Split Features of Sparse Autoencoders Can Be Merged by Training on Constrastive Text Pairs

Anonymous ACL submission

Abstract

Sparse autoencoders (SAEs) decompose the internal representations of large language models (LLMs) into interpretable, monosemantic features. However, a key limitation is *feature splitting*: as the SAE latent space grows, a single semantic concept fractures into multiple sub-concept features, each capturing only part of the whole, which degrades interpretability. In this work, we show that training SAEs on contrastive text pairs can mitigate feature splitting. We first demonstrate that adding lexical perturbations to original texts can activate distinct SAE latents without substantially affecting LLM functionality. We then introduce RoSAE (Robust SAE), a training strategy that penalizes inconsistent activation patterns across contrastive text pairs. For state-of-the-art BatchTopK SAEs trained on Gemma-2-2B, RoSAE improves performance on spurious correlation removal by 4.17%, and reduce feature absorption by 1.97%, while maintain competitive reconstruction fidelity, suggesting that training on contrastive text pairs is an effective technique for producing more general SAE features.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, yet their internal mechanism remains poorly understood (Zhao et al., 2024; Rai et al., 2024). A central challenge is that the internal representations of LLMs are dense and entangled, making it difficult to attribute model behavior to specific, interpretable concepts (Olah et al., 2017; Elhage et al., 2022). Sparse autoencoders (SAEs) have recently emerged as a promising tool for addressing this challenge: by mapping dense latent representations into a higher-dimensional sparse space, SAEs decompose complex internal activations into monosemantic, human-interpretable features (Bricken et al., 2023; Huben et al., 2024).

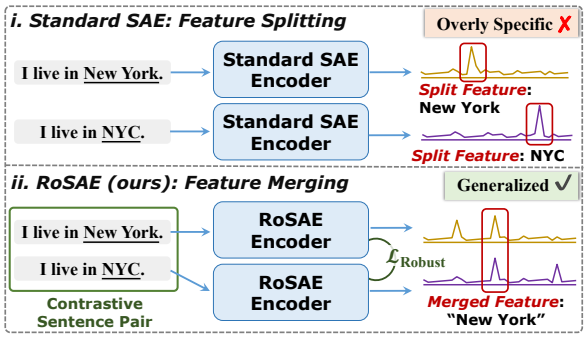


Figure 1: Comparison between the standard SAE and RoSAE in feature learning. Without explicit constraints on feature granularity, a standard SAE may learn overly specific features tailored to individual inputs (e.g., learning two single features for "New York" and "NYC" rather than a general feature), resulting in feature splitting. By enforcing activation consistency between semantically perturbed input pairs, RoSAE encourages the encoder to learn more general features.

However, current interpretation tools utilizing SAEs suffer from *feature splitting* (Bricken et al., 2023): as the SAE latent space grows, a single semantic concept is fragmented into multiple sub-concept features, each capturing only a partial aspect of the original concept. For instance, Bricken et al. (2023) observe that as the SAE dictionary size grows, features encoding base64[1] text split from one into three, separately capturing letters, digits, and ASCII-encoded characters. Feature splitting introduces obstacles for both the interpretability and downstream utility of SAEs. On the interpretability side, split features may absorb the activation of related general features (Chanin et al., 2024); Additionally, SAEs may learn composite features of frequently co-occurring concepts rather than decomposing them into orthogonal ones (Leask et al., 2025). On the application side, feature splitting complicates LLM behavior detection and steering:

[1] A binary-to-text encoding scheme that represents binary data using 64 printable ASCII characters.

when a single behavioral pattern is fragmented across multiple latents, incomplete identification or steering of all relevant features can degrade performance on modification tasks such as detoxification (Goyal et al., 2025).

While prior work has explored architectural modifications to SAEs to alleviate feature splitting (Bussmann et al., 2025; Luo et al., 2026), we instead address the problem from a data-centric perspective. We observe that substituting words with synonyms causes activated SAE features to differ significantly across the resulting sentence pair, while the LLM exhibits nearly identical behavior on both inputs, suggesting that existing SAEs capture surface-level textual variations rather than underlying semantic features. Motivated by this, we propose Robust SAE (RoSAE), which constructs such contrastive pairs and encourages SAEs to rely on latents that behave consistently across them, thereby pushing the model toward more general features. Across multiple SAE architectures and base models, RoSAE consistently improves feature quality. On state-of-the-art BatchTopK SAEs trained on Gemma-2-2B, it improves spurious correlation removal (SCR) by 4.17% and reduces feature absorption by 1.97%, while maintaining competitive reconstruction fidelity.

