompose contextual representations of polysemous words, expose interpretable lexical mixtures directly in token embeddings, and use selected coordinates for generation-time steering and clamping.
- **Appendix E: Full Quantitative Evaluation of Non-Gaussian Directions.** We give the complete protocols and dataset-level results for sparse reconstruction, sparse probing, and targeted probe perturbation. Reconstruction is further disaggregated across domains and model depth and evaluated with an alternative NMSE metric, while sparse probing includes additional comparisons with compact Matryoshka SAEs and ITDA, and TPP separates intended from unintended effects. These analyses test the recovered coordinates across datasets, layers, metrics, model scales, and alternative representation methods.
- **Appendix F: SAE Baselines and Checkpoints.** We record the exact public SAE releases and revisions used throughout the paper and specify the active-feature filtering used whenever individual SAE features are sampled for comparison.

A Further Analysis and Discussion

This section provides additional analyses that extend the main results. We first ask whether interpretable non-Gaussian directions can also be constructed directly from known concepts. We then examine how the directions recovered by ICA relate to SAE features, both geometrically and through their tokenwise response patterns. Finally, we discuss the scope of our current results and several directions for extending non-Gaussian feature discovery.

A.1 Constructing Interpretable Non-Gaussian Directions from Concepts

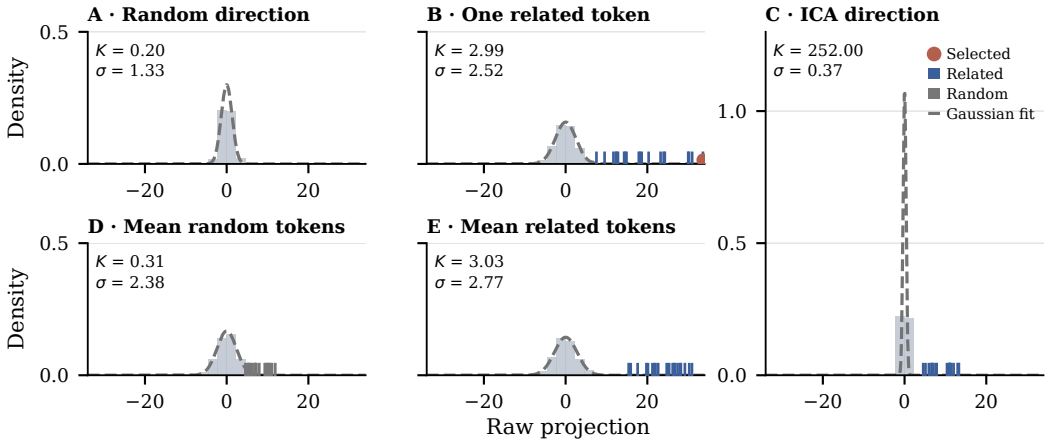


Figure 10: **Concept-constructed directions are non-Gaussian, while ICA finds a cleaner separation between the concept tail and background.** The panels show raw projections of all 50,256 token activations. K denotes excess kurtosis and σ denotes the standard deviation across all tokens. (A) A random direction produces an approximately Gaussian distribution. (B) The direction of one related token moves the concept family into the positive tail. (C) The ICA direction retains the same coherent tail while concentrating most background tokens near zero. (D) Averaging random-token directions does not produce a coherent tail. (E) Averaging the related-token directions again produces a non-Gaussian tail, but with a broader background than the ICA direction.

Our main experiments start from general feature discovery: first search for a non-Gaussian projection, then inspect its tail to determine whether it is interpretable. Here we reverse this process. Starting from a known concept, can we construct a direction from representative tokens and obtain a non-Gaussian projection distribution? This provides a simple way to test whether the connection between interpretability and non-Gaussianity extends beyond directions found directly by ICA.

Neither	␣neither	␣Neither	␣either
␣Either	␣Both	Both	Either
␣both	either	both	OTH
ither	between	␣Between	␣between
Between	ween	␣latter	␣respectively

Table 2: **The 20 related tokens used in the concept-construction experiment.** ␣ denotes a leading space. The family is defined by the largest-magnitude projections onto the selected ICA component.

We study this question using the layer 0 activations of all 50,256 non-special GPT-2 tokens. We first select an ICA component with high excess kurtosis whose largest-magnitude tokens form a coherent family involving variants of *either*, *neither*, *both*, *between*, and related tokens. We use these tokens as a controlled concept family and ask whether simple directions constructed from them also produce non-Gaussian projection distributions. Table 2 lists the 20 tokens.

We compare five unit directions in Figure 10: a random direction, the activation direction of one related token (*either*), the ICA direction, the mean direction of 20 random tokens, and the mean direction of the 20 related tokens. All projection statistics are computed over the same full vocabulary.

Simple concept-seeded directions already produce non-Gaussian structure. Using one related token as the direction moves the concept family into the positive tail and increases excess kurtosis from $K = 0.20$ for a random direction to $K = 2.99$ in Figure 10B. Averaging the 20 related-token directions gives a similar distribution with $K = 3.03$ in Figure 10E. In contrast, averaging 20 random-token directions remains close to Gaussian ($K = 0.31$). Thus, a direction constructed directly from a coherent concept can itself produce a non-Gaussian projection distribution.

The ICA direction shows a stronger version of the same geometry. The related-token family remains in the tail, while most other tokens are concentrated much closer to zero. The single-token and concept-mean directions have standard deviations of $\sigma = 2.52$ and $\sigma = 2.77$, respectively, while the ICA direction has $\sigma = 0.37$ and much larger excess kurtosis ($K = 252.00$). The fitted ICA direction therefore combines a coherent concept tail with a substantially more concentrated background. This separation is also useful in practice: when interpreting a direction from its top-activating examples, we want unrelated background tokens to remain well separated from the concept tokens in the tail.

Key Insight

Non-Gaussianity is not specific to ICA. Starting from a known concept and constructing a direction from representative tokens already produces a non-Gaussian tail. ICA provides a direct unsupervised way to search for this geometry and can produce a cleaner separation between the concept tail and unrelated background.

A.2 Relationship Between Non-Gaussian Directions and SAE Features

ICA and sparse autoencoders are fitted with very different objectives: ICA searches directly for non-Gaussian projections, while SAEs learn an overcomplete dictionary through sparse reconstruction. We therefore ask how much structure the two methods recover in common. We compare them from two complementary perspectives: geometric alignment between directions and tokenwise response patterns on actual sequences.

A.2.1 Directional Overlap and Complementarity

We first ask how closely ICA directions correspond to directions in the much larger SAE dictionaries. For each layer, we compare every ICA residual-space writing direction d_j with every decoder direction in the corresponding released SAE. Both sets are ℓ_2-normalized, and for each ICA direction we record its maximum absolute cosine similarity with any SAE decoder direction. The absolute value removes the arbitrary sign of an ICA component.

As a matched reference, we sample the same number of isotropic Gaussian directions directly in residual-stream space and compare them with the same SAE dictionary. Each layer therefore contains exactly d ICA queries and d random queries, where d is the residual dimension. The random reference uses the same residual space, SAE dictionary, and number of queries as the ICA comparison.

Figure 11 shows the two methods recover related but not identical structure. Non-Gaussian search recovers many directions also represented by SAEs, while also exposing additional directions that are geometrically distinct from their nearest SAE features. The same pattern persists across model depth. Figure 12 reports the median nearest-SAE similarity separately for every layer. ICA remains well above the matched random reference throughout all three models, showing that the overlap is not confined to a small subset of layers.

Key Insight

ICA and SAEs recover overlapping but complementary structure. Many non-Gaussian directions have close SAE counterparts, while others remain only weakly aligned with the released SAE dictionaries. This supports direct non-Gaussian search as a useful first lens for model representations: it can quickly expose substantial interpretable structure before committing to a larger learned dictionary.

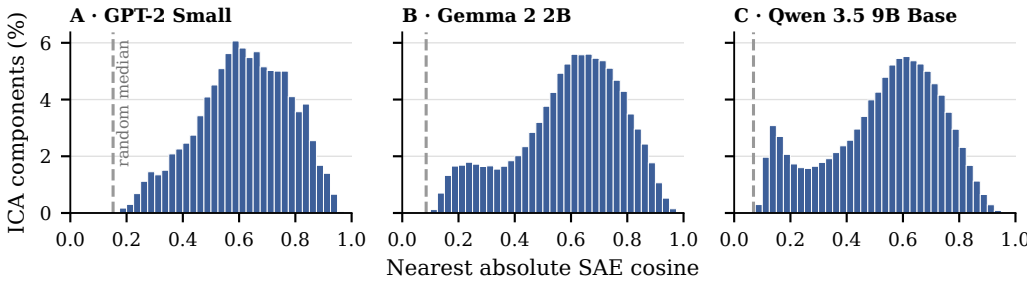


Figure 11: ICA directions show substantial but incomplete overlap with SAE features. Each panel pools all ICA components across the evaluated layers of one model and shows the maximum absolute cosine with any decoder direction from the corresponding same-layer SAE. Dashed gray lines mark the median obtained from the same number of isotropic random residual-space directions. ICA distributions are shifted well above the random references, while their broad spread also reveals directions with much weaker SAE alignment.

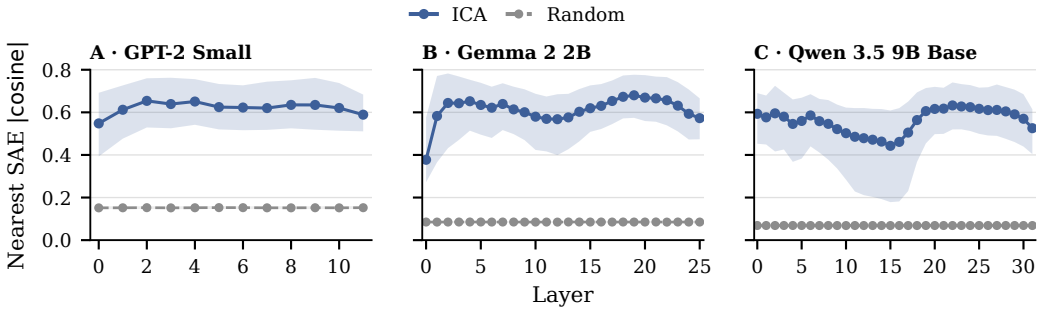


Figure 12: ICA-to-SAE directional overlap across model depth. Blue points show the median maximum absolute cosine between ICA writing directions and same-layer SAE decoder directions, with bands showing interquartile ranges. Gray points and bands report the same statistics for matched isotropic random directions in residual space. ICA remains clearly above the random reference throughout model depth, while the strength of alignment varies across layers.

A.2.2 Tokenwise Response Structure

Directional cosine measures geometric similarity, but it does not show how ICA components and SAE features respond across an actual sequence. We therefore compare their tokenwise response patterns on the same sentence:

> Maya stopped at the bank before the trip, waiting in line to deposit a check and withdraw enough cash for the weekend.

For each token position, we identify the highest-response ICA component and the highest-response SAE feature. ICA selection uses the largest absolute component score, while SAE selection uses the largest native feature activation. We keep each selected component or feature once, even if it is the strongest response at multiple token positions, and plot its response across the full sentence. For visualization, ICA scores are shown in absolute value, SAE features use their native post-activations, and each row is normalized by its own maximum.

Figures 13, 14, and 15 show a similar range of tokenwise response patterns across models and layers. Both ICA and SAE representations contain sharply localized responses concentrated around individual tokens, as well as broader responses that extend across several token positions. The relative mixture varies across models, layers, and selected components or features, but both types of structure appear consistently in the two representations.

Key Insight

ICA and SAE representations show a similar range of tokenwise response patterns across models and layers. Both contain sharply localized responses as well as components or features whose activation extends across multiple tokens.

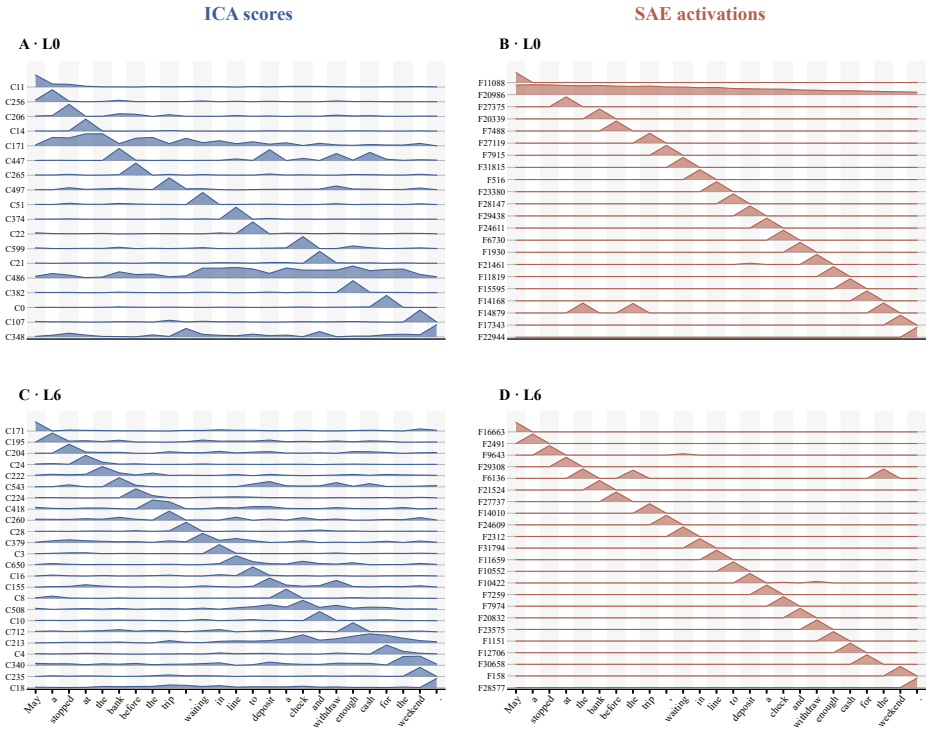


Figure 13: Tokenwise response patterns for GPT-2 Small at layers 0 and 6. Within each method, we select the union of components or features that attain the strongest response at any token position. ICA selection uses absolute component score, while SAE selection uses native feature activation. Each row shows the selected component or feature's response over the full sentence and is normalized by its own maximum.

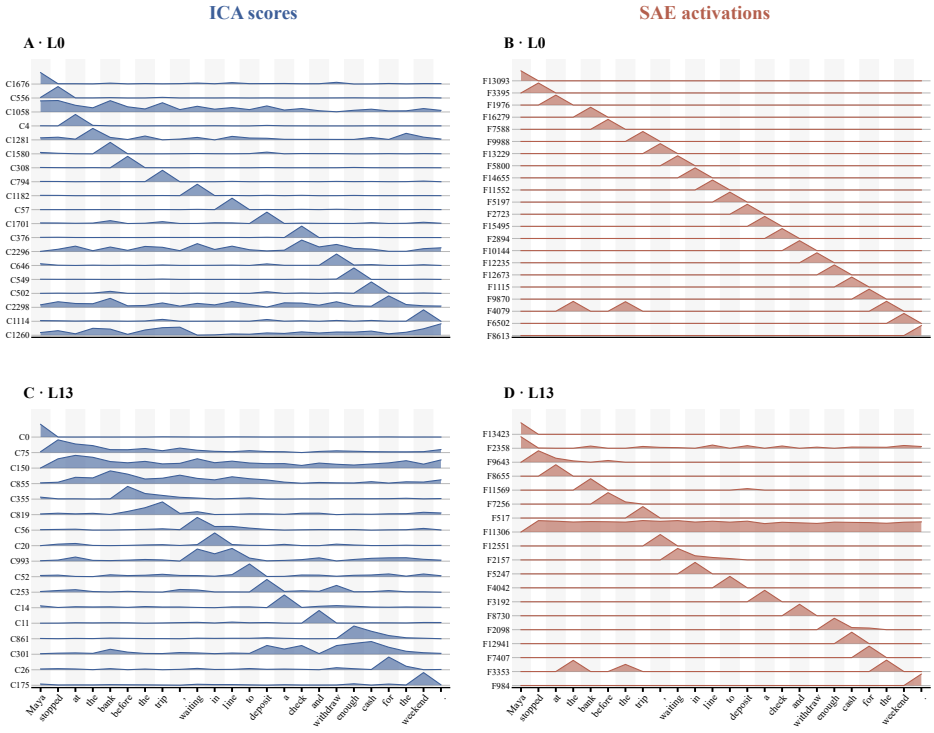


Figure 14: Tokenwise response patterns for Gemma 2 2B at layers 0 and 13.

