SPARSE AUTOENCODERS VS. TRANSCODERS: ENCODER DIRECTION USE AND SINGLE-FEATURE DISCRIMINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) reconstruct MLP outputs, whereas transcoders (TCs) predict them from the corresponding MLP inputs. We compare their single-feature discrimination and examine how encoder direction use and JumpReLU contribute to the difference. Across all 26 layers of Gemma 3 1B, we construct shared MLP-output clusters, select each coder's best feature by postactivation AUROC, and evaluate on held-out documents. TC achieves higher postactivation AUROC in 24 of 26 layers under both sparsity settings, despite SAE having higher mean preactivation AUROC. JumpReLU maps distinct preactivation values to zero, producing a larger discrimination loss for SAE. In an exploratory intervention, reducing TC's reliance on input principal components using an SAE-derived reference narrows its postactivation advantage more than matched random rotations, identifying encoder weighting of high-variance input directions as a partial contributor. Separately, TC retrieves approximately three and four additional target sentences per 200 reviewed under the lower- and higher-activity settings, respectively. Together, these results link single-feature discrimination to both encoder direction use and learned activation thresholds.

1 INTRODUCTION

Sparse autoencoders (SAEs; Cunningham et al., 2023) decompose Transformer activations into sparse features that can be inspected and intervened on (Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2024). Transcoders (TCs; Dunefsky et al., 2024) approximate Transformer multilayer perceptron (MLP) sublayers with sparse surrogate models. An SAE trained on MLP outputs reconstructs y from itself, whereas a TC predicts y from the corresponding MLP input x. Both decode into the same output space but encode different inputs. Recent work on Pythia models (Biderman et al., 2023) reported higher description-based automated interpretability scores for TC features (Paulo et al., 2025). We investigate a complementary question: how do SAE and TC differ in single-feature discrimination, and how does encoder direction use contribute to that difference?

We compare Gemma Scope 2 checkpoints (McDougall et al., 2025) across all 26 layers of Gemma 3 1B (Gemma Team et al., 2025) under two target-activity settings. We construct 32 shared MLP-output direction clusters per layer using spherical k-means (Hornik et al., 2012). For each cluster and coder, we select the feature with the highest postactivation AUROC (Fawcett, 2006) on selection documents and evaluate it on held-out documents. These clusters provide common output-space targets without assuming predefined semantic labels or causal functions.

TC achieves higher postactivation AUROC in 24 of 26 layers under both settings, despite the selected SAE features having higher mean preactivation AUROC. Comparing the same features before and after JumpReLU (Rajamanoharan et al., 2024) shows that SAE loses more discrimination when distinct preactivation values are mapped to zero. Restoring suppressed positive values recovers part of this loss. However, matching activation rates can reverse the postactivation comparison, indicating that the observed advantage depends jointly on preactivation distributions and learned thresholds.

We then test the encoder's role by changing its weighting of input directions while keeping learned thresholds fixed. Altering the contribution of directions that explain MLP-output variation harms TC discrimination more than matched random rotations. Comparable effects for leading input principal

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.

components show that this sensitivity is not unique to output-explanatory directions. An exploratory intervention that limits TC's input-PC reliance using an SAE-based reference reduces its postactivation advantage more than matched random rotations, particularly in early and middle layers. Encoder direction use thus provides a partial explanation of the advantage, while much of the later-layer difference remains unexplained.

Our contributions are threefold. First, we establish TC's single-feature discrimination advantage on shared MLP-output groups and characterize the discrimination lost through JumpReLU. Second, encoder interventions identify a functional contribution of direction use to this advantage. Third, a separate labeled-sentence retrieval evaluation demonstrates a modest practical benefit: TC retrieves approximately three and four additional target sentences per 200 reviewed under the lower- and higher-activity settings, respectively.

2 Related Works

2.1 Sparse Autoencoders and Transcoders

SAEs reconstruct activations through sparse features (Cunningham et al., 2023), whereas TCs predict MLP outputs from MLP inputs and support feature-level circuit analysis (Dunefsky et al., 2024). For SAEs trained on MLP outputs and TCs, the decoding space is shared, but the encoder inputs differ. Prior work comparing SAEs and TCs using TopK found higher description-based automated interpretability scores for TC features (Paulo et al., 2025).

We instead compare single-feature discrimination of shared MLP-output groups using the JumpReLU SAE and TC checkpoints released in Gemma Scope 2 (McDougall et al., 2025). Our focus is how the selected SAE and TC features preserve discrimination through sparse activation.

2.2 Activation Functions and Sparsity

Activation functions and sparsity constraints determine which values remain as feature activations. TopK directly limits the number of active features (Gao et al., 2024), whereas JumpReLU uses learned feature-specific thresholds to improve the reconstruction–sparsity trade-off (Rajamanoharan et al., 2024). Activation functions can also impose geometric constraints that affect which concepts can be extracted (Hindupur et al., 2025). From this perspective, we examine how JumpReLU preserves or removes discrimination among feature values.

Both SAEs and TCs in our study use JumpReLU. We therefore examine how their learned preactivation distributions and thresholds jointly affect discrimination, rather than comparing activation functions. This distinguishes discrimination available before activation from that retained in the resulting sparse features.

2.3 Evaluating Single-Feature Discrimination

Prior work evaluates label prediction using individual latent preactivations (Gao et al., 2024) or small sets of SAE features (Kantamneni et al., 2025). SAEBench combines multiple evaluations of sparse-feature quality (Karvonen et al., 2025), while other work assesses interpretability through natural-language descriptions (Paulo et al., 2024) or description-free methods (Paulo & Belrose, 2025).

We measure how well a single feature discriminates shared MLP-output clusters using postactivation AUROC without fitting a probe. This evaluates whether an output-group distinction can be read from one sparse feature, rather than establishing semantic interpretability or causal relevance.

2.4 Encoder Representations under Reconstruction and Prediction

Analyses of deep linear networks show how input–output cross-statistics shape learned representation directions (Saxe et al., 2019). Reconstruction objectives can favor high-variance directions that differ from those useful for discrimination (Balestriero & Lecun, 2024).

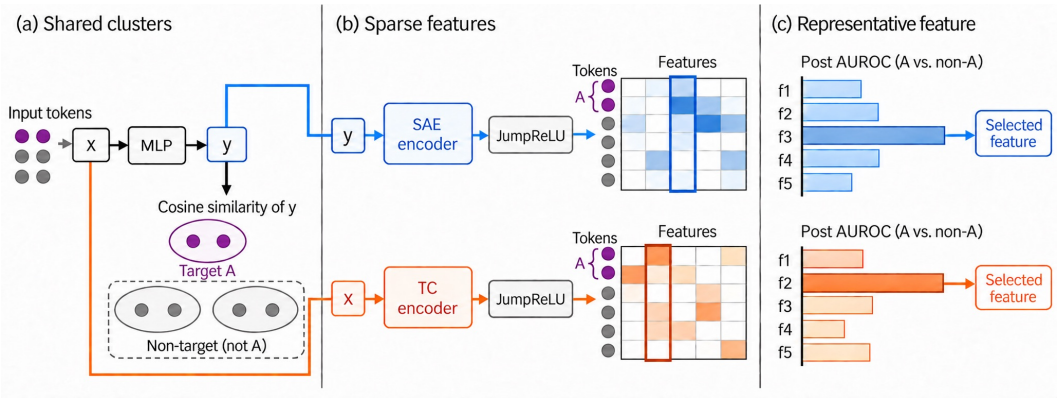

Figure 1: Representative feature selection. (a) Tokens are grouped by cosine similarity of their MLP outputs y. (b) SAE and TC features are computed from MLP outputs y and inputs x, respectively, and passed through JumpReLU. (c) For each target cluster, we select each coder's feature with the highest postactivation AUROC. Token counts and activations are schematic.

Amnesic probing tests whether encoded information supports task performance by removing selected directions and comparing against random-direction removal (Elazar et al., 2021). However, subspace interventions can affect model behavior through pathways not used in the original computation (Makelov et al., 2024). We modify selected SAE and TC features' encoder weights and compare against matched random rotations to test how their use of output-explanatory directions contributes to single-feature discrimination.

3 Methods

3.1 Models and Data

We use frozen Gemma Scope 2 SAE and TC checkpoints (McDougall et al., 2025) for all 26 layers of pretrained Gemma 3 1B (Gemma Team et al., 2025), each with 16,384 features and no skip connections. SAE reconstructs MLP output y; TC predicts y from MLP input x. Following checkpoint specifications, x and y are extracted after pre-feedforward and post-feedforward normalization, respectively. The `l0_small` and `l0_big` settings have lower and higher target numbers of active features, respectively, at the same dictionary size.