In summary, our main contributions are:

- We show that synonym substitution-based perturbations can activate substantially different SAE latents while leaving LLM behavior nearly unchanged, motivating a contrastive training signal.
- We propose RoSAE, a contrastive SAE training strategy that enforces activation consistency across semantically equivalent input pairs to mitigate feature splitting.
- We evaluate RoSAE across multiple SAE architectures and base models, demonstrating consistent improvements in SCR and feature absorption reduction while maintaining competitive reconstruction fidelity.

2 Preliminary

2.1 Sparse Autoencoders

SAEs leverage a linear projection to map an LLM representation into a high-dimensional sparse latent space, where concepts encoded in representations can be decomposed into monosemantic features. A standard SAE consists of an encoder and a decoder: the encoder projects the input representation into a sparse latent space, while the decoder reconstructs the original representation from these sparse features, enabling unsupervised training. Formally, the forward pass of an SAE is defined as:

$$\mathbf{z} = \sigma(\mathbf{W}_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}) \tag{1}$$

$$\hat{\mathbf{x}} = \mathbf{W}_{\text{dec}}\mathbf{z} + \mathbf{b}_{\text{dec}}, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^d$ denotes a dense input representation, $\sigma(\cdot)$ denotes an activation function (e.g., ReLU (Bricken et al., 2023) or JumpReLU (Rajamanoharan et al., 2024)), and $(\mathbf{W}_{\text{enc}}, \mathbf{b}_{\text{enc}})$ and $(\mathbf{W}_{\text{dec}}, \mathbf{b}_{\text{dec}})$ are the weight matrices and bias terms for the encoder and decoder, respectively. The reconstruction fidelity of an SAE is evaluated by the mean squared error (MSE) between the original input $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}}$, computed as:

$$\mathcal{L}_{\text{reconstruct}}(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2. \tag{3}$$

2.2 Sparsity Constraints of SAEs

SAE imposing a sparsity penalty on the feature activations z to drive the decomposition of dense representations into sparse, monosemantic features. In standard ReLU SAEs (Bricken et al., 2023) , it is penalized by the L_1 norm of the latent activations, which works as an auxillary loss:

$$\mathcal{L}_{\text{sparsity}}(\mathbf{x}) = \|\mathbf{z}\|_1. \tag{4}$$

The total loss would be the weighted sum of the above two loss components:

$$\mathcal{L}(\mathbf{x}) = \mathcal{L}_{\text{reconstruct}}(\mathbf{x}) + \alpha\mathcal{L}_{\text{sparsity}}(\mathbf{x}), \tag{5}$$

where α is a coefficient controlling the trade-off between sparsity and reconstruction quality.

Beyond standard SAEs, TopK SAEs (Gao et al., 2025) employ a hard selection mechanism that retains only the k largest latents, enforcing exact sparsity. The encoder in TopK SAEs is defined as:

$$\mathbf{z} = \text{TopK}(\mathbf{W}_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}) \tag{6}$$

BatchTopK SAEs (Bussmann et al., 2024) further extend this constraint to the batch level, improving training stability of the TopK approach.

2.3 Evaluation of SAEs

The two fundamental metrics for evaluating an SAE are reconstruction fidelity and sparsity.

Reconstruction. We use *Delta LM loss* to measure reconstruction fidelity. It is computed as:

$$\Delta\mathcal{L} = \hat{H} - H, \tag{7}$$

where H denotes the cross-entropy loss of the original model on next-token prediction, and $\hat{H}$ denotes the cross-entropy loss obtained by substituting the model activation $\mathbf{x}$ with its SAE reconstruction $\hat{\mathbf{x}}$ during the forward pass. A lower Delta LM loss indicates higher reconstruction fidelity.

Sparsity. We use the L_0 norm to measure sparsity, defined as the average number of active features per token. Lower L_0 values correspond to sparser representations. Karvonen et al. (2025) suggest that an L_0 value in the range of 20–200 achieves a favorable balance between interpretability and reconstruction performance.