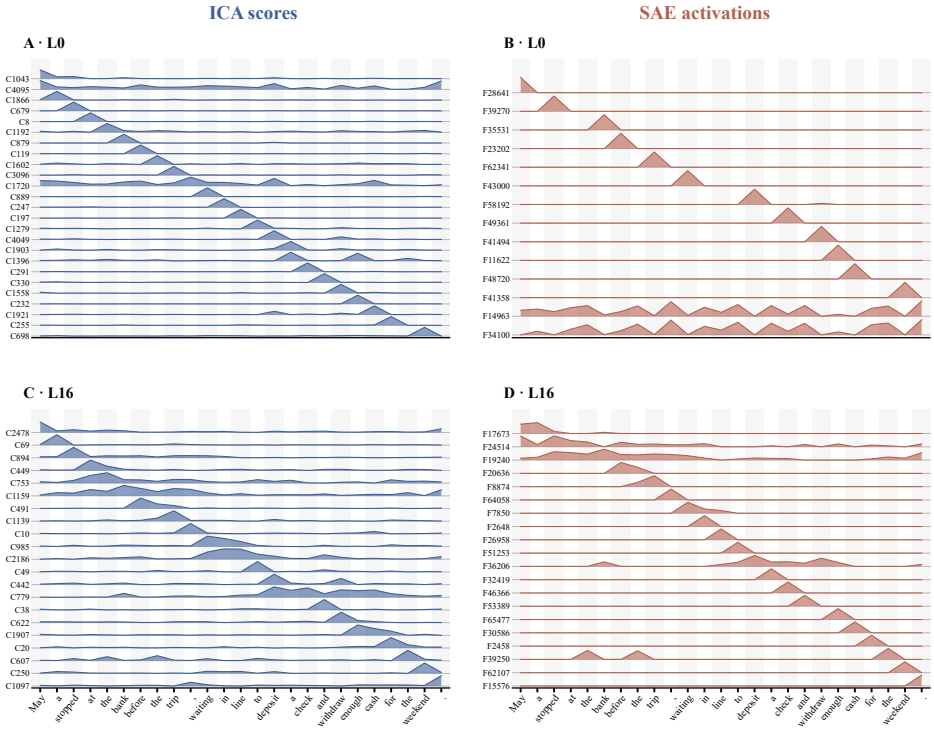


Figure 15: Tokenwise response patterns for Qwen 3.5 9B Base at layers 0 and 16.

A.3 Scope, Limitations, Future Work, and Impacts

Understanding and Learning Interpretable Non-Gaussian Geometry. Our paper provides an empirical explanation for why non-Gaussianity aligns with interpretability. From the perspective of human annotation, we often try to explain a direction using its top-k activating tokens. A nearly Gaussian direction is difficult to interpret from only its top-k tokens, while a non-Gaussian direction has at least the potential to isolate some special tokens in its tail. In LLM residual-stream geometry, we find that the tails of many non-Gaussian directions correspond to interpretable concepts. This suggests a possible geometric picture in which tokens associated with the same concept form a cluster that is separated from unrelated tokens. An important question for future work is why language-model pretraining learns this structure, how it helps reduce the training loss, and whether we can use this structure during pretraining to make models more interpretable by design.

Compact FastICA Basis as a First Lens. We use FastICA as a practical way to test our hypothesis that non-Gaussianity can serve as a feature-discovery signal. However, standard FastICA is compact: in a d-dimensional fitting space, it returns at most d components. This makes the resulting basis efficient to fit and inspect, but it does not provide the large overcomplete feature vocabulary that modern SAEs are designed to learn. We therefore position ICA as a compact first lens for language-model activations, rather than as a replacement for sparse autoencoders. ICA Lens asks how much interpretable structure is already visible from activation geometry, helping analysts decide where heavier dictionary learning is worth the cost.

A natural extension is to explore overcomplete ICA and other ICA formulations that change the capacity, optimization procedure, or non-Gaussianity objective. Examples include overcomplete ICA, preconditioned or adaptive FastICA, deflationary ICA, JADE, Infomax, extended Infomax, and objectives designed for heavy-tailed distributions (Podosinnikova et al., 2019; Ablin et al., 2018; Miettinen et al., 2018; Cardoso & Souloumiac, 1993; Bell & Sejnowski, 1995; Lee et al., 1999; Spurek et al., 2018). This perspective may also inspire new ways to design and improve SAE objectives.

Beyond single-layer residual streams. Our experiments focus on token-level residual-stream activations analyzed one layer at a time, but ICA Lens is lightweight enough to support broader and more repeated analysis. At a finer granularity, it could be applied to attention-head outputs, MLP outputs, or other submodules; across depth, normalized activations from multiple layers could be analyzed jointly to identify directions that emerge, persist, or transform. A particularly promising use case is longitudinal feature discovery across training checkpoints. Because ICA Lens can be refit quickly, the same analysis could be repeated throughout pretraining or post-training to track when a direction appears, strengthens, changes meaning, or disappears, without training a separate overcomplete dictionary at every checkpoint. This could turn non-Gaussian feature discovery from a one-shot analysis into a practical tool for studying how internal structure emerges and changes during training.

The same idea extends beyond individual token states. Pooling or segmenting activations could define representations at the sentence, reasoning-step, or trajectory level, allowing components to be tracked over a reasoning process. Comparisons across base and instruction-tuned models, reinforcement-learning stages, or successive checkpoints could reveal which latent directions are preserved, created, or suppressed by training. At the agent level, analogous analyses over observation–reasoning–action trajectories could search for directions associated with planning, uncertainty, recovery, or failure. Longer term, joint analysis of language, vision, and action representations may reveal latent factors shared across modalities.

B ICA LENS IMPLEMENTATION, EFFICIENCY, AND CONVERGENCE

This section describes the implementation of ICA Lens and studies its practical behavior at LLM scale. We first specify activation collection, whitening, and symmetric FastICA fitting, including the reading and writing directions used throughout the paper. We then measure the memory, latency, and end-to-end fitting cost of the resulting coordinate system. Finally, we examine fitting dynamics and stability across iterations and random initializations. Our results reveal an important distinction between population-level and component-level convergence. Population non-Gaussianity rises quickly and largely saturates early in fitting, while individual component identities continue to stabilize over longer runs.

B.1 ACTIVATION COLLECTION AND FASTICA FITTING

Activation collection. We collect fitting activations from the training split of `NeelNanda/pile-10k`. For each of its 10,000 documents, we tokenize the text with the corresponding model tokenizer and retain up to the first 1,024 tokens. Each document is processed as a full sequence, and we cache the residual-stream activation at every retained token position, so each activation contains the preceding document context available at that position. Unless stated otherwise, we uniformly sample one million token activations for each layer fit.

We fit a separate ICA basis to every evaluated residual-stream layer in GPT-2 Small, Gemma 2 2B, and Qwen 3.5 9B Base. For the projection-distribution analyses in the main text, we additionally sample a separate set of one million held-out activations per layer. These held-out activations are not used to fit ICA and are used to measure quantities such as excess kurtosis after fitting.

Centering and whitening. Let $X \in \mathbb{R}^{n \times d}$ denote the sampled activation matrix for one layer, with one residual-stream activation per row. We first subtract the empirical mean,

$$\mu = \frac{1}{n} X^\top \mathbf{1}, \qquad X_c = X - \mathbf{1}\mu^\top. \tag{2}$$

We then compute the covariance eigendecomposition

$$\frac{1}{n} X_c^\top X_c = U \Lambda U^\top \tag{3}$$

and whiten the centered activations as

$$Z = X_c U \Lambda^{-1/2}. \tag{4}$$

This gives $\frac{1}{n} Z^\top Z = I_d$, removing the empirical mean and second-order correlations while preserving the full residual dimension.

Symmetric FastICA. After whitening, the remaining linear freedom is an orthogonal rotation. We recover this rotation with symmetric FastICA using the logcosh contrast (Hyvarinen, 1999; Hyvarinen & Oja, 2000). Let $W \in \mathbb{R}^{d \times d}$ contain the current ICA axes as unit-norm rows and let

$$Y = ZW^\top \tag{5}$$

denote the corresponding component scores. Symmetric FastICA updates all d components in parallel using a fixed-point step derived from the chosen non-Gaussianity contrast. For the logcosh contrast, the update uses $g(y) = \tanh(y)$ and its derivative $g'(y) = 1 - \tanh^2(y)$. In matrix form, the parallel update can be written as

$$\widetilde{W} = \frac{1}{n} g(Y)^\top Z - \operatorname{diag}\left(\frac{1}{n} \sum_{i=1}^{n} g'(Y_{i,:}) \right) W. \tag{6}$$

The updated axes are then symmetrically decorrelated,

$$W \leftarrow \left(\widetilde{W} \widetilde{W}^\top \right)^{-1/2} \widetilde{W}, \tag{7}$$

which keeps the recovered axes mutually orthogonal in whitened space while allowing all components to be optimized jointly. During this decorrelation step, eigenvalues of the component Gram matrix are floored at machine epsilon before taking the inverse square root for numerical stability.

All main experiments use symmetric FastICA for 50 iterations. Using the same fixed iteration budget across models and layers keeps the feature-discovery procedure consistent throughout the main evaluation. Longer runs of up to 200 or 500 iterations are used only in the convergence and stability analyses in Appendices B.4 and B.5.

Reading and writing directions. FastICA returns axes in whitened coordinates, but model inspection and intervention take place in the original residual-stream space. For an ICA axis w_j, the score of a centered residual activation x_c can be written as

$$y_j = x_c^\top r_j, \qquad r_j = U\Lambda^{-1/2} w_j, \tag{8}$$

where r_j is the residual-space *reading direction*. Conversely, changing component coordinate j by one unit maps back to residual space along the *writing direction*

$$d_j = U\Lambda^{1/2} w_j. \tag{9}$$

The two sets form a dual pair,

$$r_i^\top d_j = \delta_{ij}, \tag{10}$$

although neither set is generally orthogonal in the original residual-stream coordinates. We retain μ, U, Λ, and the fitted ICA rotation so that component scoring, reconstruction, vocabulary readout, and intervention all use the same fitted coordinate system.

B.2 Practical Efficiency

ICA Lens produces a complete coordinate system with modest storage and application cost. Table 3 compares the fitted ICA representation with the matched public SAE at one middle layer per model. Memory is the persistent float32 parameter and buffer storage required for one fitted layer. Latencies exclude language-model inference and report the median batch-amortized time per activation over 100 synchronized CUDA trials after 20 warm-up passes, using batches of 256 float32 residual vectors.

The ICA fitting time is the end-to-end wall-clock time required to fit one layer from one million sampled activations for 50 FastICA iterations. It includes covariance estimation, eigendecomposition, whitening, FastICA optimization, and construction of the final fitted transforms rather than only the FastICA iteration loop. We report the range over two complete fits on a single NVIDIA GB10. For the public SAE baselines, checkpoint-specific training times are not available, so the table compares their stored representation and application costs rather than training cost.

Table 3: **Practical efficiency of ICA and the matched public SAE baselines.** Memory is persistent float32 parameter and buffer storage for one layer. Encode–decode latency includes mapping the representation back to residual space. Parentheses report the SAE-to-ICA ratio. ICA fitting time is end-to-end wall-clock time for a complete one-million-activation, 50-iteration fit.

Model	Layer	Memory / layer (MiB) ICA / SAE	Encode (μs) ICA / SAE	Encode–decode (μs) ICA / SAE	ICA fit (min)
GPT-2 Small	6	4.50 / 192.25 (42.7×)	0.206 / 11.616 (56.4×)	0.406 / 15.418 (38.0×)	0.92
Gemma 2 2B	13	40.51 / 288.20 (7.1×)	0.998 / 6.748 (6.8×)	2.055 / 11.903 (5.8×)	5.51–5.57
Qwen 3.5 9B Base	16	128.02 / 2048.52 (16.0×)	2.962 / 45.634 (15.4×)	5.839 / 75.975 (13.0×)	12.19–13.13

Across the three models, the fitted ICA representation uses 7.1–42.7× less persistent memory than the matched public SAEs. Encoding is 6.8–56.4× faster, and a full encode–decode pass is 5.8–38.0× faster. End-to-end fitting takes less than one minute per layer for GPT-2 Small, about 5.5 minutes for Gemma 2 2B, and 12–13 minutes for Qwen 3.5 9B Base. The resulting coordinate system can therefore be fitted across many layers without a separate representation-learning training pipeline. ICA and the public SAEs have different capacities, since ICA contains exactly d components while the SAEs are overcomplete; Appendix A.3 discusses how this difference affects their practical roles.

B.3 Fitting Dynamics

Population non-Gaussianity is recovered quickly. Figure 16 tracks the FastICA logcosh contrast throughout the 50-iteration fitting budget used in the main experiments. Across GPT-2 Small,

Gemma 2 2B, and Qwen 3.5 9B Base, the contrast increases most rapidly during the early iterations and changes much more slowly later in the run. The same pattern appears across substantially different residual dimensions, suggesting that the dominant non-Gaussian structure is found early in optimization.

The component population remains heterogeneous throughout fitting. Some axes become strongly non-Gaussian, while others stay much closer to the Gaussian reference. This broad spread is expected for a complete basis: FastICA must return d directions, even though the strength of higher-order structure is not uniform across the residual space.

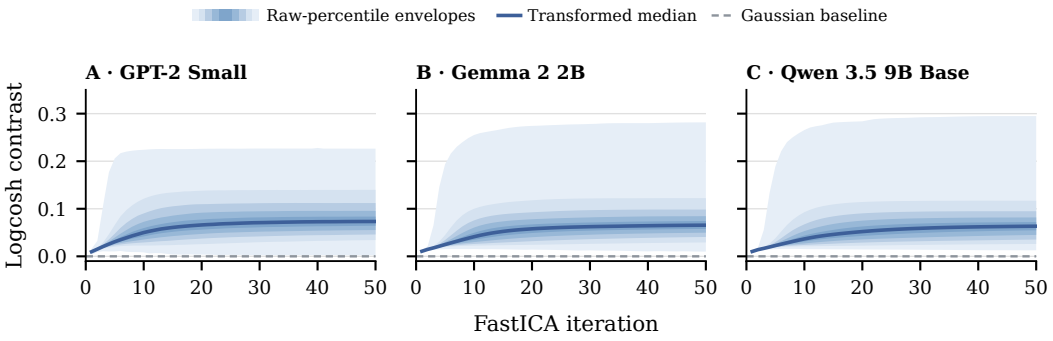


Figure 16: **FastICA rapidly recovers non-Gaussian structure across model scales.** Logcosh contrast is $C_{\text{logcosh}}(y) = |\mathbb{E}[\log\cosh(y)] - \mathbb{E}[\log\cosh(\xi)]|$ for a standardized projection y and $\xi \sim \mathcal{N}(0,1)$. The solid line is obtained by transforming the median raw logcosh statistic. From lightest to darkest, the nested bands transform the raw p0–p100, p10–p90, p20–p80, p30–p70, and p40–p60 intervals in the same way. Curves are averaged equally across fitted layers of each model, and the dashed line marks the Gaussian reference.

The same fitting pattern appears across model depth. Figures 17–19 show the corresponding trajectories separately for every layer. The rapid early increase is visible throughout model depth rather than being driven by a small subset of layers. Final contrast and component spread vary with depth, but the broad fitting dynamics remain consistent across layers and models.