Output-cluster analysis uses disjoint FineWeb (Penedo et al., 2024) sets for cluster construction/feature selection and evaluation, each containing 2,228 documents with one 256-token sequence per document (570,368 tokens). Both coders use identical token positions. The interventions in Section 4.4 are evaluated on an additional, previously unused FineWeb set of the same size, reused across those analyses. Entity detection and sentence retrieval use Few-NERD (Ding et al., 2021), deduplicated across its official train, validation, and test splits.

3.2 Feature Selection and Task Evaluation

For feature j, JumpReLU produces

$$z_j = \begin{cases} a_j, & a_j > \theta_j, \\ 0, & a_j \leq \theta_j, \end{cases}$$

where a_j, θ_j, and z_j denote preactivation, learned threshold, and postactivation. For each target, layer, coder, and sparsity setting, we select the highest-postactivation-AUROC feature among all 16,384 candidates and fix its identity and learned threshold for held-out evaluation. This threshold differs from the evaluation cutoff varied for AUROC (Fawcett, 2006) or selected by validation F1.

Output-cluster discrimination. We fit 32 spherical k-means clusters per layer to mean-centered, unit-normalized selection-set MLP outputs (Hornik et al., 2012), without coder activations or semantic labels. The mean and centroids are frozen for evaluation; each token is assigned by maximum cosine similarity. Each cluster defines a one-versus-rest task for feature selection (Figure 1) and held-out evaluation. AUROC includes zero activations and is averaged equally across clusters and layers.

Entity-type detection and sentence retrieval. For each of Few-NERD's 66 fine-grained entity types, a sentence is positive if it contains that type. Its score is the maximum feature activation over nonspecial, nonpadding tokens, without annotated entity positions. Training postactivation AUROC selects features, which are fixed for preactivation and postactivation evaluation with separate validation-F1-maximizing cutoffs. Test F1 is averaged equally across types and layers.

Using the same features, each coder retrieves the top 200 test sentences per type, retaining the 59 types with more than 200 positives. Ranking uses sentence scores, with ties broken by sentence ID without target labels. Retrieved target counts are averaged across types and layers; the review budget excludes feature-selection labels. We obtain 95% confidence intervals from 1,000 paired sentence-bootstrap resamples, repeating retrieval with eligible types and features fixed.

3.3 Thresholding and Activation Restoration

For fixed postactivation-selected features, consecutive AUROC differences across preactivation, ReLU, and JumpReLU quantify losses from zeroing nonpositive values and suppressing positive values at or below the learned threshold.

We restore suppressed positive activations to a_j where $0 < a_j \leq \theta_j$, adding $a_j d_j$ to the decoder output (d_j: selected feature's decoder vector), with all other features and parameters fixed. We measure changes in AUROC and squared Euclidean error against MLP output y. Error is summed over output dimensions and averaged separately over all target and nontarget tokens, including unaffected tokens, then equally across clusters and layers.

Comparing restoration gains across `small` and `big` assesses association with sparsity, not its isolated effect, because learned features and parameters also differ. A separate control adjusts fixed-feature thresholds to match target activation rates (Appendix B.3). Restoration intervals use 1,000 document-bootstrap resamples; the original AUROC decomposition uses document-level jackknife resampling (Efron, 1982), both paired across coders.

3.4 Encoder Direction Interventions

We investigate whether encoder use of particular input directions contributes to TC's higher postactivation discrimination and what explains differences in intervention sensitivity between SAE and TC. Using selection data, we center and whiten each coder's inputs—y for SAE and x for TC (Kessy et al., 2018)—and apply SVD to their cross-covariance with centered MLP outputs to identify input directions that linearly explain output variation (Tippett et al., 2008). Separately, we obtain input principal components (PCs) from the input covariance alone, without using input–output relationships; these identify directions of greatest input variation across tokens.

To test whether encoder direction use contributes to TC's discrimination advantage, we rescale within-subspace and orthogonal weight components while preserving selection-set preactivation mean and total variance (Figure 2). We first adjust the output-explanatory contribution toward a coder-specific random-weight reference. We then compare matched-angle decreases in output-explanatory and input-PC contributions to test whether the effect is specific to output-

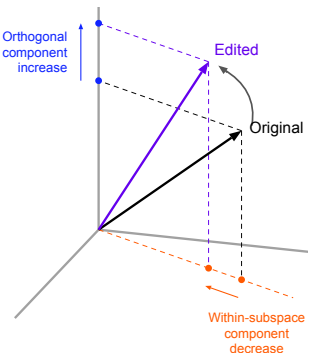

Figure 2: Norm-preserving encoder rotation in whitened coordinates decreases the within-subspace component and increases the orthogonal component.

explanatory directions. In a separate intervention, we limit TC's relative input-PC reliance using an SAE-derived reference. We compare changes in the TC–SAE postactivation AUROC gap against random rotations matched in angle and weight-vector norm in whitened coordinates.

To examine why the resulting discrimination losses differ between SAE and TC, we separately and jointly reduce or remove weight components along the leading input PCs and amplify the remaining components. This distinguishes discrimination retained by the remaining components alone from discrimination recovered by amplifying them. Throughout, feature identities and learned JumpReLU thresholds remain fixed, and AUROC is evaluated on held-out documents. Detailed intervention procedures and controls are provided in Appendix A.1.

Table 1: Single-feature discrimination before and after JumpReLU. The same postactivation-selected features are evaluated on held-out documents. Reduction denotes preactivation minus postactivation AUROC. Values are averaged equally across 32 clusters per layer and then across all 26 layers.

Setting	Coder	Pre AUROC	Post AUROC	Reduction
small	SAE	0.83393	0.66942	0.16451
	TC	0.82091	0.69447	0.12644
big	SAE	0.83540	0.67944	0.15596
	TC	0.81982	0.70895	0.11087

4 Results

4.1 TC Features Better Discriminate Shared Output Clusters

The selected TC features achieve higher held-out postactivation AUROC than SAE features in 24 of 26 layers under both `small` and `big` (Table 1). Yet the same selected SAE features have higher mean preactivation AUROC. Their AUROC reduction through JumpReLU is larger in 25 of 26 layers under `small` and all 26 layers under `big`. In each setting, 10 layers switch from an SAE advantage before JumpReLU to a TC advantage afterward (Figure 3; Appendix B.1). The TC advantage persists across three clustering seeds in both settings (Appendix B.2).

This advantage depends on the operating thresholds: after matching activation rates, SAE has higher mean postactivation AUROC at the tested rates from 0.1% to 10%, using rate-specific eligible subsets (Appendix B.3). The observed TC advantage therefore reflects its learned preactivation distribution and threshold jointly, rather than an advantage independent of activation rate.

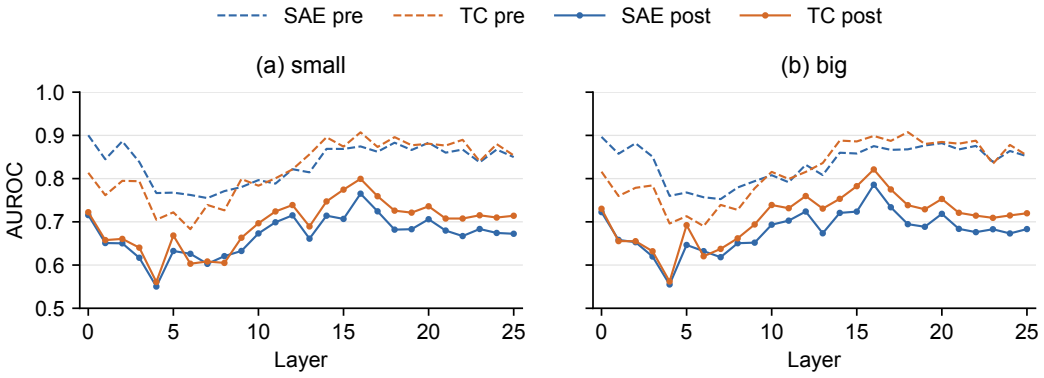

Figure 3: Layer-wise AUROC before and after JumpReLU under (a) small and (b) big. The same postactivation-selected SAE and TC features are evaluated before and after JumpReLU on held-out documents. Each point averages 32 output clusters. Dashed and solid lines denote preactivation and postactivation AUROC, respectively.

Table 2: AUROC reduction from zeroing nonpositive preactivations (Pre–ReLU) and suppressing positive values up to the learned threshold (ReLU–Post). Reduction is the sum of these two components. Results are averaged equally across 32 clusters per layer and then across all 26 layers. Values are rounded independently.

Setting	Coder	Pre–ReLU	ReLU–Post	Reduction
small	SAE	0.11959	0.04492	0.16451
	TC	0.10487	0.02157	0.12644
big	SAE	0.13636	0.01959	0.15596
	TC	0.09953	0.01134	0.11087

4.2 Discrimination Lost Through Thresholding

We decompose the AUROC change into two steps: mapping nonpositive preactivations to zero, then suppressing positive values up to the learned threshold (Table 2). Mapping distinct target and nontarget values to zero turns both correct and incorrect comparisons into ties; the net change depends on the balance between them.