These two objectives involve an inherent trade-off: optimizing solely for reconstruction reduce the SAE to an identity mapping with no feature disentanglement, whereas aggressive sparsity constraints risk discarding information. A well-trained SAE should lie on the Pareto frontier of this trade-off, preserving original representation while maintaining interpretable, sparse features.

However, Karvonen et al. (2025) argue that the Pareto frontier alone does not fully characterize SAE quality. To this end, SAEBench proposes a comprehensive evaluation framework that assesses SAEs along multiple dimensions beyond reconstruction and sparsity, including absorption, automated interpretability, and downstream task performance, among others.

3 Split Features Can Be Identified through Contrastive Text Pairs

Our approach relies on contrastive text pairs as its core training signal. For this approach to be well-founded, these pairs must satisfy two properties: first, lexical perturbed sentences should elicit substantially different SAE feature activations, thereby exposing the feature splitting that our training objective addresses; second, the perturbations should not alter the functional behavior of the underlying LLM, confirming that any activation differences reflect lexical variation rather than semantic change. In this section, we present empirical evidence that both properties hold: perturbed inputs lead to substantially different SAE feature activations, while LLM task performance remains largely unaffected.

3.1 Experimental Setup

Perturbation Methods. We utilize TextAttack (Morris et al., 2020) to introduce two levels of perturbations to the original text:

- **Character-level.** Randomly selected characters are replaced with arbitrary characters.
- **Word-level.** Randomly selected words are replaced with synonyms that maintain an embedding cosine similarity of at least 0.6 in WordNet (Miller et al., 1990).

Examples of the contrastive text pairs are provided in Table 3 in Appendix B.

Datasets. We conduct our analysis across three downstream tasks: SST-2 (Socher et al., 2013), BoolQ (Christopher et al., 2019) and ARC-e (Clark et al., 2018). 100 sentences of similar length are selected per task. For each instance, a perturbed version is generated at every perturbation level to form contrastive text pairs. Additional results on further datasets are provided in Appendix C.

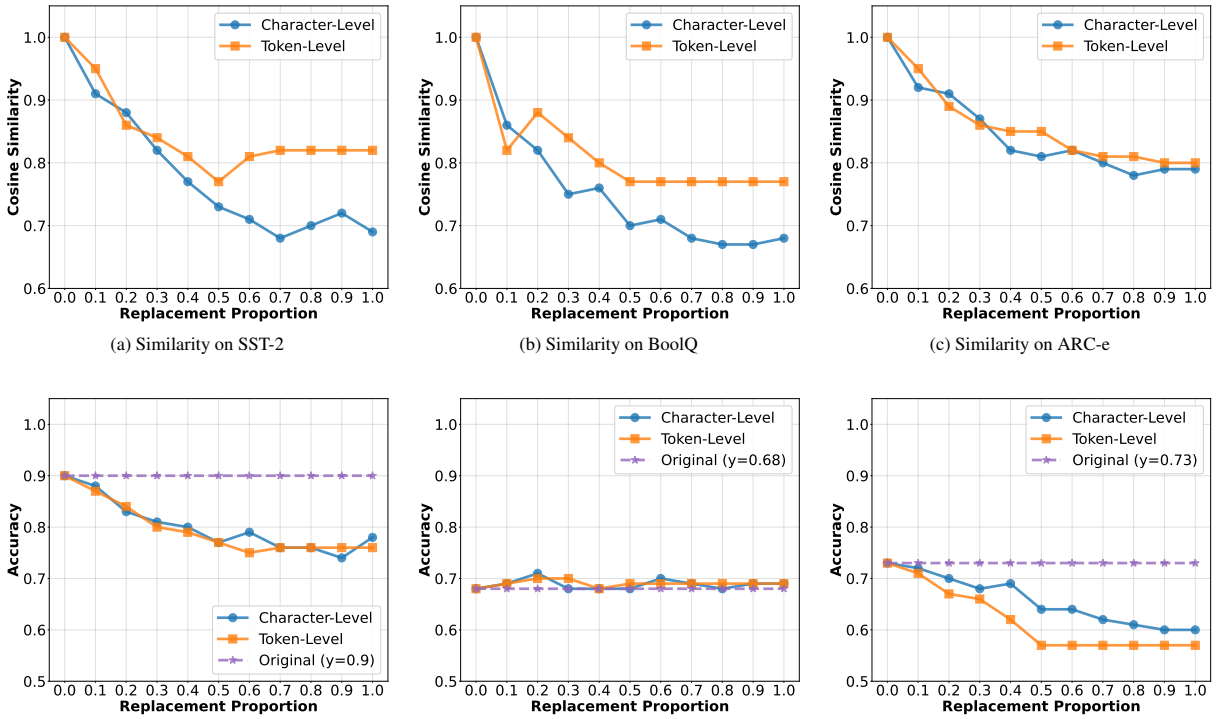
SAEs. We utilize Gemma Scope (Lieberum et al., 2024), an open-source SAE suite trained on Gemma 2 (Rivière et al., 2024). Our analysis focuses on SAEs trained on the final layer of the 2B model. We use the output on residual stream as the input x to SAEs, as it has been shown to have better interpretability (Ferrando et al., 2024).