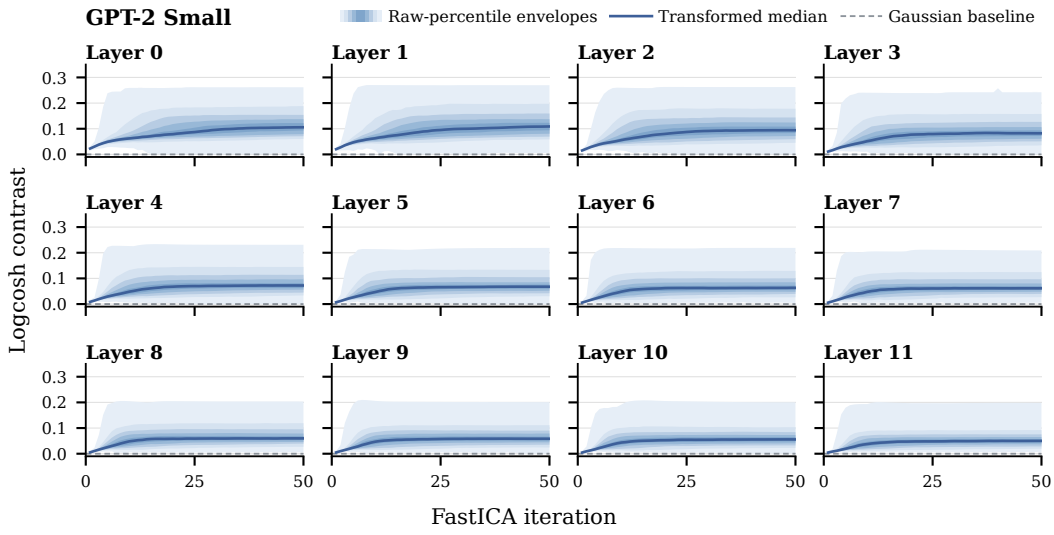


Figure 17: **Layer-wise FastICA fitting curves for GPT-2 Small.** Within each layer, the solid line and transformed percentile envelopes follow Figure 16. A shared vertical scale makes differences in final contrast and component spread visible across depth.

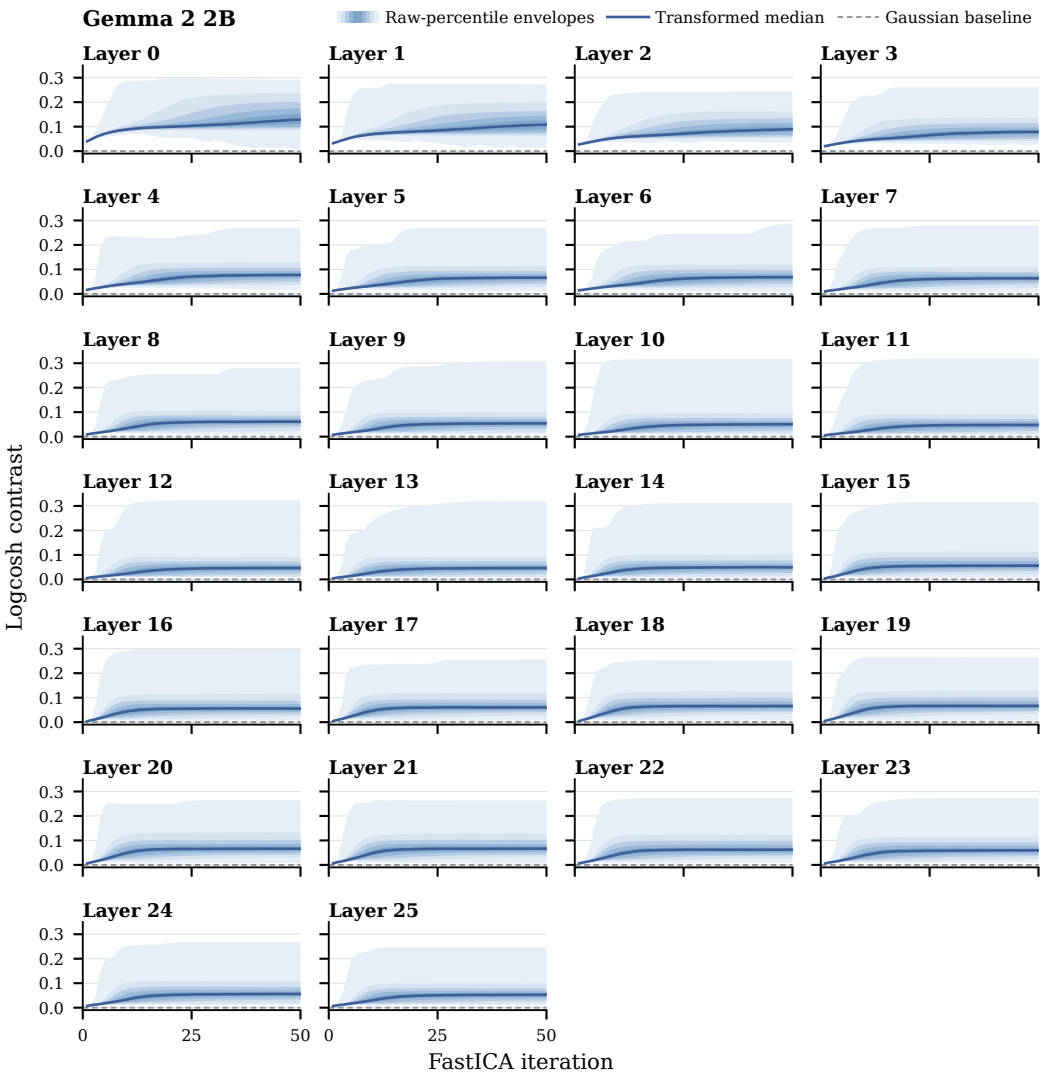


Figure 18: **Layer-wise FastICA fitting curves for Gemma 2 2B.** We use the same quantities, percentile envelopes, and axis limits as in Figure 16.

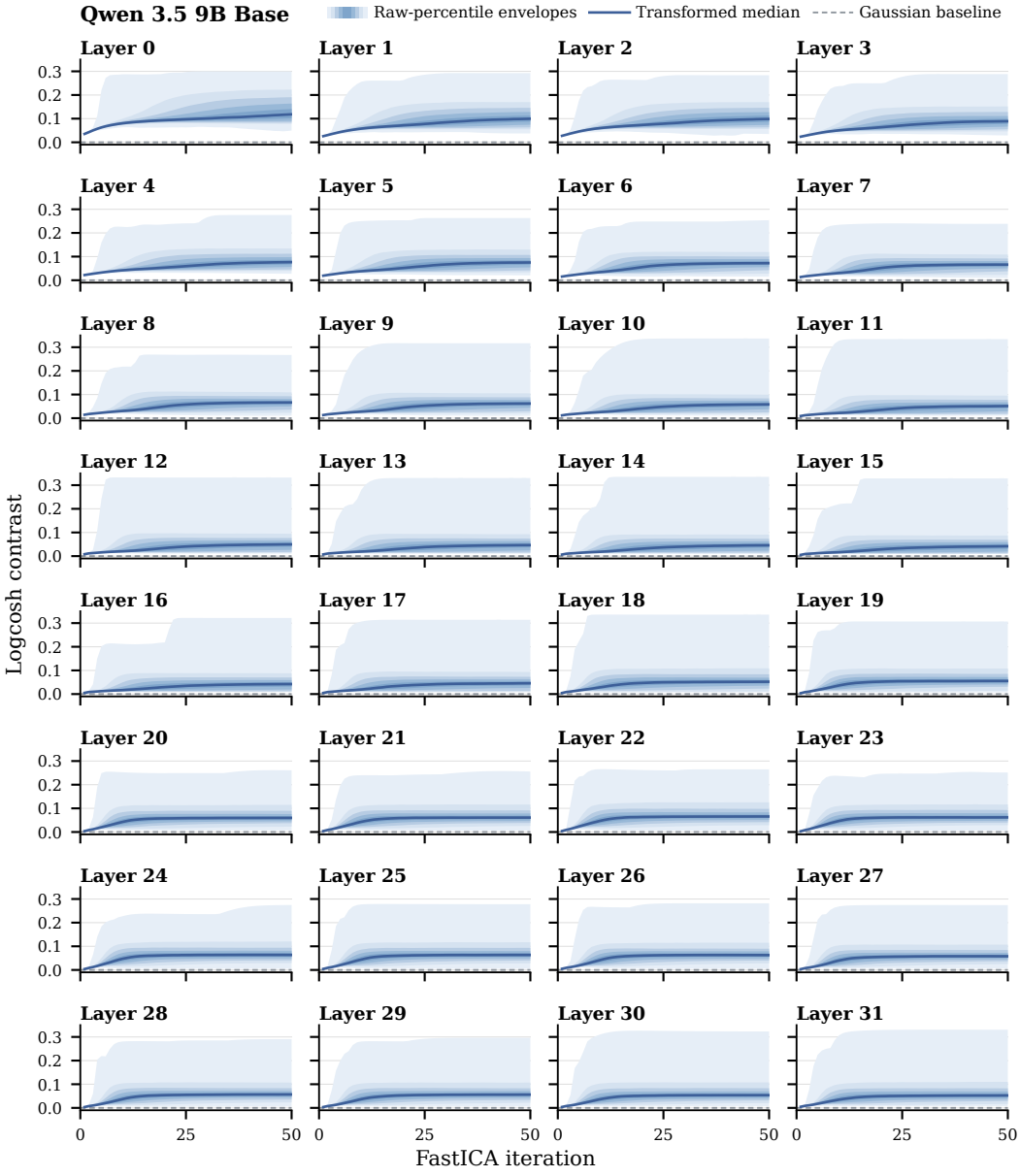


Figure 19: **Layer-wise FastICA fitting curves for Qwen 3.5 9B Base.** We use the same quantities, percentile envelopes, and axis limits as in Figure 16.

B.4 Seed Stability and Iteration Budgets

The fitting curves above describe the component population, but they do not show whether individual directions are reproducible across random initializations. We therefore measure component-level stability directly on GPT-2 Small layer 6 using five independent FastICA initializations and iteration budgets of 20, 50, 200, and 500. The one million fitting activations and all other fitting settings are held fixed across runs.

Before matching, we explicitly normalize every residual-space reading direction to unit ℓ_2 norm. For each pair of seeds, we compute the absolute cosine-similarity matrix between the two fitted bases and use Hungarian matching to obtain the one-to-one assignment that maximizes total similarity across all 768 directions.

Figure 20 shows that cross-seed agreement improves substantially with fitting time. At 20 iterations, the median matched cosine between different seeds is approximately 0.63–0.66. It rises to approximately 0.82–0.85 at 50 iterations and 0.90–0.91 at 200 iterations. Increasing the budget from 200 to 500 iterations produces only a small additional gain, with medians around 0.91–0.92. Most of the improvement in component identity therefore occurs before 200 iterations.

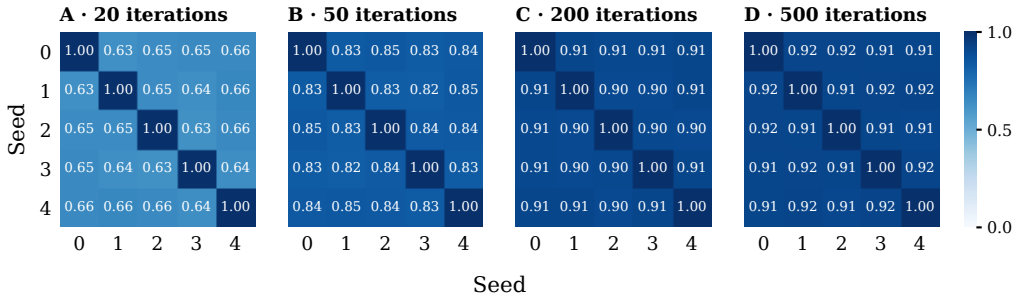


Figure 20: **Individual components become more reproducible with longer fitting.** Cross-seed stability for GPT-2 Small layer 6 at four FastICA iteration budgets. Each cell reports the median absolute cosine similarity across 768 unit-normalized reading directions after one-to-one Hungarian matching between two seeds. All fits use the same one million activation samples. Median agreement rises substantially from 20 to 50 and 200 iterations, with only a small additional increase from 200 to 500. Diagonal entries denote self-comparisons.

Figure 21A shows the full matched-similarity distributions underlying these medians. Longer runs increasingly concentrate components near high cosine similarity, while a smaller low-similarity tail remains. Figure 21B gives a complementary within-seed view by matching directions from successive iteration budgets. Agreement is highest between 200 and 500 iterations, showing that the recovered basis changes progressively less late in fitting.

The 50-iteration configuration used in the main experiments therefore recovers a largely shared basis across seeds, while longer fitting further improves the reproducibility of exact component identities. The next section shows that population-level non-Gaussianity stabilizes earlier than this component-level agreement.

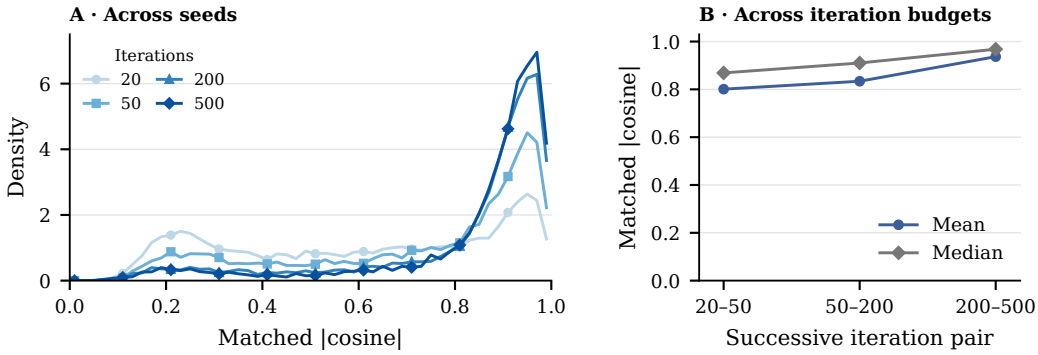


Figure 21: **Most component-level stabilization occurs by 200 iterations.** All comparisons use 768 unit-normalized GPT-2 Small layer 6 reading directions and one-to-one Hungarian matching. (A) Cross-seed distributions of matched absolute cosine similarity at each iteration budget, averaged equally across the ten distinct seed pairs. Curves connect the centers of 50 equal-width bins on $[0, 1]$. (B) Within-seed agreement between successive iteration budgets. Mean and median matched similarities are computed separately for each seed and then averaged across five seeds. Later iteration budgets produce progressively smaller changes to the recovered basis.

B.5 Convergence Diagnostics: Non-Gaussianity vs. LIM

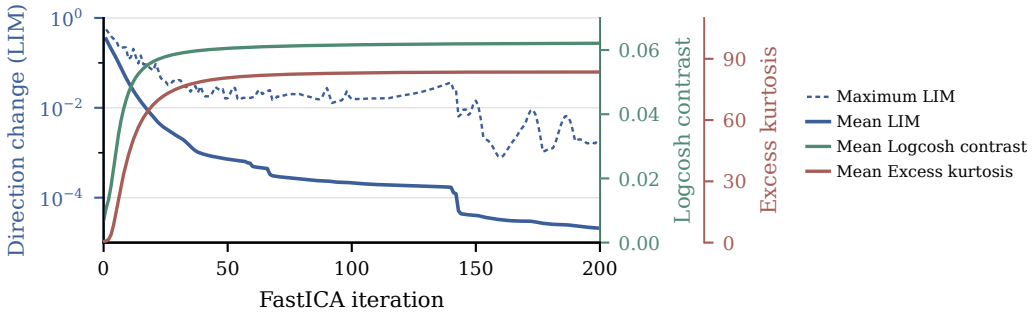


Figure 22: **Population non-Gaussianity stabilizes before individual directions.** We track all 4,096 components from each of Qwen 3.5 9B Base layers 7, 15, 23, and 31 over a 200-iteration FastICA fit, for 16,384 directions in total. Diagnostics are recomputed on the fitting activations at each checkpoint. Mean and maximum LIM measure changes between successive direction estimates, while mean logcosh contrast and mean excess kurtosis measure population non-Gaussianity. The non-Gaussianity statistics largely plateau by roughly 50 iterations, while directional changes continue to decrease over substantially longer fitting.

The results above point to two complementary notions of convergence. *Statistical convergence* asks whether the recovered component population has already captured most of the non-Gaussian structure available to FastICA. *Component convergence* asks whether individual directions have stabilized to nearly the same fixed points under additional optimization or different random initializations.