SAE loses more AUROC at the first step in 18 of 26 layers under `small` and 24 of 26 under `big`, and at the second step in all 26 layers under both settings. Its larger discrimination loss therefore extends beyond positive values suppressed by the learned threshold. Restoring suppressed positive values nearly closes the TC–SAE AUROC gap under `small` (0.02505 to 0.00170), but leaves a substantial gap under `big` (0.02951 to 0.02125). Under `big`, this remaining gap is driven primarily by SAE's larger discrimination loss from zeroing nonpositive values (Appendix B.4).

Under `small`, restoring suppressed positive values reduces the number of layers with a TC advantage from 24 to 15 and lowers output error for both target and nontarget tokens in most layers (Appendix B.4). However, TC layer-0 error increases by approximately 33% for target tokens and 61% for nontarget tokens, showing that restoration does not uniformly improve reconstruction. Moreover, restoration increases feature activity, so these gains do not establish an improvement in the sparsity-regularized training objective.

4.3 Encoder Direction Use Supports Discrimination

We first compare how much the leading 16 output-explanatory directions contribute to the selected features' preactivation variance, using random encoder weights on the same inputs as a reference. Under `small`, these directions account for 60.72% of SAE preactivation variance, compared with 51.23% for random weights. For TC, the corresponding fractions are 53.19% and 16.81%. The same pattern holds under `big`. Thus, although the absolute fraction is larger for SAE, the selected TC encoder weights place substantially more preactivation variance in these directions than their own random-weight reference does (Table 3(a)).

To test whether this direction use supports discrimination, we rotate each selected encoder weight vector so that the variance fraction moves toward its coder-specific reference. Rotation angles are matched between SAE and TC in whitened coordinates, and selection-set preactivation means and variances are preserved. We evaluate the edited features on held-out documents using their original learned thresholds. Compared with angle- and norm-matched random rotations, the targeted adjustment produces an additional postactivation AUROC reduction of 0.01189 for TC versus 0.00118 for SAE under `small`, and 0.01530 versus 0.00092 under `big` (Table 3(b)). This additional reduction is larger for TC in 23 of 26 layers under `small` and 24 of 26 under `big`.

The selected TC features' postactivation discrimination is therefore more sensitive to this targeted change in direction use than SAE's, relative to matched random rotations. TC nevertheless retains higher mean postactivation AUROC after adjustment. These results establish a functional role for encoder direction use but do not fully explain TC's original advantage or its smaller discrimination loss through JumpReLU. Results across subspace sizes are provided in Appendix B.5.

Table 3: Leading-16 output-explanatory directions. (a) Selection-set preactivation variance fractions for learned and random encoder weights. (b) Held-out postactivation AUROC before intervention (Original), after rotation toward the random-weight reference (Targeted), and after angle- and norm-matched random rotations (Random). Additional reduction is Random minus Targeted; brackets give 95% paired document-bootstrap intervals.

(a) Preactivation variance fraction (%)

Setting	Coder	Learned weights	Random weights
small	SAE	60.72	51.23
	TC	53.19	16.81
big	SAE	60.11	51.23
	TC	52.86	16.81

(b) Postactivation AUROC

Setting	Coder	Original	Random	Targeted	Additional reduction [95% CI]
small	SAE	0.66942	0.66397	0.66279	0.00118 [0.00110, 0.00126]
	TC	0.69447	0.68647	0.67458	0.01189 [0.01177, 0.01201]
big	SAE	0.67944	0.67377	0.67286	0.00092 [0.00083, 0.00101]
	TC	0.70895	0.70070	0.68541	0.01530 [0.01518, 0.01542]

4.4 Encoder Direction Use and Discrimination

TC's greater sensitivity is not specific to output-explanatory directions. At matched rotation angles, reducing the contribution of either output-explanatory directions or input PCs causes larger additional postactivation AUROC losses for TC than SAE relative to random controls (Table 4(a); Appendix B.6). For the leading 16 input PCs, the additional loss was 0.00395 for TC versus 0.00048 for SAE under `small`, and 0.00425 versus 0.00079 under `big`. Thus, TC discrimination also depends on how its encoder weights input directions that vary most across tokens.

TC's use of input PCs also contributes to its original postactivation discrimination advantage. In an exploratory follow-up using the same evaluation documents, limiting TC's relative input-PC use to the SAE-derived reference narrowed the mean TC–SAE AUROC gap. For the leading 16 PCs, the gap fell from 0.02495 to 0.00725 under `small` and from 0.02932 to 0.00707 under `big`. Matched random rotations left larger gaps of 0.01731 and 0.02091, respectively (Table 4(b)). Limiting TC's input-PC use therefore reduced its advantage more than matched random changes, although it did not eliminate the gap.

Table 4: Leading-16-direction interventions, averaged equally across 32 clusters and 26 layers. (a) Additional postactivation AUROC loss: matched-random minus targeted AUROC. (b) TC–SAE postactivation AUROC gaps under SAE-referenced TC adjustment, with SAE unchanged. Random denotes angle- and norm-matched rotations.

(a) Additional postactivation AUROC loss

Setting	Coder	Output-explanatory	Input PCs
small	SAE	0.00048	0.00048
	TC	0.00335	0.00395
big	SAE	0.00079	0.00079
	TC	0.00391	0.00425

(b) TC–SAE postactivation AUROC gap

Setting	Original	Random	SAE-referenced adjustment
small	0.02495	0.01731	0.00725
big	0.02932	0.02091	0.00707

Greater recovery from amplifying the remaining weight components partly explains SAE's smaller loss under variance-preserving interventions. Completely removing the leading 16 input-PC components substantially reduced discrimination in both coders. Amplifying the remainder recovered some discrimination, with a larger gain for SAE, but did not restore either coder's original performance (Table 5(a)). After partial reduction, amplifying the remaining components yields greater mean AUROC recovery for SAE than TC across all five tested subspace sizes ($k = 1, 4, 16, 64, 128$) and both sparsity settings (Table 5(b); Appendix B.8).

Table 5: Postactivation AUROC for leading-16-input-PC interventions, averaged equally across 32 clusters and 26 layers. (a) Complete removal, with the remainder unchanged or amplified to restore selection-set preactivation variance. (b) AUROC gain from amplifying the remainder relative to reducing the leading components alone. Brackets give 95% paired document-bootstrap confidence intervals.

(a) Complete removal of the leading components

Setting	Coder	Original	Remainder only	Remainder amplified
small	SAE	0.66918	0.56143	0.60317
	TC	0.69413	0.55922	0.59724
big	SAE	0.67930	0.56314	0.60560
	TC	0.70862	0.56296	0.59824

(b) Recovery after partial reduction of the leading components

Setting	Coder	AUROC recovery [95% CI]
small	SAE	0.00589 [0.00584, 0.00594]
	TC	0.00438 [0.00433, 0.00442]
big	SAE	0.00522 [0.00518, 0.00527]
	TC	0.00316 [0.00312, 0.00320]

4.5 Single-Feature Entity Detection and Sentence Retrieval

On Few-NERD, TC achieves higher mean test F1 than SAE under both sparsity settings (Table 6), with higher layer-wise F1 in 23/26 layers under `small` and 21/26 under `big`. However, preactivation and postactivation F1 are nearly identical (Appendix B.9). This provides complementary evidence of single-feature detection utility, rather than showing that the JumpReLU effects observed for output clusters explain the entity-detection advantage.

When retrieving 200 sentences per entity type, TC returns more target-containing sentences on average, with gains in 19/26 layers under `small` and 16/26 under `big`. The benefit is concentrated in early layers; across layers 13–25, the mean gain drops to approximately 0.23 and 1.41 sentences, respectively. Thus, TC provides a modest average benefit after supervised feature selection rather than a uniform layer-wise advantage, excluding the labeled-data cost of feature selection.

Table 6: Few-NERD test results averaged equally across entity types and 26 layers: detection over 66 types and retrieval over 59 eligible types. Δ denotes TC minus SAE; brackets give 95% paired sentence-bootstrap confidence intervals from 1,000 resamples.

Metric	Setting	SAE	TC	Δ [95% CI]
Postactivation F1	small	0.36770	0.38749	+0.01979 [0.01896, 0.02067]
	big	0.38349	0.40600	+0.02252 [0.02166, 0.02339]
Targets retrieved / 200	small	81.92	84.67	+2.75 [2.29, 3.03]
	big	87.28	91.63	+4.35 [3.92, 4.68]

5 Discussion

TC's higher postactivation discrimination does not arise from better discrimination before thresholding. The selected SAE features have higher mean preactivation AUROC but lose more discrimination through JumpReLU. Restoring suppressed positive values nearly closes the gap under `small`,

whereas a substantial gap remains under `big`, where SAE loses more from mapping nonpositive values to zero. Positive-value suppression alone therefore cannot explain both settings. Matching activation rates can also reverse the comparison, indicating that TC's observed advantage depends jointly on the learned preactivation distributions and thresholds.

TC uses input directions that vary substantially across tokens to distinguish target from nontarget tokens after JumpReLU. Limiting the contribution of these directions using the SAE-derived reference reduces TC's advantage more than matched random rotations. The matched-angle decrease experiments also show that TC's greater sensitivity is not specific to output-explanatory directions. SAE, by contrast, recovers more discrimination when the remaining encoder-weight components are amplified after reducing the leading input-PC components, partly explaining why variance-preserving interventions harm SAE less. However, complete removal still substantially harms both coders, and the SAE-referenced adjustment leaves much of TC's later-layer advantage unexplained (Appendix B.7).

Better discrimination does not necessarily imply a better training objective. Restoring suppressed positive values can improve discrimination and output reconstruction, but also increases feature activity. Separately, TC retrieves more target sentences on average, although the benefit is modest and concentrated in earlier layers. Because entity-detection F1 changes little before and after thresholding, we do not attribute this practical benefit to the JumpReLU effects observed for output clusters.

Our findings concern task-selected features in the released Gemma Scope 2 checkpoints for Gemma 3 1B. The clusters group tokens by MLP-output direction, not established semantic functions. The SAE-referenced input-PC adjustment is exploratory and reuses evaluation documents from the preceding analysis. These findings do not establish broader TC superiority in interpretability.

6 Conclusion

The selected TC features preserve more discrimination through JumpReLU than SAE features. TC's use of input principal components contributes to its higher postactivation discrimination. After reducing the leading input-PC weight components, amplifying the remaining components restores more discrimination for SAE, partly explaining its smaller intervention losses. These findings provide a partial encoder-level explanation of the SAE–TC difference. Separately, TC offers a modest average benefit in target-sentence retrieval.

Reproducibility Statement

We use publicly released Gemma Scope 2 checkpoints and the FineWeb and Few-NERD datasets without retraining the coders. Section 3 describes the model settings, data splits, output clustering, feature selection, evaluation metrics, and activation restoration procedure. Feature selection and evaluation use separate documents, and the selected feature identities remain fixed in preactivation–postactivation comparisons. The appendix provides layer-wise results and additional analyses supporting the aggregate results.

AI Use Statement

The authors used generative AI tools to draft portions of the manuscript and polish the writing. The authors reviewed and revised the AI-assisted text and directly refined the organization and wording of each section to clearly communicate the paper's arguments and results.

At each stage of the research, the authors formulated the research questions and hypotheses and critically evaluated and selected AI-proposed methods for testing them, making the final decisions on experimental design and analysis procedures. AI tools also assisted with conducting experiments and generating code to implement the selected methods. The authors performed the initial interpretation of the experimental results, while AI tools assisted in assessing the validity of those interpretations. The authors determined the final interpretations and conclusions based on the results and supporting evidence.

Generative AI tools were also used for literature search and discovery. The authors read the sources used in the manuscript and made the final decisions regarding citations.

The authors take responsibility for the entirety of this work, including the final manuscript, research design, methods, statistical analyses, code, results, interpretations, citations, and conclusions.

REFERENCES

Randall Balestriero and Yann Lecun. How learning by reconstruction produces uninformative features for perception. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st international conference on machine learning*, volume 235 of *Proceedings of machine learning research*, pp. 2566–2585. PMLR, July 2024. URL https://proceedings.mlr.press/v235/balestriero24b.html.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: a suite for analyzing large language models across training and scaling, 2023. URL https://arxiv.org/abs/2304.01373. arXiv: 2304.01373 [cs.CL].

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.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models, September 2023. URL https://arxiv.org/abs/2309.08600v3.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. Few-NERD: A Few-Shot Named Entity Recognition Dataset, September 2021. URL http://arxiv.org/abs/2105.07464. arXiv:2105.07464 [cs.CL].

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders Find Interpretable LLM Feature Circuits, November 2024. URL http://arxiv.org/abs/2406.11944. arXiv:2406.11944 [cs.LG].

Bradley Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, January 1982. ISBN 978-0-89871-179-0 978-1-61197-031-9. doi: 10.1137/1.9781611970319. URL https://epubs.siam.org/doi/book/10.1137/1.9781611970319.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.10.010. URL https://www.sciencedirect.com/science/article/pii/S016786550500303X.

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, June 2024. URL http://arxiv.org/abs/2406.04093. arXiv:2406.04093 [cs.LG].

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and others. Gemma 3 technical report, March 2025.

Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba Ba. Projecting Assumptions: The Duality Between Sparse Autoencoders and Concept Geometry, March 2025. URL https://arxiv.org/abs/2503.01822v2.

Kurt Hornik, Ingo Feinerer, Martin Kober, and Christian Buchta. Spherical k-Means clustering. *Journal of Statistical Software*, 50(10):1–22, 2012. doi: 10.18637/jss.v050.i10. URL https://www.jstatsoft.org/article/view/v050i10.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are Sparse Autoencoders Useful? A Case Study in Sparse Probing, 2025. URL https://arxiv.org/abs/2502.16681. Version Number: 1.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A Comprehensive Benchmark for Sparse Autoencoders in Language Model Interpretability, June 2025. URL http://arxiv.org/abs/2503.09532. arXiv:2503.09532 [cs.LG].

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, October 2018. ISSN 0003-1305, 1537-2731. doi: 10.1080/00031305.2016.1277159. URL http://arxiv.org/abs/1512.00809. arXiv:1512.00809 [stat.ME].

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is This the Subspace You Are Looking for? An Interpretability Illusion for Subspace Activation Patching. *International Conference on Learning Representations*, 2024:26486–26515, May 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/70b8505ac79e3e131756f793cd80eb8d-Abstract-Conference.html.

Callum McDougall, Arthur Conmy, János Kramár, Tom Lieberum, Senthooran Rajamanoharan, and Neel Nanda. Gemma scope 2: Technical paper. Technical Report, Google DeepMind, September 2025. URL https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/gemma-scope-2-helping-the-ai-safety-community-deepen-understanding-of-complex-language-model-behavior/Gemma_Scope_2_Technical_Paper.pdf.

Gonçalo Paulo and Nora Belrose. Evaluating SAE interpretability without explanations, July 2025. URL http://arxiv.org/abs/2507.08473. arXiv:2507.08473 [cs.LG].

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically Interpreting Millions of Features in Large Language Models, October 2024. URL https://arxiv.org/abs/2410.13928v3.

Gonçalo Paulo, Stepan Shabalin, and Nora Belrose. Transcoders beat sparse autoencoders for interpretability, 2025. URL https://arxiv.org/abs/2501.18823. arXiv: 2501.18823 [cs.LG].

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale, June 2024. URL https://arxiv.org/abs/2406.17557v2.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders, August 2024. URL http://arxiv.org/abs/2407.14435. arXiv:2407.14435 [cs.LG].

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116 (23):11537–11546, 2019. doi: 10.1073/pnas.1820226116. URL https://www.pnas.org/doi/full/10.1073/pnas.1820226116.

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.

Michael K. Tippett, Timothy DelSole, Simon J. Mason, and Anthony G. Barnston. Regression-Based Methods for Finding Coupled Patterns. *Journal of Climate*, 21(17):4384–4398, September 2008. ISSN 0894-8755, 1520-0442. doi: 10.1175/2008JCLI2150.1. URL https://journals.ametsoc.org/view/journals/clim/21/17/2008jcli2150.1.xml.

A Additional Methods

A.1 Encoder Direction Interventions

Output-explanatory directions. Let u denote the coder input (y for SAE and x for TC), with selection-set means μ_u and μ_y. From the selection-set input covariance, we retain eigenvalues at least 10^{-6} times the largest eigenvalue. Let Q contain the corresponding eigenvectors as columns and Λ be the diagonal matrix of retained eigenvalues. We whiten the inputs (Kessy et al., 2018) and compute their cross-covariance C with centered MLP outputs:

$$\widetilde{u} = \Lambda^{-1/2} Q^\top (u - \mu_u), \qquad C = \mathbb{E}_{\text{sel}}[\widetilde{u}(y - \mu_y)^\top] = USV^\top.$$

Here, U and V contain the input-side and output-side singular vectors, and S contains singular values in descending order. Following Tippett et al. (2008), the first k columns U_k define the output-explanatory subspace, for $k \in \mathcal{K} = \{1, 4, 16, 64, 256\}$.