FastICA commonly tracks component convergence using the per-component limit statistic

$$\mathrm{LIM}_j^{(t)} = \left| \left| (w_j^{(t)})^\top w_j^{(t-1)} \right| - 1 \right|, \tag{11}$$

which measures how much a direction changes between successive updates, up to the arbitrary sign of an ICA component. Small LIM therefore indicates that an individual direction is approaching a fixed point. Maximum LIM is a particularly strict summary when thousands of components are optimized jointly because it is determined by the least stable direction.

Figure 22 compares directional change with population non-Gaussianity over a 200-iteration run. We track all 4,096 components from each of Qwen 3.5 9B Base layers 7, 15, 23, and 31, for 16,384 directions in total. At every checkpoint, LIM, logcosh contrast, and excess kurtosis are recomputed on the fitting activations.

Mean LIM decreases steadily throughout fitting, and maximum LIM remains noisier because it is driven by a small number of slowly stabilizing components. In contrast, mean logcosh contrast and mean excess kurtosis increase rapidly and largely plateau by roughly 50 iterations. FastICA directly optimizes logcosh rather than kurtosis, yet the two statistics follow closely aligned trajectories.

Together with the seed-stability analysis, this result explains the 50-iteration operating point used throughout the main experiments. By roughly 50 iterations, the population-level non-Gaussianity recovered by FastICA has largely saturated, while the full fitting procedure remains fast enough to apply across every layer of the evaluated models. Individual directions continue to become more reproducible beyond this point: on GPT-2 Small layer 6, the cross-seed median matched cosine rises from roughly 0.83 at 50 iterations to roughly 0.91 at 200 iterations. Longer runs therefore primarily improve the stability of exact component identities after most of the population-level statistical structure has already been recovered.

This distinction is also useful for interpreting the non-Gaussianity metric used in the main text. FastICA optimizes the logcosh contrast, while we report excess kurtosis because it has a simple Gaussian reference at zero and a wider numerical range that makes differences across directions, layers, and methods easier to visualize. Their closely aligned fitting trajectories show that both statistics track the same broad emergence and saturation of non-Gaussian structure in these fits.

C Labeling and Interpreting Non-Gaussian Directions

A useful feature representation should support interpretations that remain meaningful beyond a few selected examples. We therefore treat component labeling as an iterative hypothesis-testing process rather than a one-shot description of the largest activations. For each component, we combine evidence from both signed tails, high-scoring corpus contexts, Logit Lens and R-Lens vocabulary readouts, and controlled positive and negative examples. We use the same component profiles for human annotation and for scalable automated interpretability evaluation. This section describes the orientation rule, manual and automated evaluation protocols, robustness checks, and an error analysis of the automated metric.

C.1 Selecting the Semantic Tail by Skewness

FastICA identifies each component only up to sign: if w_j is a valid component, then $-w_j$ represents the same ICA solution. This matters for interpretation because a coherent semantic pattern is often concentrated in one tail, while the opposite tail is comparatively diffuse. Inspecting only the arbitrary positive side can therefore miss the more informative tail.

We select the semantic tail using the skewness of the component scores. Let y_{ij} denote the score of activation i on component j, consistent with the notation in Section 2.1. We compute

$$\gamma_j = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_{ij} - \bar{y}_j)^3}{\left[\frac{1}{n}\sum_{i=1}^{n}(y_{ij} - \bar{y}_j)^2\right]^{3/2}}, \qquad \bar{y}_j = \frac{1}{n}\sum_{i=1}^{n} y_{ij}. \tag{12}$$

In our main experiments, this statistic is computed on the one-million-token activation capture used to fit and profile each component. Positive skew selects the positive tail and negative skew selects the negative tail. If $\gamma_j = 0$, we break the tie using the fraction of squared score mass on the positive side,

$$q_j^+ = \frac{\sum_{i=1}^{n} y_{ij}^2 \mathbf{1}[y_{ij} > 0]}{\sum_{i=1}^{n} y_{ij}^2}. \tag{13}$$

We record the selected tail with a sign $s_j \in \{-1, +1\}$,

$$s_j = \begin{cases} +1, & \gamma_j > 0, \\ -1, & \gamma_j < 0, \\ +1, & \gamma_j = 0 \text{ and } q_j^+ \geq 0.5, \\ -1, & \text{otherwise}, \end{cases} \tag{14}$$

and define the semantic-tail score

$$\widetilde{y}_{ij} = s_j y_{ij}. \tag{15}$$

The fitted ICA basis itself is unchanged. We store s_j and apply it to component scores and the corresponding reading and writing directions when needed. Thus, the selected semantic tail is always the positive side of $\widetilde{y}_{ij}$.

For procedures that require a one-sided feature, such as automated interpretability and targeted probe perturbation, we use

$$y_{ij}^+ = \max(0, \widetilde{y}_{ij}). \tag{16}$$

Sparse probing instead uses the signed semantic-tail score $\widetilde{y}_{ij}$. Example retrieval and manual labeling use high-scoring examples from the selected semantic tail. Sign-invariant quantities such as excess kurtosis and reconstruction are unaffected by this choice.

C.2 Manual Audit Protocol and Full Annotations

We first use manual annotation to examine the structure of individual components in detail and to test whether their interpretations remain stable beyond the strongest corpus examples. We uniformly sample 50 components from the 50-iteration fits without replacement over all available layer–component pairs from each model, giving 150 randomly selected components in total. The sample therefore covers components across model depth rather than selecting only visually prominent examples.

For each component, we first inspect both signed tails, the highest-scoring corpus occurrences, and the corresponding Logit Lens and R-Lens vocabulary readouts. We use this evidence to form an initial concise hypothesis about the component. When the strongest activations suggest a broader phrase- or context-level pattern, we also trace the component across complete sentences to determine whether it responds locally to individual tokens or tracks a signal over a longer span.

We then test the hypothesis with controlled examples. LLMs assist in generating positive examples that should express the proposed concept and negative or counterfactual examples that preserve nearby surface structure while removing it. We evaluate the component on these examples, inspect mismatches, refine the hypothesis when needed, and repeat the process until the interpretation is stable or the evidence remains ambiguous. This iterative procedure helps separate broad semantic or contextual patterns from narrower lexical, positional, or syntactic explanations that can fit the original top-activating examples.

Each component receives a confidence score from 0 to 5 that summarizes the full annotation process, including how easily a coherent hypothesis emerges, how much refinement is required during contextual and counterfactual testing, and how consistently the final label explains the observed behavior. We use the following compact rubric: 0 unknown, 1 weak, 2 partial, 3 plausible, 4 stable, and 5 very clear. Scores 4–5 are treated as high-confidence labels.

Tables 4–6 report all 150 sampled components. Figure 23 summarizes the resulting confidence distributions. High-confidence labels are assigned to 82% of sampled GPT-2 Small components, 84% of Gemma 2 2B components, and 80% of Qwen 3.5 9B Base components. Across the three models, the annotations include lexical, syntactic, semantic, positional, code-related, and longer-context patterns.

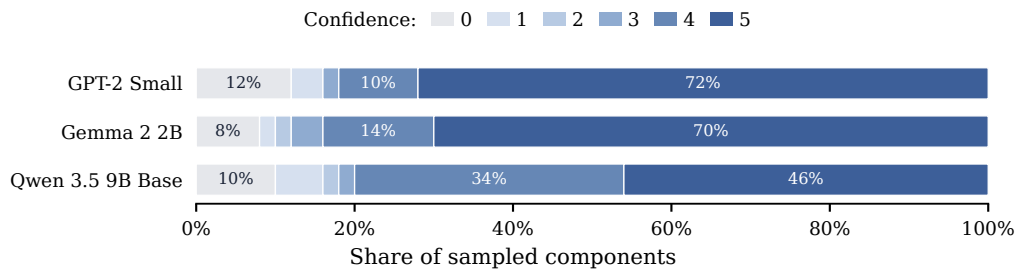


Figure 23: **Randomly sampled ICA components are frequently assigned high-confidence human interpretations.** We sample 50 components per model uniformly without replacement over all available layer–component pairs. Bars show the fraction assigned confidence 0–5 under the annotation protocol above. Scores 4–5 account for 82% (41/50), 84% (42/50), and 80% (40/50) of GPT-2 Small, Gemma 2 2B, and Qwen 3.5 9B Base components, respectively.

Table 4: Manual component annotations for GPT-2 Small.

Layer	Component	Manual label	Skewness	Excess kurtosis	Confidence
0	C421	Key	4.00	77.50	5
0	C459	Full	3.55	70.78	5
0	C524	Personal	1.35	16.82	5
0	C642	Light	-0.07	11.17	5
1	C191	5	5.28	53.48	5
1	C284	Accented Latin characters	-5.95	93.59	4
2	C46	First token after ¡—endoftext—¿	11.69	199.3	5
3	C61	’	-7.94	106.1	5
3	C393	Contact relative clause / omitted “that”	3.12	30.44	5
3	C409	Christ	-2.37	30.23	5
3	C455	Item	-1.74	15.11	5
3	C650	Include	0.69	8.41	5
3	C728	Unknown	0.68	2.69	0
4	C32	Be	8.19	100.9	5
4	C103	Enumerations/lists	-5.53	58.31	5
5	C66	He	6.33	80.76	5
5	C312	Unknown	2.25	12.87	0
5	C333	Studies	-2.83	32.28	5
5	C482	Post-{ newline	-1.69	12.21	5
5	C598	Pre	-1.02	10.85	5
5	C751	Unknown	-0.01	0.91	0
6	C159	://	-7.00	156.4	5
6	C174	Unknown	2.83	16.06	0
7	C215	Combining, Interaction	2.86	22.25	5
7	C222	Been	3.26	34.15	5
7	C347	Name	2.42	26.41	5
7	C380	Post-colon newline	-2.56	25.65	5
7	C395	Sentence-ending period	-1.90	13.48	5
7	C422	Repeated variable in later polynomial term	-1.92	21.70	5
7	C452	Unknown	-1.44	7.63	0
7	C467	Repeated math questions	1.42	10.62	1
7	C625	//	-1.65	23.46	5
7	C628	Prefix non-	1.42	17.89	5
7	C757	Post-URL for details or name a few	0.07	0.75	1
8	C152	Post-quote reporting verb	-4.34	52.80	5
8	C396	Programming/API specification language	-1.84	7.33	5
8	C743	Mr	0.24	2.92	5
9	C56	Do/does	-7.09	122.1	5
9	C119	Japanese language	4.38	49.88	4
9	C464	Group/list	-1.56	11.03	5
9	C619	Way/approach descriptor	1.00	7.15	4
9	C730	Who	0.25	2.21	5
10	C294	Suffix comp-	-2.21	17.61	5
10	C307	Opening parentheses in code	2.43	20.42	5
10	C445	Expecting a temporal-duration phrase	-1.67	13.63	4
10	C446	Opening quotation mark	1.79	15.26	5
10	C493	Occupational title or official role	-1.42	10.41	5
11	C243	Expecting letter D but need more conditions	2.12	18.73	3
11	C593	Official agency or department	0.83	5.68	4
11	C752	Unknown	-0.02	0.62	0

Table 5: Manual component annotations for Gemma 2 2B.

Layer	Component	Manual label	Skewness	Excess kurtosis	Confidence
0	C603	Pressure	24.53	1317	5
0	C761	3rd-person singular present verb	-6.38	68.75	5
0	C1423	Australian	-3.14	109.3	5
1	C1290	Words related to diagnosis	5.88	220.2	5
1	C2180	Super	-0.20	14.64	5
2	C1994	Long underscore separator	-0.05	8.70	3
3	C2037	Substring tab or cat	-0.06	5.18	5
4	C271	Why	9.26	251.6	5
4	C494	Programming errors / exceptions	5.50	100.7	5
4	C1229	Letter Q	-2.36	29.08	5
4	C1372	Check	-3.42	74.02	5
4	C2205	Fast	-0.08	3.62	5
5	C1839	U.S. Supreme Court case citation	3.46	181.3	4
6	C1503	Definitional statement	-2.15	22.26	5
6	C2082	Unknown	0.06	2.45	0
7	C39	Post-Soviet context	6.79	84.98	5
7	C878	Code syntax class=”	-5.40	143.5	5
7	C1016	Because of, a cause/reason expression	2.61	28.14	5
8	C138	Previously introduced unfamiliar proper noun	-3.96	29.91	4
10	C834	Website/E-mail domain name	-2.38	27.79	5
10	C2197	Unknown	0.00	0.84	0
12	C790	Month-day in date	-2.67	36.86	4
12	C1615	PHP web-development	0.99	9.38	5
12	C2205	Digit in document-initial Federal Reporter citation	-0.01	1.02	2
12	C2262	Long divider	0.06	0.92	1
13	C210	Toy/example code	-3.97	45.84	5
13	C528	So	3.43	53.65	5
14	C211	Second token position	4.76	59.64	4
14	C1109	Data schema / field type definition	1.94	24.06	5
14	C1846	Unknown	-0.54	3.13	0
14	C1937	Conditional compilation in C/C++ and other languages	-0.40	5.83	3
15	C1900	Present	-0.89	8.93	5
18	C1304	Nuclear medicine imaging	3.27	60.50	5
18	C1345	Sentence-final additive/concessive adverb pattern, anyway	2.27	22.12	5
20	C86	C	12.08	302.8	5
20	C736	Banking / lending	3.52	43.39	5
20	C949	Decorative aesthetics	-2.87	32.85	5
20	C1154	1960s–1970s American culture/history	2.77	41.62	5
21	C1514	){	-2.85	58.89	5
22	C618	After	4.80	82.84	5
22	C1380	Digit in numeric identifier	2.11	22.71	4
22	C1458	Greek letters in scientific notation	2.41	37.61	5
22	C1852	Am-, prefer capitalized, American	-2.12	48.57	5
24	C507	7	4.02	64.19	5
24	C776	Adverse health effects / medical complications	2.33	23.78	5
24	C1827	Expecting “include”	-1.06	7.87	5
24	C2018	JavaScript DOM property/method access	0.07	5.61	4
25	C1291	Anticipated future change	1.49	10.05	4
25	C1701	Salesforce development/customization	1.68	31.30	5
25	C1952	Unknown	-0.27	3.52	0

Table 6: Manual component annotations for Qwen 3.5 9B Base.

Layer	Component	Manual label	Skewness	Excess kurtosis	Confidence
0	C609	Store	-29.79	1594	5
0	C3472	Promise	-1.53	60.16	5
1	C3036	Do something / nothing	-0.79	41.92	5
1	C3298	spokesman	0.50	53.94	5
2	C231	User	-18.70	666.4	5
2	C2327	—	4.17	96.64	5
2	C3057	-y, -ie nickname	0.30	25.22	1
4	C565	Week / month	-9.49	276.6	4
4	C1156	Introduction	-11.61	618.1	5
4	C2219	Measurement/unit-conversion math question	-2.51	34.17	4
5	C3133	Diabetes	-0.67	16.44	5
6	C269	Figurative expression / metaphor	-6.82	127.0	5
6	C609	Data	-6.85	146.0	5
6	C992	Notice / note as information	7.44	229.8	5
6	C2199	Counterfactual / hypothetical conditional	4.88	160.7	5
7	C2650	/*	-6.81	376.6	5
7	C3045	Helicopter / rotorcraft	2.31	77.51	5
7	C3826	Unknown	-0.02	1.74	0
8	C935	Repeated structured code / machine output	3.07	27.33	4
9	C2175	Children's / preschool educational content	-2.68	58.13	4
10	C170	Footnote / reference marker	20.95	1463	5
10	C4002	Unknown	0.03	1.38	0
13	C637	Online forum / discussion thread	-4.52	87.23	4
13	C889	Commented-out code	5.52	175.4	4
15	C1201	Enumerating numbers	2.25	27.98	1
16	C61	Wikipedia category / category-page	10.41	230.4	4
16	C244	Repeated previously mentioned named entity	-3.25	22.96	4
16	C1365	Closing tag ¡/	-2.80	45.07	4
16	C3363	Unknown	-0.00	0.85	0
17	C530	Country/state/region name	-3.07	31.39	2
18	C3329	Mc- surname prefix	1.32	34.35	5
19	C146	Italian / Italy-related context	8.09	133.3	4
19	C1472	HTML DOM access/manipulation	-2.44	23.88	4
19	C2291	Unknown	1.85	17.55	0
19	C3174	Unknown	0.08	4.05	0
20	C352	Both	-10.24	283.2	5
20	C1145	Related to letter F or expecting F	2.76	23.25	1
20	C2259	Document/source attribution	-2.72	42.52	4
21	C704	Microsoft platform technical documentation	-3.94	46.69	4
22	C2808	Urination / bladder function	-2.18	28.84	4
22	C3385	Right-wing / conservative political context	-0.21	7.65	4
23	C1583	Limit	-4.81	111.5	5
25	C602	Lawsuit / civil litigation	-5.22	96.00	5
25	C1260	Maryland / Baltimore	7.15	231.3	4
25	C2695	Tell	3.42	76.78	5
25	C3939	Causal graphical-model reasoning	-0.70	18.01	3
26	C68	Suffix Sh-	-12.89	330.7	5
26	C1210	Deep-learning framework/model	4.81	95.20	5
28	C3025	Suffix Var-	-1.94	33.56	4
30	C2683	Expecting "see"	1.54	13.75	5

Tracing component behavior across context. Sentence-level traces provide one way to refine an initial component hypothesis. A component whose largest activation occurs at one token may still represent a broader phrase or contextual state, so we inspect its behavior across complete sequences when useful.