Output targets are neither whitened nor unit-normalized. Each layer–coder basis is shared across clusters and sparsity settings and remains fixed during evaluation.

Variance-preserving encoder edits. For a selected feature with preactivation $a = w^\top u + b$, its encoder coefficients in whitened coordinates are $\beta = \Lambda^{1/2} Q^\top w$. We separate the components within and orthogonal to the output-explanatory subspace:

$$\beta_\parallel = U_k U_k^\top \beta, \qquad \beta_\perp = \beta - \beta_\parallel, \qquad r_k = \frac{\|\beta_\parallel\|^2}{\|\beta\|^2}.$$

Because the whitened inputs have identity covariance, squared coefficient norms give preactivation variances within the retained space. Thus, r_k is the fraction contributed by the selected subspace.

For $0 < r_k < 1$, let $\eta_k = \arccos\sqrt{r_k}$ be the angle between β and $\beta_\parallel$. Given a new angle $\eta_k' \in [0, \pi/2]$, measured from the same $\beta_\parallel$ direction, we set

$$\beta' = \frac{\cos\eta_k'}{\sqrt{r_k}}\beta_\parallel + \frac{\sin\eta_k'}{\sqrt{1 - r_k}}\beta_\perp.$$

This preserves the two component directions and the total norm, changes the variance fraction to $\cos^2\eta_k'$, and rotates the weight vector by $|\eta_k' - \eta_k|$ in whitened coordinates. We map the edit back to the original coordinates and adjust the bias:

$$\Delta w = Q\Lambda^{-1/2}(\beta' - \beta), \qquad w' = w + \Delta w, \qquad b' = b - \mu_u^\top \Delta w.$$

Weights outside the retained space remain unchanged. The norm constraint preserves selection-set preactivation variance, and the bias adjustment preserves its mean. These constraints are imposed on selection data; held-out moments are not rematched.

1. Rotation toward a random-weight reference. For each layer, coder, and k, we sample 1,024 normalized isotropic Gaussian vectors v_m in the retained covariance-eigenvector coordinates and compute

$$g_k = \frac{1}{1024}\sum_{m=1}^{1024} \frac{\|U_k^\top \Lambda^{1/2} v_m\|^2}{\|\Lambda^{1/2} v_m\|^2}.$$

Each summand is the output-explanatory variance fraction produced by one random weight vector. Their mean defines the reference, shared across clusters and sparsity settings.

For each paired SAE–TC cluster comparison, define the angle required for coder c to reach its reference:

$$d_c = \left|\arccos\sqrt{g_{k,c}} - \eta_{k,c}\right|.$$

We rotate both coders toward their respective references by $t\min(d_{\mathrm{SAE}}, d_{\mathrm{TC}})$, with $t \in \{0, \frac{1}{2}, 1\}$. This matches rotation angles without requiring both coders to reach their references. Depending on the original fraction, the rotation can increase or decrease the subspace contribution.

2. Decreasing output-explanatory contribution. We rotate both coders toward lower output-explanatory variance fractions:

$$\delta_{\max} = \min\left\{\frac{\pi}{18}, \frac{1}{2}\min_{\substack{c\in\{\mathrm{SAE,TC}\}\\ k\in\mathcal{K}}} \arcsin\sqrt{r_{k,c}}\right\}, \qquad \eta'_{k,c} = \eta_{k,c} + t\delta_{\max},$$

where $t \in \{0, \frac{1}{4}, \frac{1}{2}, 1\}$. The angle is computed separately for each layer, cluster, and sparsity setting, but is common across both coders and all five subspace sizes. It is capped at 10° and at half the smallest angle required to reduce a subspace contribution to zero. The intervention therefore matches rotation magnitude, not the amount of variance-fraction reduction.

3. Decreasing input-PC contribution. To test whether the effect is specific to output-explanatory directions, we instead reduce the contribution of input principal components. Let P_k select the whitened coordinates corresponding to the k largest retained input-covariance eigenvalues. Their preactivation variance fraction is

$$p_k = \frac{\|P_k^\top \beta\|^2}{\|\beta\|^2}.$$

Starting from the original weights, we replace U_k with P_k in the encoder edit and use

$$\eta_k = \arccos\sqrt{p_k}, \qquad \eta'_k = \eta_k + \delta_{\max}.$$

The angle is unchanged from the full-strength output-explanatory decrease intervention. A comparison is included only if this angle satisfies $\delta_{\max} \le \frac{1}{2}\arcsin\sqrt{p_k}$ for the TC input-PC edit. All compared conditions use the same eligible layer–cluster pairs. For SAE, the input-PC and output-explanatory subspaces coincide because its input and target are both y.

4. SAE-referenced TC adjustment. We next adjust only TC weights, using the unchanged SAE weights to determine the target input-PC contribution. All quantities below are computed separately for each layer, sparsity setting, and k.

Let $p_{c,j}$ be the input-PC variance fraction for coder c and the feature selected for cluster j. Let g_c^{PC} be its coder-specific random-weight reference, computed using the same 1,024-vector procedure with P_k replacing U_k. We define

$$q_{c,j} = \frac{p_{c,j}/(1-p_{c,j})}{g_c^{\mathrm{PC}}/(1-g_c^{\mathrm{PC}})}.$$

The numerator compares variance contributed by the input PCs with variance contributed by the remaining directions. The denominator makes the same comparison for the random-weight reference, accounting for differences between the two input spaces.

We take the geometric mean across the 32 cluster-selected SAE features and apply it to TC's random-weight reference:

$$\bar{q}_{\mathrm{SAE}} = \left(\prod_{j=1}^{32} q_{\mathrm{SAE},j}\right)^{1/32}, \qquad R^\star = \bar{q}_{\mathrm{SAE}}\frac{g_{\mathrm{TC}}^{\mathrm{PC}}}{1-g_{\mathrm{TC}}^{\mathrm{PC}}}, \qquad p^\star = \frac{R^\star}{1+R^\star}.$$

Here, $R^\star$ is the target PC-to-remainder variance ratio, and $p^\star$ is the corresponding target fraction. Each TC feature is edited to

$$p'_{\mathrm{TC},j} = \min(p_{\mathrm{TC},j}, p^\star), \qquad \eta'_j = \arccos\sqrt{p'_{\mathrm{TC},j}}.$$

Features already below the target remain unchanged. Otherwise, the rotation angle is the amount required to reach the target. SAE weights remain fixed.

5. Separating component removal and amplification. For $k \in \{1, 4, 16, 64, 128\}$, we decompose the original encoder coefficients into leading input-PC and remaining components:

$$\beta_\parallel = P_k P_k^\top \beta, \qquad \beta_\perp = \beta - \beta_\parallel.$$

To measure discrimination retained after complete removal and recovered by amplification, we compare

$$\beta_{\mathrm{removed}} = \beta_\perp, \qquad \beta_{\mathrm{restored}} = \frac{\|\beta\|}{\|\beta_\perp\|}\beta_\perp.$$

The latter restores the original coefficient norm and hence selection-set preactivation variance.

For partial reduction, let $\eta_k = \arccos\sqrt{p_k}$ and define

$$\alpha_k = \frac{\cos(\eta_k + \delta_k)}{\cos\eta_k}, \qquad \gamma_k = \frac{\sin(\eta_k + \delta_k)}{\sin\eta_k}.$$

We use $\delta_k = \delta_{\max}$ for $k = 16$. For the other sizes, we additionally cap the angle at half the smaller angle required to remove the SAE or TC input-PC contribution:

$$\delta_k = \min\left\{\delta_{\max}, \frac{1}{2}\min_{c\in\{\mathrm{SAE},\mathrm{TC}\}}\arcsin\sqrt{p_{k,c}}\right\}.$$

We evaluate reduction alone, amplification alone, and their combination:

$$\begin{aligned}\beta_{\mathrm{reduced}} &= \alpha_k\beta_\parallel + \beta_\perp,\\ \beta_{\mathrm{amplified}} &= \beta_\parallel + \gamma_k\beta_\perp,\\ \beta_{\mathrm{combined}} &= \alpha_k\beta_\parallel + \gamma_k\beta_\perp.\end{aligned}$$

Recovery is postactivation AUROC for $\beta_{\mathrm{restored}}$ minus that for β_{removed} after complete removal, and for $\beta_{\mathrm{combined}}$ minus that for β_{reduced} after partial reduction.

All conditions use the coordinate mapping and bias correction above, preserving the selection-set mean and weights outside the retained space. Total variance is preserved only in the restored and combined conditions. Random controls match each edited vector's angle from the original vector and its edited norm. Evaluation reuses the preceding analysis's held-out documents and fixed features and thresholds.

Matched random controls and evaluation. For an edited coefficient vector β', let ϕ be its angle from the original β. We draw a Gaussian vector, remove its projection onto β, and normalize the remainder to a unit vector h. The random control is

$$\beta_{\mathrm{random}} = \|\beta'\|\left(\cos\phi\,\frac{\beta}{\|\beta\|} + \sin\phi\,h\right),$$

where h is a unit vector orthogonal to β. This matches the edited vector's rotation angle and norm. For variance-preserving edits, $\|\beta'\| = \|\beta\|$. We average results over eight fixed random seeds.

Repeated selections of the same feature reuse its random directions within a coder and checkpoint. Controls match the actual angle of the corresponding edit, including feature-specific angles in the SAE-referenced adjustment.

All edits are determined from selection data and fixed before evaluation. We recompute preactivations on held-out documents and apply the original learned JumpReLU thresholds, keeping feature identities, target labels, and decoder parameters unchanged. Additional postactivation AUROC reduction is the random-control AUROC minus the targeted-edit AUROC. Results are averaged equally across included clusters within each layer and then across all 26 layers. Confidence intervals use 1,000 paired document-bootstrap resamples shared across compared conditions.

B Additional Results

B.1 Layer-Wise Discrimination Before and After JumpReLU

Table 7: Layer-wise AUROC before and after JumpReLU under `small`. The same postactivation-selected features are evaluated on held-out documents, including zero activations. Reduction denotes preactivation minus postactivation AUROC. Each layer averages 32 clusters equally; the mean averages all 26 layers equally.

	Pre AUROC		Post AUROC		Reduction	
Layer	**SAE**	**TC**	**SAE**	**TC**	**SAE**	**TC**
0	0.90027	0.81345	0.71557	0.72226	0.18471	0.09118
1	0.84473	0.76159	0.65096	0.65774	0.19377	0.10386
2	0.88635	0.79485	0.65024	0.66052	0.23611	0.13433
3	0.83833	0.79393	0.61682	0.64027	0.22152	0.15366
4	0.76658	0.70489	0.55033	0.56092	0.21624	0.14397
5	0.76755	0.72203	0.63253	0.66831	0.13502	0.05372
6	0.76216	0.68295	0.62604	0.60311	0.13612	0.07984
7	0.75494	0.73904	0.60279	0.60827	0.15215	0.13077
8	0.77111	0.72660	0.62064	0.60497	0.15048	0.12163
9	0.78040	0.79905	0.63259	0.66319	0.14781	0.13585
10	0.79655	0.78358	0.67317	0.69669	0.12338	0.08689
11	0.78840	0.80075	0.69895	0.72403	0.08946	0.07672
12	0.82179	0.82132	0.71498	0.73881	0.10682	0.08250
13	0.81449	0.85612	0.66105	0.68937	0.15344	0.16675
14	0.86884	0.89565	0.71421	0.74720	0.15464	0.14845
15	0.86866	0.87402	0.70676	0.77448	0.16189	0.09954
16	0.87446	0.90713	0.76516	0.79943	0.10929	0.10770
17	0.86213	0.87315	0.72433	0.75924	0.13780	0.11391
18	0.88312	0.89600	0.68192	0.72586	0.20119	0.17013
19	0.86624	0.87714	0.68295	0.72126	0.18329	0.15588
20	0.88296	0.88070	0.70603	0.73593	0.17693	0.14478
21	0.86005	0.87646	0.67979	0.70766	0.18026	0.16881
22	0.86727	0.88974	0.66707	0.70762	0.20020	0.18212
23	0.83763	0.84015	0.68341	0.71515	0.15422	0.12499
24	0.86724	0.87976	0.67429	0.70994	0.19296	0.16982
25	0.84987	0.85368	0.67238	0.71405	0.17749	0.13963
Mean	**0.83393**	**0.82091**	**0.66942**	**0.69447**	**0.16451**	**0.12644**

Table 8: Layer-wise AUROC before and after JumpReLU under `big`. Feature selection, evaluation, aggregation, and reduction follow Table 7.

	Pre AUROC		Post AUROC		Reduction	
Layer	**SAE**	**TC**	**SAE**	**TC**	**SAE**	**TC**
0	0.89650	0.81595	0.72246	0.73025	0.17404	0.08570
1	0.85744	0.75979	0.65819	0.65539	0.19926	0.10440
2	0.88171	0.77868	0.65287	0.65519	0.22884	0.12349
3	0.85069	0.78427	0.61975	0.63191	0.23094	0.15236
4	0.76001	0.69611	0.55510	0.56201	0.20491	0.13411
5	0.76786	0.71307	0.64647	0.69232	0.12139	0.02075
6	0.75686	0.69007	0.63216	0.62035	0.12470	0.06972
7	0.75258	0.73918	0.61819	0.63742	0.13439	0.10176
8	0.77946	0.72762	0.65053	0.66183	0.12893	0.06579
9	0.79413	0.77755	0.65182	0.69393	0.14232	0.08363
10	0.80791	0.81558	0.69341	0.73893	0.11451	0.07665
11	0.79157	0.79915	0.70273	0.73151	0.08883	0.06764
12	0.83199	0.81548	0.72384	0.75972	0.10815	0.05575
13	0.80818	0.83604	0.67352	0.73062	0.13465	0.10542
14	0.85989	0.88801	0.72064	0.75321	0.13925	0.13480
15	0.85808	0.88600	0.72364	0.78249	0.13445	0.10352
16	0.87513	0.89867	0.78578	0.82122	0.08935	0.07745
17	0.86653	0.88723	0.73374	0.77495	0.13279	0.11228
18	0.86771	0.90753	0.69462	0.73852	0.17309	0.16901
19	0.87668	0.88017	0.68856	0.72900	0.18811	0.15117
20	0.88198	0.88478	0.71849	0.75295	0.16349	0.13183
21	0.86753	0.88082	0.68371	0.72091	0.18382	0.15992
22	0.87568	0.88782	0.67616	0.71431	0.19952	0.17351
23	0.83859	0.83432	0.68284	0.70924	0.15575	0.12508
24	0.86393	0.87796	0.67314	0.71481	0.19080	0.16316
25	0.85185	0.85349	0.68320	0.71980	0.16865	0.13369
Mean	**0.83540**	**0.81982**	**0.67944**	**0.70895**	**0.15596**	**0.11087**

B.2 Robustness to Clustering Seeds

Table 9: Held-out postactivation AUROC across clustering seeds, averaged equally across 32 clusters and 26 layers. Features are reselected for each seed. Δ denotes TC minus SAE. Initialization uses the listed seed plus the layer index.

Setting	Seed	SAE	TC	Δ	TC > SAE (layers)
small	4201	0.66943	0.69474	+0.02531	24/26
	4202	0.66788	0.69484	+0.02696	24/26
	4203	0.66741	0.69491	+0.02750	24/26
big	4201	0.67934	0.70884	+0.02950	24/26
	4202	0.67776	0.70693	+0.02916	24/26
	4203	0.67558	0.70721	+0.03164	25/26

The original seed (4201) is rerun using regenerated activations on the same evaluation documents.

B.3 Discrimination at Matched Activation Rates

With feature identities fixed, we calibrate nonnegative JumpReLU thresholds to the requested activation rates on 2,228 separate documents, then apply them unchanged to evaluation data. Tables 10 and 11 include comparisons with calibration errors within 0.01 percentage points for both coders and evaluation activation-rate differences within 0.1 percentage points. Eligible comparisons vary across rates, with fewer clusters and layers at higher rates.

Table 10: Matched-activation-rate comparison under `small`. N/L gives the number of eligible layer–cluster comparisons and represented layers. Δ denotes TC minus SAE postactivation AUROC. Intervals are pointwise 95% paired document-jackknife confidence intervals, conditional on the fixed features, calibrated thresholds, and eligible comparisons. Values are rounded independently.

Requested active (%)	Observed active (%)		N/L	Post AUROC		Δ [95% CI]
	SAE	TC		SAE	TC	
0.10	0.101	0.100	832/26	0.51324	0.51144	−0.00180 [−0.00191, −0.00169]
0.25	0.252	0.251	830/26	0.53067	0.52691	−0.00376 [−0.00391, −0.00361]
0.50	0.503	0.501	823/26	0.55584	0.55009	−0.00575 [−0.00596, −0.00554]
1	1.005	1.001	785/26	0.59163	0.58550	−0.00613 [−0.00637, −0.00589]
2	2.010	2.004	475/26	0.61597	0.61528	−0.00070 [−0.00100, −0.00040]
5	5.056	5.056	172/23	0.65519	0.64275	−0.01244 [−0.01292, −0.01195]
10	10.151	10.152	83/19	0.71000	0.69430	−0.01569 [−0.01658, −0.01481]
20	20.314	20.306	20/8	0.72204	0.70383	−0.01821 [−0.02061, −0.01581]
30	30.670	30.684	10/5	0.78768	0.78868	+0.00100 [−0.00114, +0.00314]
40	41.148	41.178	5/2	0.86238	0.87534	+0.01296 [+0.01139, +0.01454]
50	51.303	51.312	1/1	0.88001	0.89405	+0.01404 [+0.01030, +0.01778]

No eligible comparisons remained at requested rates of 60, 70, 80, 90, 95, 99, or 100%.