Sentence 1. She arrived at the library to study for her exam.

Sentence 2. Maya stopped at the bank before the trip, waiting in line to deposit a check and withdraw enough cash for the weekend.

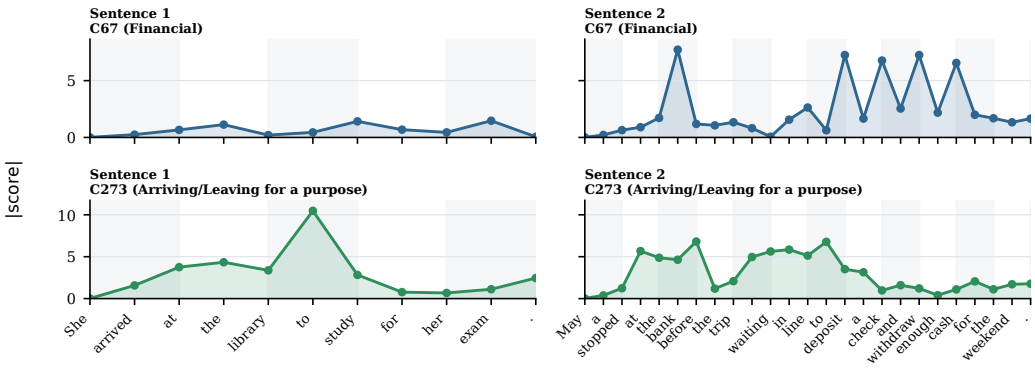


Figure 24: **Sentence-level traces help refine component interpretations.** Raw absolute scores are shown for GPT-2 Small layer 6 components C67 and C273. C67 follows financial context across several related tokens, while C273 responds to a broader arrival- and purpose-related construction.

Figure 24 illustrates two cases where the trace provides more information than the top-activating token alone. C67 is labeled as a financial component. Its score stays near zero in Sentence 1, which contains no financial context, but rises around *bank* in Sentence 2 and remains active near *deposit*, *check*, *withdraw*, and *cash*. The trace therefore supports a broader financial-context interpretation rather than a detector for the word *bank*.

C273 shows a more construction-level pattern. In Sentence 1, it rises across "arrived at the library to study" and peaks near the *to study* region, where motion, destination, and purpose come together. It also responds to related movement and purpose structure in Sentence 2. This behavior supports an arrival- or purpose-related interpretation rather than a label tied to a single token.

Example counterfactual checks. Table 7 illustrates the counterfactual stage of the same annotation workflow for five Qwen 3.5 9B Base components drawn from the random audit. For each component, we construct two examples expected to activate the proposed interpretation and two controls expected not to activate it. The controls retain relevant surface form or sentence structure where practical while changing the hypothesized concept.

We report each component's signed ICA score and its one-based rank by absolute score among all 4,096 components at the underlined target token. All ten predicted positive examples rank within the top four components at their target positions, whereas the ten controls rank between 94 and 3,582. These examples illustrate how controlled tests help refine and verify working labels.

Interactive annotation interface. To support systematic human inspection, we built a dedicated ICA Lens interface for annotating ICA components. For any supplied prompt, the interface shows the highest-scoring components at each token and links them directly to the evidence used for interpretation. Selecting a component and semantic tail opens its profile, including Logit Lens and R-Lens token readouts, distribution statistics, and representative high-scoring corpus occurrences. This interface allows annotators to move between model behavior, component scores, vocabulary readouts, and corpus evidence within a single workflow rather than inspecting each source separately. Figure 25 shows the interface for Qwen 3.5 9B Base at layer 27.

Table 7: **Examples from the counterfactual stage of the manual annotation workflow.** Five Qwen 3.5 9B Base components are tested on two positive and two negative examples each.

Component	Expected	Prompt	Score	Rank
L1/C3036: Do something / nothing				
L1/C3036	activate	The committee did nothing to prevent the closure.	−40.724	1
L1/C3036	activate	Please do something to fix the broken lock.	−24.828	3
L1/C3036	not activate	The committee knew about the closure.	+1.588	581
L1/C3036	not activate	Please use a wrench to fix the broken lock.	+1.123	1678
L9/C2175: Children's / preschool educational content				
L9/C2175	activate	Designed for preschoolers learning colors, this is a picture book.	−34.483	1
L9/C2175	activate	A cheerful alphabet song teaches young children their ABC.	−21.774	2
L9/C2175	not activate	Designed for graduate engineers studying thermodynamics, this is a reference book.	−0.334	2991
L9/C2175	not activate	The television affiliate announced a change to its ABC schedule.	+0.491	2913
L19/C146: Italian / Italy-related context				
L19/C146	activate	Rome, Milan, and Naples are cities in Italy.	+10.361	4
L19/C146	activate	She ordered pasta and espresso while traveling through Italy.	+10.687	3
L19/C146	not activate	Beijing, Shanghai, and Guangzhou are cities in China.	+1.107	898
L19/C146	not activate	She ordered sushi and tea while traveling through Japan.	−0.132	3582
L25/C1260: Maryland / Baltimore				
L25/C1260	activate	The court issued its ruling under the laws of Maryland.	+25.503	1
L25/C1260	activate	After leaving Annapolis, the train continued toward Baltimore.	+23.690	1
L25/C1260	not activate	The court issued its ruling under the laws of California.	+0.252	2904
L25/C1260	not activate	After leaving Providence, the train continued toward Boston.	−0.729	1285
L26/C1210: Deep-learning framework/model				
L26/C1210	activate	Built with PyTorch, the classifier is a deep-learning model.	+17.809	2
L26/C1210	activate	Using TensorFlow, we trained a convolutional neural network.	+24.952	1
L26/C1210	not activate	Built from cardboard, the display is an architectural model.	+1.733	259
L26/C1210	not activate	Using fiber optics, the offices share a communication network.	+1.896	94

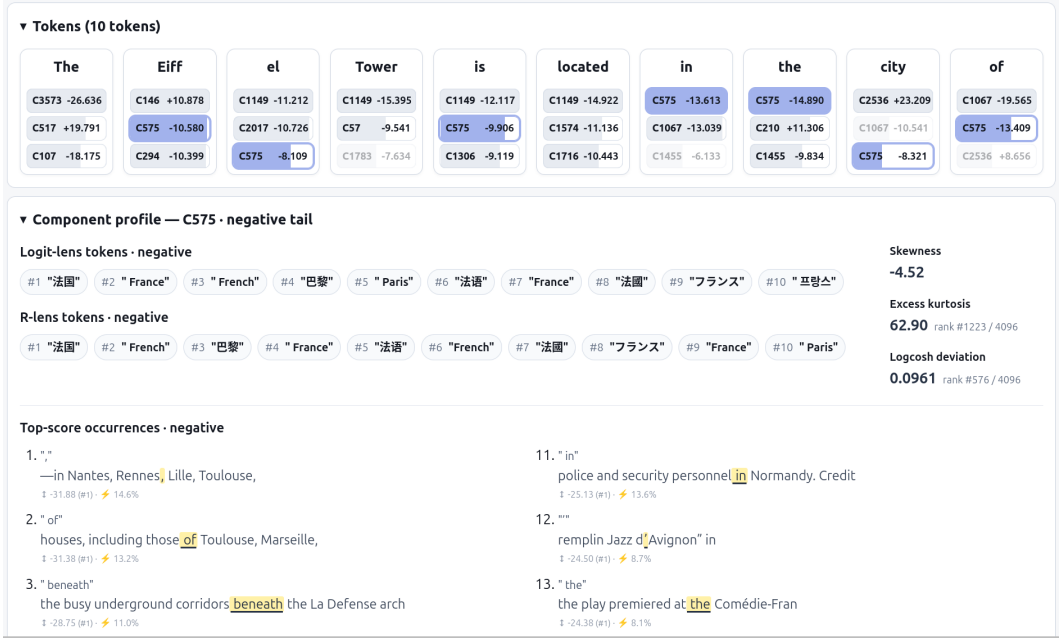


Figure 25: **ICA Lens interactive annotation interface.** For the prompt "The Eiffel Tower is located in the city of", the upper panel shows the highest-scoring ICA components at each token for Qwen 3.5 9B Base, layer 27. Selecting component C575's negative semantic tail opens its profile below, including Logit Lens and R-Lens token readouts, skewness, excess kurtosis, Logcosh deviation, and representative high-scoring corpus occurrences. The interface is used to support systematic component inspection and annotation.

C.3 Automated Interpretability Protocol

We next use explanation–prediction evaluation to extend interpretability measurement to a larger sample of components. An explainer receives high-activating text fragments and their component scores and produces a concise description of the activation pattern. A predictor then receives that explanation and predicts component activations on held-out fragments. We report the Pearson correlation between predicted and observed activations, so a high score requires the explanation to generalize beyond the examples used to construct it.

For the historical Pythia 70M reproduction, we follow the released explanation–prediction protocol of Huben et al. (2024). GPT-4.1-mini is used for both explanation generation and held-out activation prediction. For Qwen 3.5 9B Base, `thinkingmachines/Inkling` generates explanations and `Qwen3.8-27B` predicts held-out activations. The arbitrary-sign and skewness-selected ICA conditions use exactly the same fitted axes and differ only in which signed tail is presented to the evaluator.

We compare usable ICA and SAE features under matched sampling rules. For Pythia, we follow the released procedure by sampling 150 feature IDs uniformly from the full SAE dictionary and skipping dead features. For Qwen, we first restrict each representation to features that are active on the shared fragment pool and then uniformly sample 150 features from the eligible set. Eligibility requires at least 20 top-activation records and 20 nonzero random records for both ICA and SAE features.

For the fitting-trajectory analysis in Figure 5, we follow the same 100 Qwen components from each of layers 7, 15, 23, and 31 across FastICA checkpoints. At every checkpoint, we retrieve that checkpoint's own top 25 occurrences from distinct documents. Five fragments are used to generate an explanation and the remaining 20 are used for activation prediction. This checkpoint-specific retrieval lets the evaluation follow the component as its direction changes during fitting.

C.4 ICA–SAE Reproduction and Robustness Checks

We next test whether the interpretability comparison between ICA and SAE features is stable across changes in ICA fitting budget, training corpus, and evaluator pipeline. These experiments extend the historical Pythia 70M reproduction and separate the effect of ICA sign orientation from other choices in the evaluation setup.

The sign effect is stable under a much smaller ICA fitting budget. The released Pythia reproduction uses 2,098,176 fitting activations and up to 200 FastICA iterations. We compare it with a substantially smaller configuration using 524,288 activations and 20 iterations. Figure 26 shows that the two fitting budgets produce closely aligned interpretability curves on both Pile-10k and OpenWebText. Reducing both the sample count and iteration budget changes the aggregate skewness-selected ICA score by only about 0.02 under either corpus, while the difference between arbitrary-sign and skewness-selected ICA remains much larger.

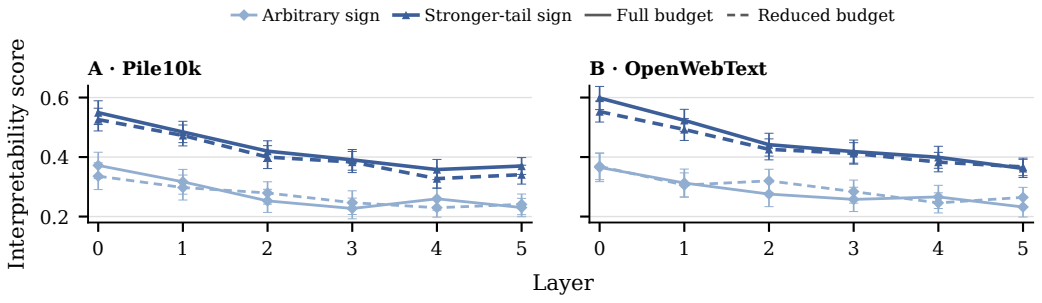


Figure 26: **The sign correction is stable under a substantially reduced ICA fitting budget.** Solid curves use the released-code fitting budget of 2,098,176 activations and up to 200 iterations, while dashed curves use 524,288 activations and 20 iterations. The two budgets produce similar interpretability curves on both Pile-10k and OpenWebText, while the gap between arbitrary-sign and skewness-selected ICA remains large.

The same pattern persists across corpora and evaluator pipelines. Figure 27 provides a triangular robustness comparison. Panels A and B keep the Inkling–Qwen3.8-27B evaluator fixed while changing the ICA and SAE training corpus from Pile-10k to OpenWebText. Panels B and C keep OpenWebText training fixed while changing the evaluator to GPT-4.1-mini. Absolute interpretability scores move across these settings, but resolving the arbitrary ICA sign consistently recovers a large part of the measured gap to the SAE baseline and yields comparable scores at several layers.

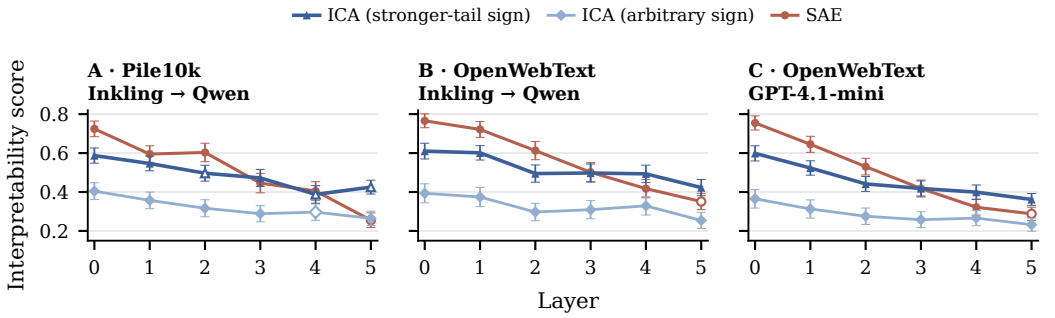


Figure 27: **The skewness-selected ICA comparison is stable across training corpora and evaluator pipelines.** (A–B) We hold the Inkling–Qwen3.8-27B evaluator fixed while changing the ICA and SAE training corpus from Pile-10k to OpenWebText. (B–C) We hold OpenWebText training fixed while changing the evaluator to GPT-4.1-mini. Absolute scores vary across settings, but resolving ICA's arbitrary sign produces the same qualitative improvement throughout. Hollow markers denote incomplete evaluation sets described in the text.

In panel A, the final SAE point uses $n = 144$, arbitrary-sign ICA at layer 4 uses $n = 149$, and skewness-selected ICA at layers 2, 4, and 5 uses $n = 149$. The final-layer SAE points in panels B and C use $n = 143$; all other points use $n = 150$.

Together, these controls show that the gain from resolving ICA's sign ambiguity is stable across substantial changes to fitting compute, training corpus, and evaluator choice.