Table 11: Matched-activation-rate comparison under `big` . Definitions and confidence intervals follow Table 10.

Requested active (%)	Observed active (%)		N/L	Post AUROC		Δ [95% CI]
	SAE	TC		SAE	TC	
0.10	0.100	0.101	832/26	0.51326	0.51131	−0.00196 [−0.00206, −0.00185]
0.25	0.251	0.251	832/26	0.53111	0.52727	−0.00384 [−0.00399, −0.00369]
0.50	0.504	0.502	829/26	0.55624	0.55017	−0.00607 [−0.00627, −0.00588]
1	1.008	1.005	779/26	0.59053	0.58381	−0.00673 [−0.00694, −0.00651]
2	2.012	2.009	461/26	0.62019	0.61674	−0.00345 [−0.00375, −0.00315]
5	5.061	5.069	172/23	0.65521	0.63728	−0.01793 [−0.01852, −0.01734]
10	10.194	10.202	77/15	0.66893	0.66013	−0.00880 [−0.00972, −0.00788]
20	20.466	20.442	19/7	0.67368	0.69465	+0.02097 [+0.01891, +0.02302]
30	31.174	31.140	4/1	0.84003	0.85032	+0.01029 [+0.00786, +0.01271]
40	40.715	40.714	9/3	0.72958	0.75258	+0.02300 [+0.02133, +0.02466]
50	51.320	51.269	3/1	0.87572	0.87863	+0.00291 [+0.00098, +0.00485]

No eligible comparisons remained at requested rates of 60, 70, 80, 90, 95, 99, or 100%.

B.4 AUROC Decomposition and Activation Restoration

Tables 12 and 13 decompose the pre-to-post AUROC reduction and report AUROC after restoring suppressed positive values. Restoration replaces zero activations with their preactivation values when $0 < a_j \leq \theta_j$, while keeping all other features and parameters fixed. The restored scores equal ReLU scores, so the restoration gain equals the ReLU–Post reduction.

Table 12: Layer-wise AUROC decomposition and restoration under `small`. Pre–ReLU and ReLU–Post denote the two successive AUROC reductions. Restored AUROC follows restoration of suppressed positive values. Held-out results are averaged equally across 32 clusters per layer and then across 26 layers. Values are rounded independently.

	Pre–ReLU		**ReLU–Post**		**Restored AUROC**	
Layer	**SAE**	**TC**	**SAE**	**TC**	**SAE**	**TC**
0	0.15733	0.08146	0.02737	0.00972	0.74294	0.73199
1	0.17712	0.10079	0.01664	0.00306	0.66760	0.66080
2	0.22217	0.13132	0.01394	0.00301	0.66418	0.66353
3	0.20590	0.14981	0.01562	0.00384	0.63244	0.64412
4	0.18921	0.14158	0.02703	0.00239	0.57736	0.56331
5	0.07473	0.04131	0.06028	0.01240	0.69282	0.68071
6	0.07518	0.06403	0.06094	0.01581	0.68698	0.61892
7	0.09985	0.10767	0.05229	0.02309	0.65509	0.63136
8	0.08322	0.09708	0.06726	0.02455	0.68790	0.62952
9	0.08942	0.11968	0.05839	0.01617	0.69098	0.67937
10	0.06068	0.05041	0.06271	0.03648	0.73587	0.73317
11	0.03679	0.05422	0.05267	0.02250	0.75162	0.74653
12	0.06940	0.06049	0.03742	0.02201	0.75240	0.76083
13	0.10020	0.13393	0.05324	0.03282	0.71429	0.72219
14	0.10089	0.11755	0.05374	0.03090	0.76795	0.77811
15	0.09771	0.07527	0.06418	0.02428	0.77094	0.79875
16	0.05334	0.07582	0.05596	0.03188	0.82112	0.83131
17	0.08541	0.08966	0.05238	0.02424	0.77672	0.78348
18	0.15469	0.14565	0.04651	0.02448	0.72843	0.75034
19	0.14357	0.12756	0.03972	0.02832	0.72267	0.74958
20	0.13126	0.11576	0.04567	0.02902	0.75169	0.76494
21	0.13363	0.13237	0.04662	0.03644	0.72642	0.74410
22	0.15501	0.14861	0.04519	0.03351	0.71226	0.74113
23	0.11642	0.09536	0.03780	0.02963	0.72122	0.74478
24	0.15459	0.14989	0.03837	0.01993	0.71265	0.72987
25	0.14151	0.11933	0.03598	0.02030	0.70836	0.73435
Mean	**0.11959**	**0.10487**	**0.04492**	**0.02157**	**0.71434**	**0.71604**

Table 13: AUROC decomposition and restoration under `big`. Definitions and aggregation follow Table 12. Features remain fixed within each comparison but are selected separately under `small` and `big`.

	Pre–ReLU		ReLU–Post		Restored AUROC	
Layer	**SAE**	**TC**	**SAE**	**TC**	**SAE**	**TC**
0	0.15668	0.07169	0.01736	0.01401	0.73982	0.74426
1	0.18892	0.10201	0.01033	0.00239	0.66852	0.65778
2	0.21912	0.12145	0.00972	0.00204	0.66259	0.65723
3	0.22067	0.15005	0.01028	0.00230	0.63003	0.63422
4	0.18993	0.13087	0.01499	0.00323	0.57008	0.56524
5	0.09981	0.01511	0.02158	0.00564	0.66805	0.69796
6	0.10380	0.06086	0.02091	0.00887	0.65306	0.62922
7	0.12498	0.09452	0.00941	0.00724	0.62760	0.64466
8	0.11680	0.06533	0.01213	0.00046	0.66266	0.66229
9	0.12465	0.07605	0.01767	0.00758	0.66949	0.70150
10	0.09233	0.06462	0.02218	0.01203	0.71558	0.75096
11	0.06647	0.05592	0.02236	0.01173	0.72510	0.74324
12	0.08865	0.04567	0.01950	0.01009	0.74334	0.76981
13	0.11344	0.08915	0.02122	0.01627	0.69474	0.74689
14	0.11977	0.12162	0.01949	0.01319	0.74012	0.76639
15	0.10885	0.08524	0.02560	0.01828	0.74923	0.80076
16	0.06714	0.06359	0.02221	0.01386	0.80799	0.83508
17	0.11001	0.09999	0.02278	0.01229	0.75652	0.78724
18	0.14967	0.15168	0.02342	0.01734	0.71804	0.75585
19	0.16252	0.13602	0.02559	0.01515	0.71415	0.74416
20	0.13824	0.11462	0.02525	0.01721	0.74374	0.77016
21	0.15647	0.14187	0.02735	0.01805	0.71106	0.73895
22	0.17180	0.15283	0.02772	0.02068	0.70388	0.73499
23	0.13181	0.10968	0.02394	0.01540	0.70678	0.72465
24	0.17148	0.14890	0.01932	0.01426	0.69246	0.72907
25	0.15149	0.11854	0.01716	0.01515	0.70036	0.73495
Mean	**0.13636**	**0.09953**	**0.01959**	**0.01134**	**0.69904**	**0.72029**

Table 14: Layers with decreased or increased output error after restoration under `small`. Squared output error is averaged over all tokens in each target or nontarget group, including unaffected tokens, then equally across clusters. Counts indicate direction, not magnitude; the final column lists layer indices.

Coder	Token group	Layers with lower error	Layers with higher error
SAE	Target	23/26	0, 4, 6
TC	Target	24/26	0, 1
SAE	Nontarget	22/26	0, 1, 4, 6
TC	Nontarget	23/26	0, 1, 5

B.5 Encoder Interventions Across Subspace Sizes

We extend the reference-directed intervention in Table 3 to the leading $k \in \{1, 4, 16, 64, 256\}$ output-explanatory directions. For each k, rotation angles are matched between SAE and TC, and each targeted rotation is compared with eight angle- and norm-matched random rotations. Results are evaluated on held-out documents and averaged equally across 32 clusters per layer and all 26 layers.

Table 15 shows that the additional postactivation AUROC reduction is larger for TC at $k = 4, 16, 64, 256$ under both settings. At $k = 1$, targeted rotations instead yield higher AUROC than random rotations for both coders, with a larger difference for TC. Thus, the greater sensitivity of TC does not extend to every subspace size. Rotation angles also vary with k, so these results do not isolate the effect of adding directions at a fixed rotation angle.