C.5 Error Analysis of the Relationship Between Non-Gaussianity and Automated Interpretability

Automated interpretability and non-Gaussianity are positively associated overall, but individual components can deviate substantially from this trend. To understand these outliers, we inspect components with high excess kurtosis but low automated interpretability, as well as components with low excess kurtosis but relatively high automated interpretability. We identify two recurring failure modes in the automated evaluation: insufficient coverage of sparse semantic tails and dataset-specific shortcuts that make some components easier to predict.

Figure 28 compares three representative regimes at FastICA iteration 50. Each histogram shows the signed component-score distribution over the one-million-token Pile-10k fitting population. The dashed Gaussian curve provides a reference distribution. The red selection cutoff marks the maximum token activation of the weakest among the 25 distinct-document OpenWebText fragments admitted to explanation and held-out evaluation, expressed on the standardized scale of the fitting population. This view shows whether the evaluation examples actually reach the component tail that drives its non-Gaussianity.

Well-covered semantic tails produce agreement. Figure 28A shows Qwen L7/C338. Its selected examples consistently concern courts and judicial proceedings, and the evaluation fragments reach the same high-activation tail visible in the fitting distribution. The resulting explanation therefore predicts held-out activations well, giving both high non-Gaussianity and high automated interpretability.

Sparse semantic tails can be missed by the evaluation corpus. Figure 28B shows Qwen L7/C2035. The component has fitting-population excess kurtosis $\kappa = 665.34$ and reaches a standardized score of 88.03. In a systematic 100,000-token diagnostic sample, all seven scores above 20

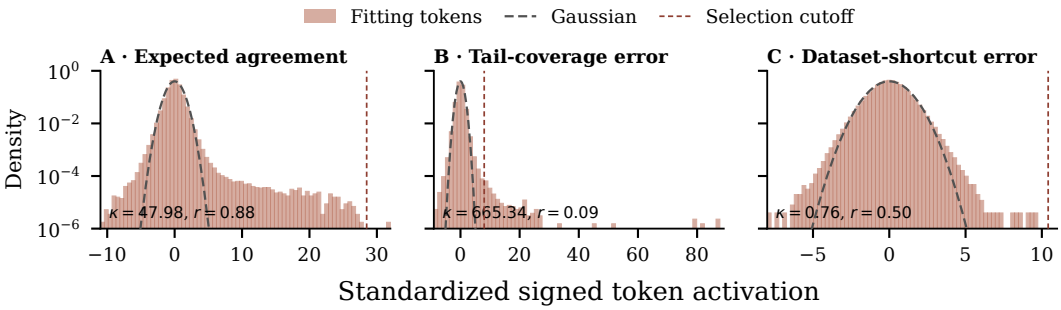


Figure 28: **Three regimes of agreement between non-Gaussianity and automated interpretability.** Histograms show signed component scores on the one-million-token Pile-10k fitting population; gray dashed curves show standard Gaussian densities. The muted-red dashed line marks the weakest selected OpenWebText evaluation fragment on the same standardized scale. Insets report excess kurtosis κ and automated interpretability score r. Panels show expected agreement (A), a sparse-tail coverage error (B), and a dataset-shortcut error (C).

occur in C/C++ preprocessor contexts, primarily header guards and conditional compilation. The component therefore contains an extremely sparse but coherent tail. The 50,000-fragment OpenWebText candidate pool does not reach comparable examples: the weakest selected top fragment has a standardized score of only 8.02, and the explanation examples instead contain more moderate activations from unrelated contexts. The automated evaluator consequently sees little of the rare structure responsible for the component's high kurtosis.

Dataset regularities can make lower-kurtosis components predictable. Figure 28C shows Qwen L15/C1503. The explainer initially proposes a lexical pattern involving words such as *Human*, *What*, and *How*, which appear at position 0 in the selected examples and receive raw activations of 11.82, 11.75, and 11.74. Controlled examples reveal a broader positional pattern: the same words activate near zero when placed later in the sequence, while unrelated position-0 tokens such as *The*, *I*, and *will* still receive activations of 8.99, 8.68, and 7.59. The evaluation examples therefore contain a regularity that makes the component predictable even though the initial lexical explanation does not capture the underlying positional structure.

These cases suggest that automated interpretability can underestimate the relationship between non-Gaussianity and human interpretability. Sparse but coherent semantic tails may be missed by the sampled evaluation contexts, while dataset-specific regularities can inflate scores for less meaningful patterns. The observed correlation with automated interpretability may therefore be a conservative estimate of the underlying relationship.

These failure modes also suggest a concrete direction for improved evaluation. A coverage-aware protocol could guarantee examples from the selected semantic tail across multiple activation quantiles and distinct documents, evaluate the resulting explanation on a second corpus to reduce dataset-specific shortcuts, and validate a subset with blinded human judgments. We leave systematic validation of such a protocol to future work.

D Reading and Intervening with Non-Gaussian Directions

Once non-Gaussian directions have been labeled, they provide a coordinate system for both reading and editing model activations. We can decompose new activations into labeled component mixtures to inspect how information is represented, or modify selected coordinates to test their effect on model behavior. We demonstrate these two uses through contextual decomposition, embedding analysis, and generation-time interventions.

D.1 Reading Component Mixtures

D.1.1 Contextual Decomposition of Polysemy

Polysemous words provide a useful setting for studying contextual decomposition because the surface token stays fixed while the surrounding evidence changes. We use a paragraph containing four occurrences of *bank*: two referring to a financial institution, denoted F1 and F2, and two referring to a river edge, denoted R1 and R2.

> Maya stopped at the bank [F1] before the trip, waiting in line to deposit a check and withdraw enough cash for the weekend. The teller asked about her travel plans, and Maya said she was driving north to visit the old village by the river. By sunset she was sitting on the grassy bank [R1] with her shoes beside her, watching the water curl around stones and reeds. It amused her that the same word had followed her all day: first a bank [F2] with counters, accounts, and vaults, then a bank [R2] of earth holding the river in place.

Figure 29 decomposes the four *bank* tokens into their largest-absolute-score ICA components at the embedding layer and several GPT-2 Small residual-stream layers. Each cell lists the top five components at that token position, with bar length showing relative score magnitude within the cell and the annotation badge showing component type and confidence.

The embedding row provides a simple control. All four columns are identical because the input embedding depends only on the token identity *bank*. Its strongest components include directions labeled *Bank, Mortgage*, *A Room or Place*, *Group, Team*, and *Letter B*. Before contextual processing begins, the four occurrences therefore share the same lexical mixture.

The residual-stream representations separate as context accumulates. F1 appears after the prefix "Maya stopped at the," which already supports a financial reading before later phrases such as "deposit a check" and "withdraw enough cash" become available. Later layers contain components related to finance, organizations, companies, investment, and departments. R1 instead follows a context involving a village and river, with the immediate prefix "sitting on the grassy." Its mixture shifts toward directions related to water, place, posture, movement, and outdoor space.

F2 returns to the financial sense but appears after both meanings of *bank* have already been introduced. Its representation combines financial components such as *Business*, *Organization*, *Pay*, and *Investment* with local phrase-level and discourse-related components. The same token can therefore combine semantic content with syntax and discourse state rather than mapping to a single sense feature.

R2 also highlights the importance of autoregressive context. A human reader identifies the phrase as a river edge after seeing "bank of earth holding the river in place," but at the target token the model has only seen the prefix ending in "then a." That prefix still contains the earlier financial use and explicitly sets up a contrast. Finance-related components at R2 are therefore consistent with the information available to the model at that position.

Overall, ICA does not reduce polysemy to one financial-versus-river axis. The same surface token is represented by a changing mixture of lexical, syntactic, semantic, positional, and longer-context directions, and that mixture evolves with the information available at each token position and layer.

D.1.2 Lexical Structure in Token Embeddings

The previous analyses study contextual residual-stream representations. We now remove context and apply the same component view to token embeddings, where each token has a fixed representation before any transformer layer is applied.

As an additional test on a newer model outside the main three-model evaluation, we fit ICA to the complete token-embedding matrix of Qwen 3.5 2B Base. No vocabulary sampling is used. This setting asks whether the same non-Gaussian decomposition can expose readable lexical structure directly in the embedding space.

Classical word-embedding analyses often summarize lexical relations through vector offsets, such as $king - man + woman \approx queen$. ICA provides a complementary view by decomposing each token into multiple readable directions rather than summarizing the relation with one displacement.



Figure 29: **Contextual decomposition of a polysemous word in GPT-2 Small.** We probe two financial uses of *bank* (F1, F2) and two river-edge uses (R1, R2). Each cell lists the five largest-absolute-score ICA components at the target token. The embedding representation is identical across occurrences, while later layers separate into different mixtures of lexical, syntactic, semantic, positional, and longer-context components.

Figure 30 shows several familiar lexical pairs. For *king* and *queen*, both embeddings contain a strong *Royal* component, while the remaining mixture separates them through directions such as *Woman, Female*, letter identity, morphology, and common-token structure. The pair is therefore represented as shared structure together with additional token-specific factors.

The same pattern appears across other lexical families. *Father* and *mother* share a strong *Parent* component and related family directions, while gender-associated and birth-related components distinguish them. *Brother* and *sister* share several social and family-relation components. *Actor* and *actress* share acting-, film-, and creation-related directions, while *actress* additionally activates a strong *Woman, Female* component.

The component view therefore exposes both shared structure and asymmetry. Familiar lexical relations are represented by mixtures of semantic, morphological, surface-form, and other token-specific directions rather than by a single isolated axis. This extends the same reading framework used for contextual residual streams to the model's static lexical representation.

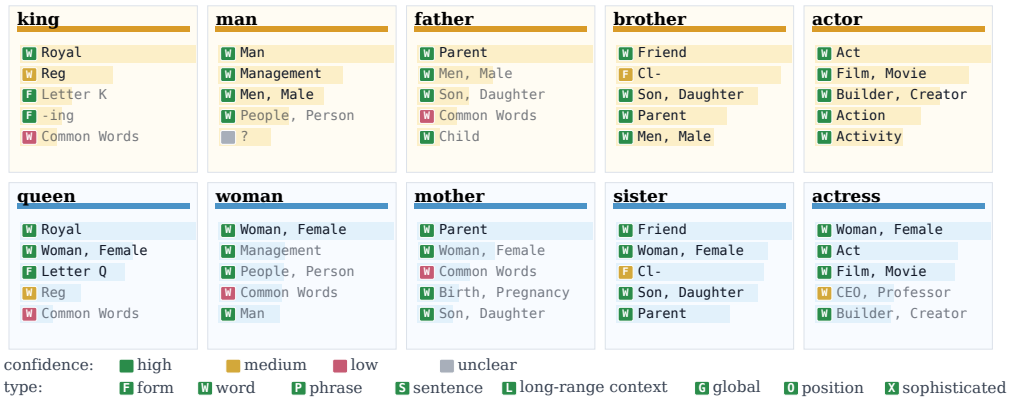


Figure 30: **Embedding-layer ICA decomposition of familiar lexical relations in Qwen 3.5 2B Base.** Each box lists the five largest-absolute-score ICA components for one token embedding. Related tokens share interpretable directions while differing along additional semantic, gender-associated, morphological, and surface-form components.

D.2 Illustrative Generation Interventions

The analyses above use labeled ICA coordinates to read model activations. We next use the same coordinates for generation-time intervention. We consider two operations: additive steering along an ICA writing direction and coordinate clamping in the fitted ICA basis.

Selecting intervention components. The GPT-2 example uses layer 7 component C317, selected from the labeled components for its weekend-activity pattern. For Gemma and Qwen, we select language-related components from 1,000 seeded English–target-language sentence pairs from Many-Things. For each sentence, we extract the residual activation at the final non-padding text token, excluding EOS, and transform it into ICA coordinates.

For component c, we compute

$$\Delta(c) = \overline{y}_{\text{target}}(c) - \overline{y}_{\text{English}}(c), \tag{17}$$

where $\overline{y}_{\text{target}}(c)$ and $\overline{y}_{\text{English}}(c)$ are the mean ICA scores over the 1,000 target-language and English sentences. We rank components by $|\Delta(c)|$, so the selected component is the one that most strongly separates the two languages within the searched layers. The sign of $\Delta(c)$ indicates which side of the ICA axis is associated with the target language. Searching Gemma layers 18–22 selects layer 20 component C105 for Chinese versus English, while searching Qwen layers 23–27 selects layer 26 component C9 for French versus English.

Additive steering. Let d_j be the residual-space writing direction of component j, and let α denote the desired change in its ICA coordinate. Without a cap, additive steering is

$$x' = x + \alpha d_j. \tag{18}$$

The intervention is applied at the current generation position: the final prompt token during prefill and, when continued steering is enabled, each newly generated token during decoding.

For GPT-2 C317, we use $\alpha = -20$ at the final prompt token and no further steering. For Gemma C105, we use $\alpha = +35$ at the final prompt token. For Qwen C9, we use an initial $\alpha = -40$ followed by $\alpha = -20$ during decoding. The sign of the steering strength follows the target-language side identified by the selected ICA component.

Coordinate clamping. Additive steering moves the activation along a selected writing direction, while clamping directly removes that coordinate from the current representation. Given a residual activation, we transform it into ICA coordinates, set the selected component score to zero, map the

Table 8: **Generation interventions with individual ICA components.** GPT-2 C317 is a labeled weekend-activity component. Gemma C105 and Qwen C9 are selected from 1,000 English–target-language pairs as the components with the largest absolute difference in mean ICA score. Additive steering changes the selected coordinate along its writing direction, while clamping sets that coordinate to zero. All generations use greedy decoding.

Model	Prompt	Baseline continuation	Intervened continuation
GPT-2 Small L7/C317	*Prompt:* "My favorite thing to do on the weekend is to"	*Base:* "go to a local bar and have a drink."	*Steer:* "play with the kids."
Gemma 2 2B L20/C105	*EN:* "Hangzhou is a historic city known for" *ZH:* "杭州是一座历史悠久的城市，它以"	*EN:* "its beautiful scenery and rich culture." *ZH:* "其悠久的历史和丰富的文化遗产而闻名。"	*EN to ZH, steer:* "其历史悠久的文化和美丽的风景。" *ZH to EN, ablate:* "its rich history and culture as well as its beautiful scenery."
Qwen 3.5 9B Base L26/C9	*EN:* "Deep learning is a machine learning technique that" *FR:* "L'apprentissage profond est une technique d'apprentissage automatique qui"	*EN:* "uses artificial neural networks to learn from data." *FR:* "utilise des réseaux de neurones artificiels pour apprendre à partir de données."	*EN to FR, steer:* "utilise des réseaux de neurones artificiels pour apprendre à partir de données." *FR to EN, ablate:* "uses neural networks to learn from data."

modified coordinates back to residual space, and restore the original residual norm. For the Gemma and Qwen language examples, this clamp is applied throughout generation.

The GPT-2 example generates up to 10 new tokens, while the Gemma and Qwen examples generate up to 24. In these examples, steering moves generation toward the behavior associated with the selected component, while clamping the language-related coordinate shifts target-language prompts toward English. These examples connect labeled non-Gaussian directions to direct edits of model behavior.

E Full Quantitative Evaluation of Non-Gaussian Directions

The main text summarizes reconstruction, sparse probing, and targeted probe perturbation (TPP) after averaging across datasets and evaluated layers. Here we give the full evaluation protocol and disaggregate the results by dataset, model depth, and metric. The breakdowns show that the main trends are not driven by a small number of evaluation settings.

E.1 Evaluation Protocol and Aggregation

Table 9 summarizes the layer coverage for the three evaluations. Reconstruction is evaluated at every model layer, while sparse probing and TPP use two representative layers per model. All downstream evaluation datasets are separate from the Pile-10k activations used to fit ICA.

At each top-k budget, we first average the evaluated layers within each dataset. The main-text curves then average equally across datasets, with shaded regions showing ± 1 standard deviation across datasets. The figures below report the corresponding dataset-level results, and reconstruction is additionally disaggregated across model depth.

Evaluation	**GPT-2 Small**	**Gemma 2 2B**	**Qwen 3.5 9B Base**
Reconstruction	All layers	All layers	All layers
Sparse probing	6, 10	12, 20	12, 20
TPP	6, 10	12, 20	12, 20

Table 9: **Layer coverage for representation-coordinate evaluations.** Reconstruction uses all layers, while sparse probing and TPP use two representative layers per model.

Reconstruction datasets. We evaluate reconstruction on six datasets spanning news, general text, code, multilingual text, and dialogue:

- `fancyzhx/ag_news`;

- `Salesforce/wikitext` (`wikitext-103-raw-v1`);
- `codeparrot/github-code`;
- `wikimedia/wikipedia` (`20231101.es`; Spanish Wikipedia);
- `wikimedia/wikipedia` (`20231101.zh`; Chinese Wikipedia); and
- `HuggingFaceH4/ultrachat_200k`.

Sparse-probing datasets. We use eight SAEBench sparse-probing datasets:

- `LabHC/bias_in_bios_class_set1`;
- `LabHC/bias_in_bios_class_set2`;
- `LabHC/bias_in_bios_class_set3`;
- `canrager/amazon_reviews_mcauley_1and5`;
- `canrager/amazon_reviews_mcauley_1and5_sentiment`;
- `codeparrot/github-code`;
- `fancyzhx/ag_news`; and
- `Helsinki-NLP/europarl`.

TPP datasets. We evaluate TPP on `LabHC/bias_in_bios_class_set1` and `canrager/amazon_reviews_mcauley_1and5`.

E.2 Sparse Reconstruction

Sparse reconstruction asks how much of a held-out residual activation can be recovered from only a small number of representation coordinates. For a linear dictionary such as ICA, PCA, or a random basis, let y_j be the signed score for coordinate j and d_j its residual-space writing direction. We rank coordinates by their contribution magnitude,

$$\mathrm{contrib}_j = |y_j| \, \|d_j\|_2, \tag{19}$$

retain the original signed scores for the top-k coordinates, set the rest to zero, and reconstruct

$$\hat{x} = \mu + \sum_{j \in S_k} y_j d_j, \tag{20}$$

where μ is the fitted center and S_k is the selected coordinate set. We also evaluate the complete-basis reconstruction at $k = d$.

For SAE baselines, we first encode with the published checkpoint using its native nonlinearity or TopK mechanism. We then rank the resulting non-negative latent codes by the same contribution criterion, retain the top-k active codes, and decode with the published SAE decoder. Thus, top-k truncation is applied after the SAE's native encoder sparsity rather than replacing it.

The reconstruction plots use the mean number of nonzero retained features per token on the horizontal axis. For the dense linear bases, this equals the requested k. For an SAE, the effective feature count can be smaller than the requested budget when the native encoder already produces fewer nonzero latents. The endpoint stars show complete-basis reconstruction for ICA, PCA, and random directions, and the native sparse encode–decode reconstruction for each SAE.

E.2.1 Across Datasets

Figure 31 reports cosine-similarity reconstruction separately for every model–dataset pair. The same broad ordering appears across news, general text, code, multilingual Wikipedia, and dialogue. At small coordinate budgets, ICA consistently reconstructs better than PCA and random bases across all three models. The public SAE baselines are often strongest at sparse budgets, but ICA remains substantially closer to SAE than to the generic linear baselines across domains.

This consistency is notable because ICA is fitted once per model layer and is not adapted to any of these downstream datasets. The recovered non-Gaussian coordinates therefore preserve useful activation structure across a wide range of held-out text distributions.

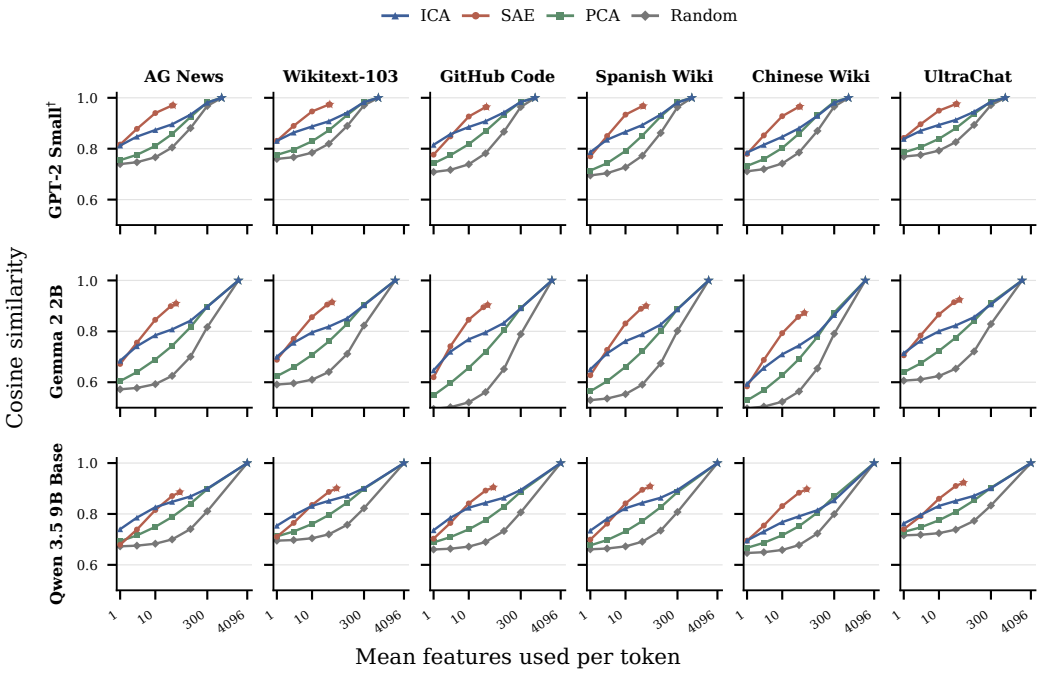


Figure 31: **Sparse reconstruction across evaluation domains.** Cosine similarity is shown separately for every model–dataset pair. ICA consistently outperforms PCA and random bases at sparse budgets and remains close to the public SAE baselines across six evaluation domains. Rows correspond to models and columns to datasets.

E.2.2 Across Layers

Figure 32 disaggregates reconstruction across model depth. For the five panels we use residual-stream layers $\{0, 3, 5, 8, 11\}$ for GPT-2 Small, $\{0, 6, 12, 19, 25\}$ for Gemma 2 2B, and $\{0, 8, 16, 23, 31\}$ for Qwen 3.5 9B Base, corresponding to the layers nearest 0%, 25%, 50%, 75%, and 100% of each stack.

ICA remains a strong sparse reconstruction basis throughout depth. Its advantage over random directions is large across nearly all layers, and it also generally improves over PCA at small budgets. The exact gap to SAE varies across models and depth, but the ICA advantage over generic linear bases is not confined to one part of the network.

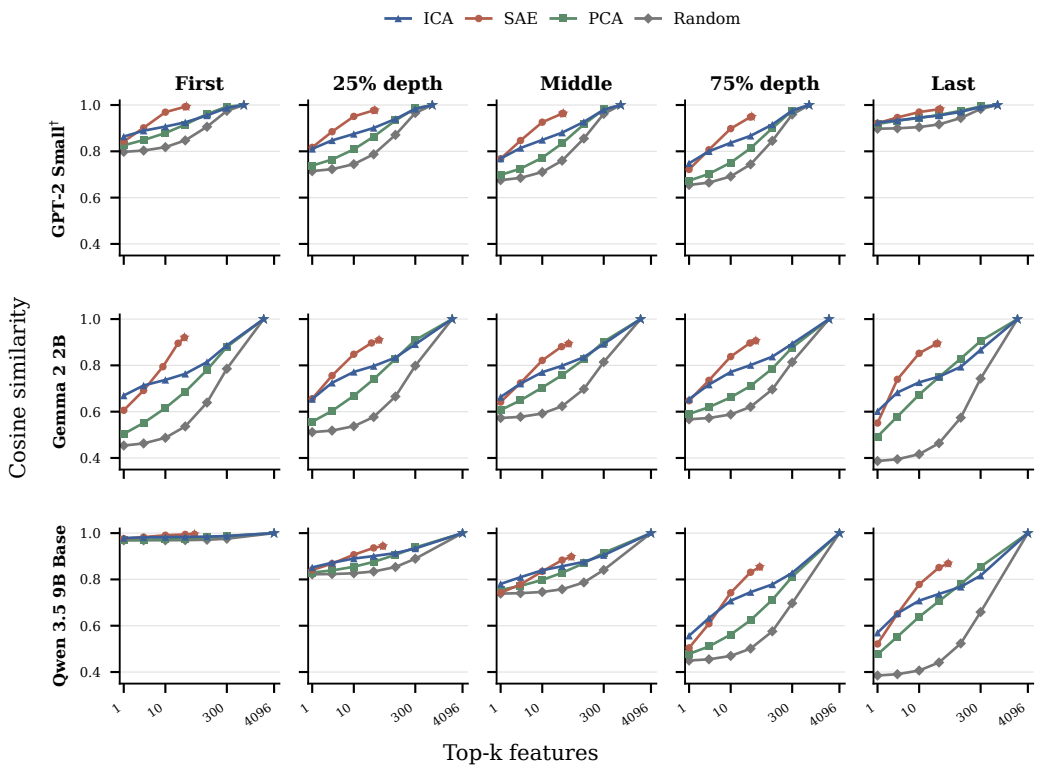


Figure 32: **Sparse reconstruction across model depth.** Cosine similarity is averaged across evaluation datasets at five normalized depth positions. ICA retains a strong sparse-reconstruction advantage over generic linear baselines across depth, while its gap to SAE varies by layer and model.

E.2.3 Alternative Metric

Figure 33 repeats the aggregate comparison using normalized mean squared error (NMSE), where lower values are better. The qualitative ordering agrees with cosine similarity. At sparse budgets, ICA generally has lower reconstruction error than PCA and random bases across all three models, while SAE is often strongest at intermediate sparse budgets. The agreement across metrics shows that the reconstruction result is not specific to cosine similarity.

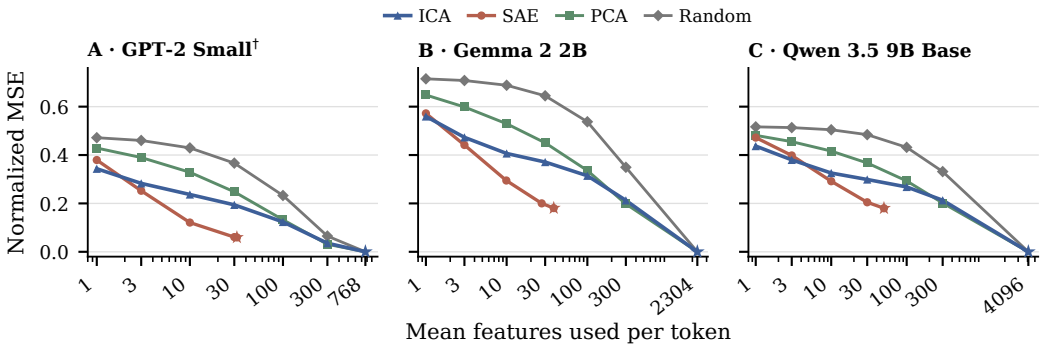


Figure 33: **Sparse reconstruction under an alternative metric.** Held-out top-k reconstruction measured by NMSE, averaged equally across evaluated layers and datasets. Lower is better. The comparison gives the same broad ordering as cosine similarity.

E.3 Sparse Probing

Sparse probing asks whether concept-relevant information can be read from a small number of representation coordinates. We use the SAEBench sparse-probing evaluator with feature budgets

$$k \in \{1, 2, 5, 10, 20, 50, 100\}, \tag{21}$$

a 4,000-example training split, a 1,000-example test split, and context length 128. The reported sparse probes are L2-regularized logistic-regression classifiers.

Feature selection follows the SAEBench protocol. Let $\ell \in \{+, -\}$ denote the binary class label. For each coordinate j, we compute the absolute difference in its mean activation between the positive and negative classes,

$$\text{score}_j = |\mathbb{E}[a_j \mid \ell = +] - \mathbb{E}[a_j \mid \ell = -]|\,, \tag{22}$$

and retain the top-k coordinates before fitting the probe. The same selector is used for ICA, SAE, PCA, and random features.

For sparse probing, ICA exposes signed, skewness-oriented coordinates. If y_j is the raw ICA score, $s_j \in \{-1, +1\}$ is the orientation selected from the fitting distribution, and d_j is the corresponding writing direction, the feature passed to SAEBench is

$$a_j = (s_j y_j)\, \|d_j\|_2. \tag{23}$$

Additional baselines. We further broaden the comparison on Gemma 2 2B layer 12. **Matryoshka SAEs** test whether smaller effective SAE dictionaries improve sparse readout under compact feature budgets (Bussmann et al., 2025), while **ITDA** provides another training-light, non-SAE activation-dictionary baseline (Leask et al., 2025). Figure 34 shows that ICA and the full SAE form the strongest pair across the tested budgets: ICA has a small advantage at the smallest budgets, while the full SAE becomes slightly stronger as k increases. Both remain clearly above the compact Matryoshka variants and ITDA in this setting. This comparison broadens the sparse-probing result beyond a single SAE baseline and shows that non-Gaussian coordinates provide strong concept readout relative to both compact learned dictionaries and another training-light alternative.

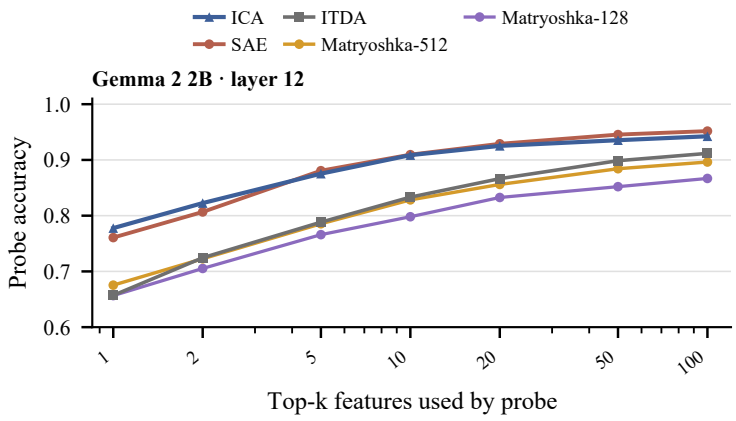


Figure 34: **Sparse probing with additional baselines on Gemma 2 2B layer 12.** We compare ICA with the public SAE, compact Matryoshka SAE variants, and ITDA. ICA and the full SAE achieve the strongest sparse-readout accuracy across feature budgets, with ICA slightly stronger at the smallest budgets and the SAE slightly stronger as the budget increases.

Dataset-level results. Figures 35–37 disaggregate the main-text result by dataset. ICA and SAE are close on many tasks, with the relative advantage depending on the concept and feature budget. For GPT-2 Small, ICA is particularly strong at small budgets on Amazon sentiment, GitHub code, AG News, Amazon category, and several Bias in Bios tasks, while SAE has a clear advantage on Europarl language identification. Gemma 2 2B and Qwen 3.5 9B Base show a more mixed pattern, with small advantages alternating across tasks.

Across models, a small number of ICA coordinates often support strong concept prediction. As k increases, the methods generally approach similar probe accuracy, suggesting that much of the concept-relevant information is concentrated in a small subset of non-Gaussian directions.