For $k = 16$, the additional reduction is larger for TC in 23 of 26 layers under `small` and 24 of 26 under `big`. The exceptions are layers 3, 4, and 16 under `small` and layers 3 and 16 under `big`.

Table 15: Additional held-out postactivation AUROC reduction (Random minus Targeted) across subspace sizes. Δ denotes TC minus SAE; brackets give 95% paired document-bootstrap confidence intervals.

Setting	k	**SAE**	**TC**	**Δ [95% CI]**
small	1	-0.00014	-0.00259	-0.00245 [-0.00250, -0.00239]
	4	-0.00265	$+0.00615$	$+0.00880$ [$+0.00871$, $+0.00890$]
	16	$+0.00118$	$+0.01189$	$+0.01071$ [$+0.01060$, $+0.01081$]
	64	$+0.00547$	$+0.01411$	$+0.00864$ [$+0.00853$, $+0.00875$]
	256	$+0.00314$	$+0.00854$	$+0.00540$ [$+0.00532$, $+0.00548$]
big	1	-0.00024	-0.00180	-0.00156 [-0.00161, -0.00150]
	4	-0.00306	$+0.00907$	$+0.01213$ [$+0.01204$, $+0.01222$]
	16	$+0.00092$	$+0.01530$	$+0.01438$ [$+0.01427$, $+0.01450$]
	64	$+0.00508$	$+0.01671$	$+0.01162$ [$+0.01151$, $+0.01173$]
	256	$+0.00292$	$+0.00904$	$+0.00612$ [$+0.00604$, $+0.00620$]

B.6 INPUT-PC CONTROLS ACROSS SUBSPACE SIZES

At matched rotation angles, reducing TC's input-PC contribution produces lower postactivation AUROC than reducing its output-explanatory contribution at all five subspace sizes under both settings (Tables 16 and 17). Thus, the sensitivity observed for output-explanatory directions is not exclusive to those directions. At $k = 1$, the output-explanatory intervention outperforms random rotation, whereas the input-PC intervention underperforms it.

All conditions within each comparison use the same eligible layer–cluster pairs. Eligibility follows the angle constraint in Appendix A.1; the eligible sets vary at $k = 1, 4$. Results are averaged equally across eligible clusters within each layer and then across all 26 layers.

Table 16: Held-out postactivation AUROC for matched-angle subspace decreases. Output and PC denote reduced contributions from output-explanatory directions and input principal components, respectively. Random averages eight matched rotations. N is the number of eligible layer–cluster pairs out of 832.

Setting	k	Coder	N	Original	Random	Output	PC
small	1	SAE	561	0.66843	0.66788	0.67152	0.67152
		TC	561	0.69291	0.69209	0.69357	0.69058
	4	SAE	817	0.67021	0.66943	0.67148	0.67148
		TC	817	0.69535	0.69419	0.69353	0.69196
	16	SAE	832	0.66918	0.66840	0.66792	0.66792
		TC	832	0.69413	0.69297	0.68963	0.68902
	64	SAE	832	0.66918	0.66840	0.66651	0.66651
		TC	832	0.69413	0.69297	0.68838	0.68796
	256	SAE	832	0.66918	0.66840	0.66700	0.66700
		TC	832	0.69413	0.69297	0.68902	0.68868
big	1	SAE	606	0.67456	0.67402	0.67694	0.67694
		TC	606	0.70431	0.70361	0.70446	0.70149
	4	SAE	820	0.67905	0.67834	0.68003	0.68003
		TC	820	0.70852	0.70754	0.70590	0.70508
	16	SAE	832	0.67930	0.67857	0.67778	0.67778
		TC	832	0.70862	0.70762	0.70371	0.70337
	64	SAE	832	0.67930	0.67857	0.67642	0.67642
		TC	832	0.70862	0.70762	0.70262	0.70238
	256	SAE	832	0.67930	0.67857	0.67704	0.67704
		TC	832	0.70862	0.70762	0.70354	0.70323

Table 17: Additional postactivation AUROC reduction: Random minus the corresponding intervention in Table 16. SAE's two interventions coincide. The final column compares TC's additional reductions; positive values indicate greater reduction from input-PC intervention. Brackets give 95% paired document-bootstrap intervals from 1,000 resamples.

Setting	k	SAE Both	TC Output	TC PC	TC: PC − Output [95% CI]
small	1	−0.00364	−0.00148	0.00151	0.00299 [0.00293, 0.00305]
	4	−0.00205	0.00066	0.00223	0.00157 [0.00152, 0.00161]
	16	0.00048	0.00335	0.00395	0.00061 [0.00056, 0.00065]
	64	0.00189	0.00460	0.00502	0.00042 [0.00037, 0.00047]
	256	0.00140	0.00396	0.00429	0.00034 [0.00029, 0.00039]
big	1	−0.00291	−0.00085	0.00212	0.00297 [0.00292, 0.00302]
	4	−0.00170	0.00165	0.00246	0.00081 [0.00077, 0.00085]
	16	0.00079	0.00391	0.00425	0.00034 [0.00029, 0.00039]
	64	0.00215	0.00500	0.00525	0.00024 [0.00020, 0.00029]
	256	0.00153	0.00408	0.00440	0.00031 [0.00027, 0.00036]

B.7 SAE-Referenced TC Adjustment Across Layer Groups

Table 18 separates the leading-16-PC adjustment by layer group. The adjustment reduces TC's advantage more in layers 0–12, whereas a substantial positive gap remains in layers 13–25. Mean rotation angles are also smaller in the later layers, so these comparisons do not isolate depth from intervention magnitude.

Table 18: SAE-referenced TC adjustment for the leading 16 input PCs, averaged equally across 32 clusters and then across layers within each group. SAE remains unchanged. Random averages eight angle- and norm-matched rotations. Brackets give 95% paired document-bootstrap confidence intervals.

(a) Postactivation AUROC

Setting	Layers	SAE Original	TC Original	TC Random	TC Adjusted
small	0–8	0.62936	0.63591	0.62503	0.61245
	9–12	0.68016	0.70541	0.68660	0.66578
	13–25	0.69337	0.73096	0.72902	0.72400
big	0–8	0.63943	0.64931	0.63755	0.62055
	9–12	0.69355	0.73034	0.70983	0.67964
	13–25	0.70252	0.74300	0.74062	0.73400

(b) Remaining gap, additional reduction, and rotation angle

Setting	Layers	Adjusted TC–SAE	TC Random − Adjusted [95% CI]	Mean angle
small	0–8	−0.01691	0.01258 [0.01241, 0.01275]	22.23°
	9–12	−0.01439	0.02082 [0.02055, 0.02107]	18.44°
	13–25	+0.03063	0.00501 [0.00494, 0.00508]	3.36°
big	0–8	−0.01887	0.01700 [0.01682, 0.01718]	21.94°
	9–12	−0.01392	0.03019 [0.02992, 0.03048]	17.58°
	13–25	+0.03148	0.00662 [0.00655, 0.00669]	3.71°

B.8 Discrimination Recovery from Remaining Encoder Components

Amplifying the remaining encoder-weight components yields greater mean postactivation AUROC recovery for SAE than TC after either complete removal or partial reduction of the leading input-PC components, across all five tested subspace sizes and both settings (Table 19). After complete removal, however, amplification does not restore either coder's original mean AUROC for $k \geq 4$.

Table 19: Postactivation AUROC gain from amplifying the remaining encoder-weight components after complete removal or partial reduction of the leading k input-PC components. Each gain is measured against the corresponding condition without amplification and averaged equally across 32 clusters and 26 layers.

		Complete removal		Partial reduction	
Setting	k	SAE	TC	SAE	TC
small	1	0.02022	0.00988	0.00462	0.00287
	4	0.03181	0.02643	0.00681	0.00500
	16	0.04173	0.03802	0.00589	0.00438
	64	0.04961	0.02833	0.00426	0.00224
	128	0.04814	0.02031	0.00358	0.00140
big	1	0.01861	0.01005	0.00418	0.00222
	4	0.03152	0.02483	0.00614	0.00386
	16	0.04246	0.03528	0.00522	0.00316
	64	0.04934	0.02664	0.00365	0.00121
	128	0.04771	0.01972	0.00301	0.00048

B.9 Entity Detection Before and After JumpReLU

Table 20 compares test F1 using the same postactivation-selected features, with evaluation cutoffs selected separately by validation F1. Mean preactivation and postactivation F1 differ by less than 2×10^{-5} in all four conditions.

Table 20: Few-NERD test F1, averaged equally across 66 entity types and 26 layers.

Setting	Coder	Pre F1	Post F1
small	SAE	0.36770	0.36770
	TC	0.38751	0.38749
big	SAE	0.38349	0.38349
	TC	0.40601	0.40600