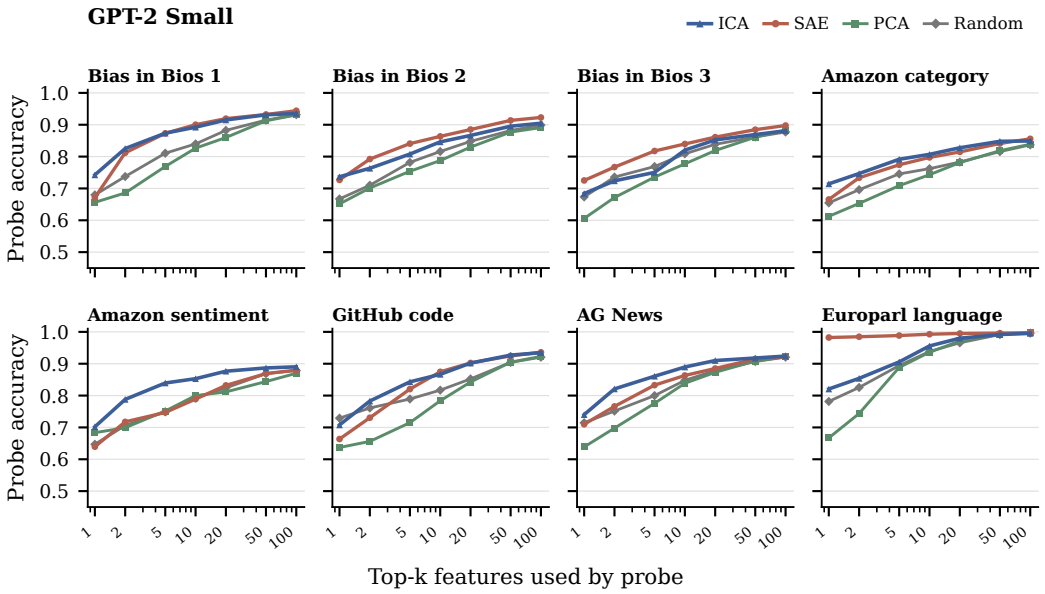


Figure 35: **Dataset-level sparse probing for GPT-2 Small.** Each panel reports one SAEBench dataset and averages equally over layers 6 and 10. ICA is especially competitive at small feature budgets across several tasks.

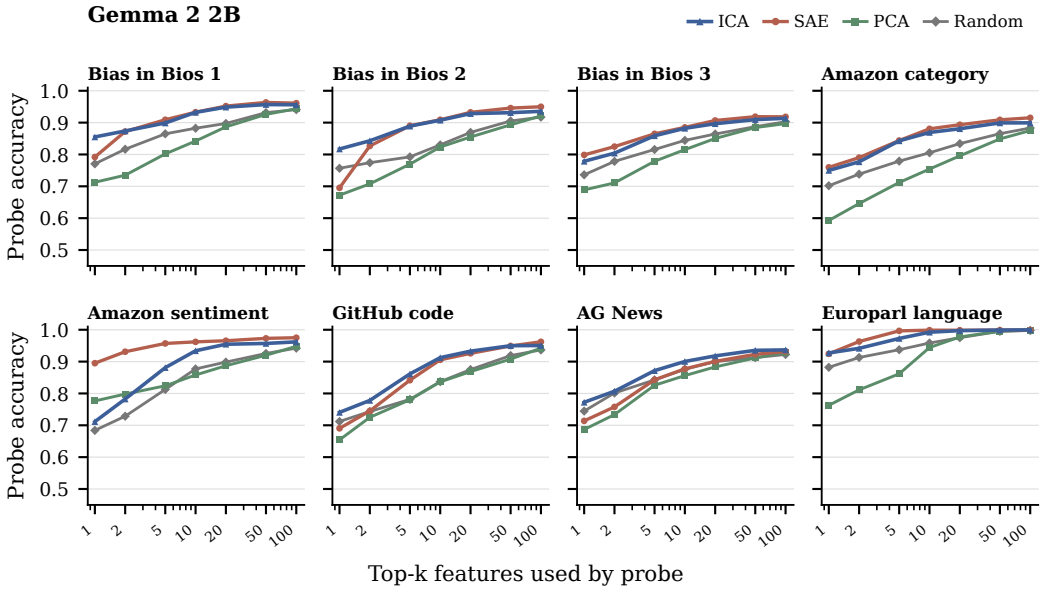


Figure 36: **Dataset-level sparse probing for Gemma 2 2B.** Each panel averages equally over layers 12 and 20. ICA and SAE remain close across most tasks, with the relative advantage depending on the dataset and feature budget.

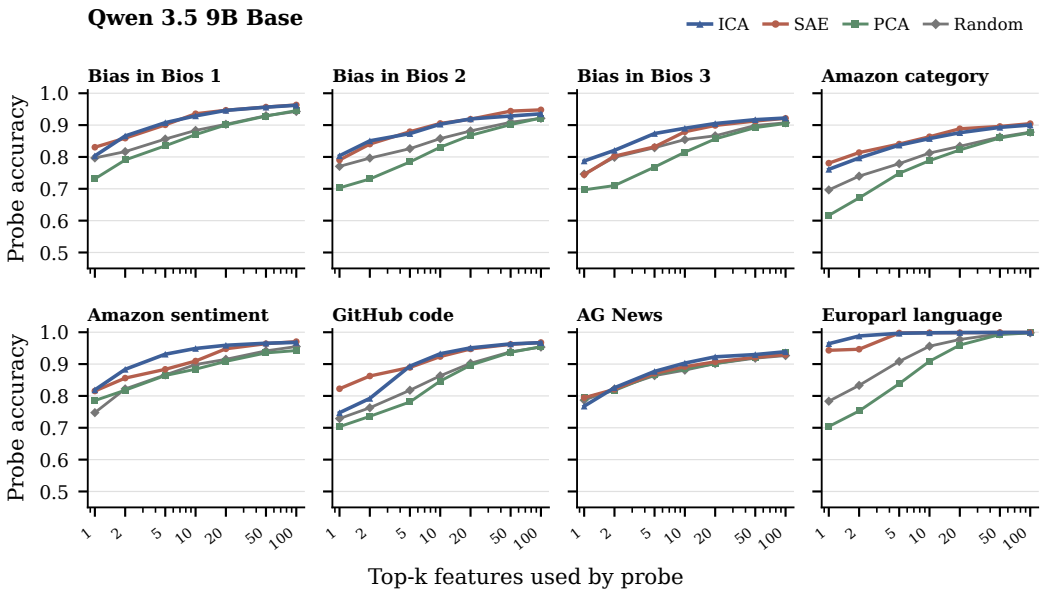


Figure 37: **Dataset-level sparse probing for Qwen 3.5 9B Base.** Each panel averages equally over layers 12 and 20. ICA and SAE are broadly competitive across datasets, particularly once more than a few coordinates are available.

E.4 Targeted Probe Perturbation

Sparse probing measures whether a small number of coordinates can predict a concept. Targeted probe perturbation asks whether those coordinates can also be removed selectively. For a target class c and an ablation budget k, SAEBench defines the intended effect as the drop in the target probe's accuracy,

$$\Delta_{\text{intended}}(c,k) = \text{Acc}_{\text{clean}}(c) - \text{Acc}_{\text{ablated}\to c}(c), \tag{24}$$

and the unintended effect as the mean accuracy drop over all non-target probes,

$$\Delta_{\text{unintended}}(c,k) = \frac{1}{|C \setminus \{c\}|} \sum_{c' \neq c} \left[\text{Acc}_{\text{clean}}(c') - \text{Acc}_{\text{ablated}\to c}(c')\right]. \tag{25}$$

Here, C is the set of evaluated classes, and $\text{Acc}_{\text{ablated}\to c}(c')$ is probe c''s accuracy after ablating the coordinates selected for target class c. The TPP score is

$$\text{TPP}(c,k) = \Delta_{\text{intended}}(c,k) - \Delta_{\text{unintended}}(c,k). \tag{26}$$

A high score therefore requires a large effect on the intended concept together with small collateral effects on unrelated concepts.

SAEBench ranks candidate features separately for each target class. For feature j, the ranking combines its class-conditional activation difference with the corresponding probe readout,

$$\text{effect}_j = \max\left(\bar{a}_j^+ - \bar{a}_j^-, 0\right)\left(w_{\text{probe}}^\top d_j\right), \tag{27}$$

and the top-k nonzero effects are selected for ablation. Here, $\bar{a}_j^+$ and $\bar{a}_j^-$ are the mean feature activations for the positive and negative examples of target class c, w_{probe} is the target probe's residual-space readout, and d_j is the feature's decoder direction.

For TPP, each oriented ICA component is exposed as a one-sided non-negative feature,

$$a_j = \max(s_j y_j, 0)\,\|d_j\|_2, \tag{28}$$

with decoder direction

$$\widetilde{d}_j = \frac{s_j d_j}{\|d_j\|_2}. \tag{29}$$

This representation uses the positive half-wave of the skewness-oriented ICA component as the feature passed to SAEBench.

ICA and SAE use the same reconstruction-error-preserving intervention:

$$\tilde{x} = \text{decode}\big(\text{ablate}(\text{encode}(x))\big) + x - \text{decode}(\text{encode}(x)). \tag{30}$$

Ablation sets the selected feature codes to zero while adding back the original representation's reconstruction error. For ICA, this removes the selected oriented positive half-wave; for SAE, it zeroes the selected checkpoint latents.

Figures 38–40 separate the aggregate TPP score into its intended and unintended effects for both evaluation datasets. The same pattern appears across GPT-2 Small, Gemma 2 2B, and Qwen 3.5 9B Base. ICA obtains higher TPP scores than SAE across the useful sparse-ablation range on both Bias in Bios and Amazon Reviews.

The decomposition shows why. Ablating ICA features produces a substantially larger drop in the target probe than ablating the corresponding number of SAE features. Unintended effects also increase with larger ablation budgets, but remain much smaller than the intended effect. The resulting separation between target and non-target behavior is therefore consistently larger for ICA across models and datasets.

These results complement sparse probing. The same non-Gaussian coordinates that concentrate enough concept information to support accurate sparse probes can also be selected and removed in a way that disproportionately affects the intended concept.

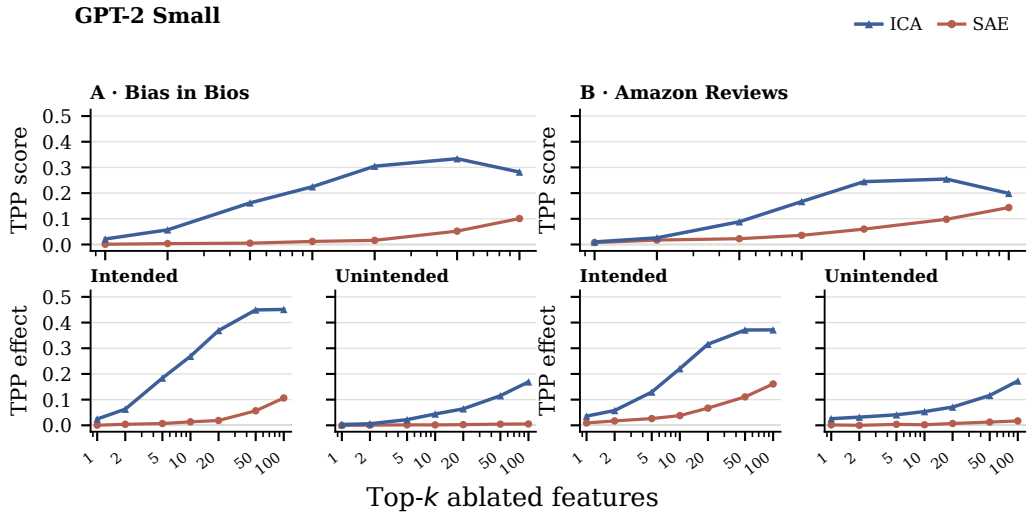


Figure 38: **Dataset-level targeted probe perturbation for GPT-2 Small.** Results are averaged equally over layers 6 and 10. ICA produces a substantially larger intended effect while unintended effects remain comparatively small, yielding higher TPP scores on both datasets.

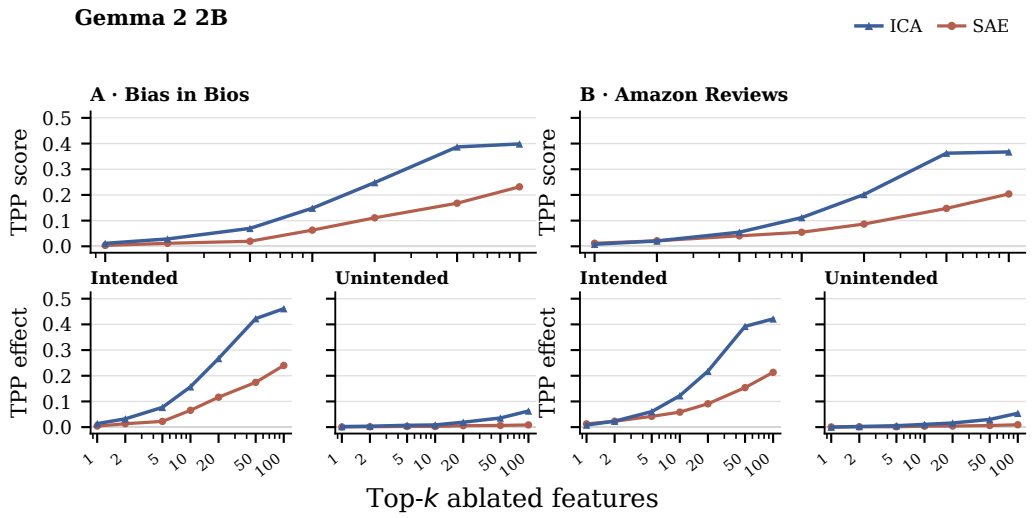


Figure 39: **Dataset-level targeted probe perturbation for Gemma 2 2B.** Results are averaged equally over layers 12 and 20. ICA again produces a larger target-probe effect than SAE, yielding a consistent advantage in total TPP score.

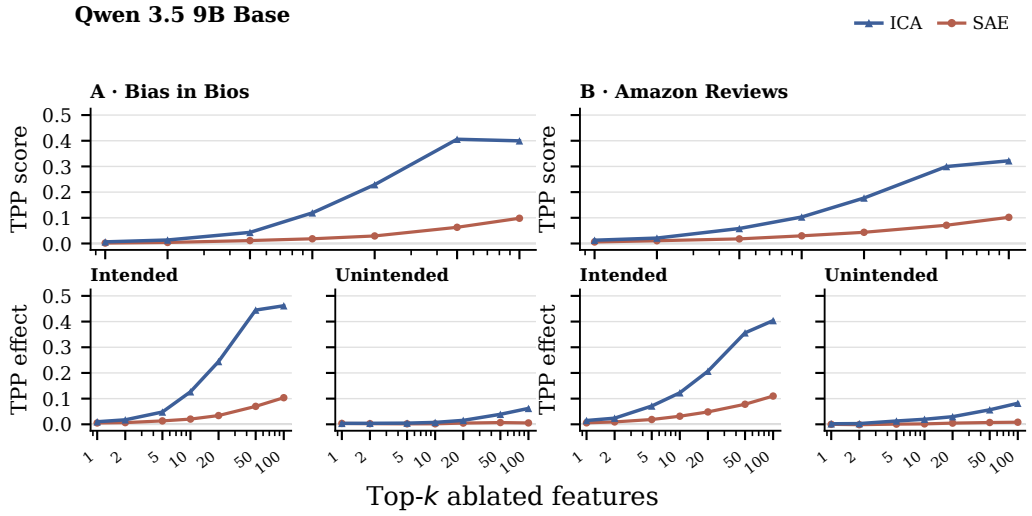


Figure 40: **Dataset-level targeted probe perturbation for Qwen 3.5 9B Base.** Results are averaged equally over layers 12 and 20. The same selectivity pattern persists at larger model scale: ICA produces substantially larger intended effects while unintended effects remain comparatively small.

F SAE Baselines and Checkpoints

We compare ICA against public SAE releases matched to the same models and post-block residual-stream locations used in our experiments. Whenever an evaluation samples individual SAE features, we exclude dead features before sampling and draw only from features that are active under that evaluation protocol.

The exact checkpoints are:

- **GPT-2 Small:** `jbloom/GPT2-Small-OAI-v5-32k-resid-post-SAEs`, revision `21df74db7b9f690cd2403a20485cf75f46330d0d`.

Table 10: **Public SAE baselines used throughout our comparisons.**

Model	SAE release	Width	Native activation
GPT-2 Small	OAI v5 32k	32,768	TopK-32
Gemma 2 2B	Gemma Scope residual 16k	16,384	JumpReLU
Qwen 3.5 9B Base	Qwen Scope residual 64k	65,536	TopK-50

- **Gemma 2 2B:** `google/gemma-scope-2b-pt-res`, revision `fd571b47c1c64851e9b1989792367b9babb4af63`. We use the released width-16k residual SAEs at the registry's layer-specific average L_0 operating points.
- **Qwen 3.5 9B Base:** `Qwen/SAE-Res-Qwen3.5-9B-Base-W64K-L0_50`, revision `7bb2370d360e566b2d8acb8b3d15a0b2b88f52a8`.