Metrics. We use the average cosine similarity to measure the divergence between feature vectors extracted from the original and perturbed inputs. To eliminate the influence of non-informative features, we excluded features that exhibit persistent activation (i.e., activated by 100% of the sentences across the entire dataset) during similarity computation. We hypothesize that such features encode invariant linguistic information rather than the task-specific functional properties required for our analysis.

3.2 Results

As shown in Figure 2, we derive the following observations:

SAE feature similarity drops rapidly under perturbations. Fig. 2(a)-(c) show the cosine similarity between SAE feature vectors extracted from original and perturbed inputs across three tasks. As the replacement proportion increases, the feature similarity drops rapidly under both perturbation



(a) Similarity on SST-2 (b) Similarity on BoolQ (c) Similarity on ARC-e

(d) Accuracy on SST-2 (e) Accuracy on BoolQ (f) Accuracy on ARC-e

Figure 2: Impact of input perturbations on SAE feature similarity and downstream task performance across SST-2, BoolQ, and ARC-e datasets. Panels (a)–(c) illustrate the cosine similarity between SAE feature vectors of original and perturbed inputs, showing a rapid decline as the replacement proportion increases. Panels (d)–(f) show the corresponding LLM accuracy on these tasks, remaining relatively stable despite significant shifts in feature representations.

levels[2]. We assume that low similarity scores indicate that the SAE activates different features for original and perturbed inputs.

Robustness of Downstream Performance against Representation Shift. Despite the divergence of activated features, the downstream performance of the LLM remains resilient. As shown in Fig. 2(d)-(f), the model's accuracy on perturbed data shows no significant degradation compared to the original results. The misalignment between SAE features and LLM performance suggests that while the SAE activates different features under perturbations, these features may have similar functions.

Together, these results support the use of lexically perturbed text pairs as a reliable training signal: they expose differences in SAE feature representations without introducing confounding changes in LLM behavior.

[2]The decline in similarity plateaus at higher replacement proportions due to saturation: high-level character noise results in semantic-free gibberish, while the number of synonyms limits token-level substitution.

4 Merge Split SAE Features in Training

In this section, we propose RoSAE (Robust SAE), a training strategy that mitigates feature splitting by leveraging contrastive text pairs as a supervision signal. Specifically, We encourage SAEs to activate features that respond consistently across semantically related sentences, promoting the learning of more generalizable and semantically grounded representations, thereby mitigating feature splitting.

4.1 Training Method

Let $\mathbf{x} \in \mathbb{R}^d$ denote a hidden-state representation extracted from a designated layer of a frozen LLM, and $\mathbf{x}' \in \mathbb{R}^d$ its perturbed counterpart, where d is the hidden dimension. The SAE encodes each input into sparse latent vectors $\mathbf{z} \in \mathbb{R}^m$ and $\mathbf{z}' \in \mathbb{R}^m$, where m denotes the SAE dictionary size. We define a split feature as any feature z_i that is active for $\mathbf{x}$ but inactive for $\mathbf{x}'$. Since $\mathbf{x}$ and $\mathbf{x}'$ are semantically equivalent by construction, a feature activated by only one of them is unlikely to capture a stable, generalizable concept. As shown in Sec. 3.2, such features contribute negligibly to LLM func-

tion, suggesting they arise from surface-level variation rather than meaningful semantic content.

We therefore introduce a penalty term to suppress these split features. For standard SAEs, this takes the form of an auxiliary L_1 penalty applied exclusively to split features. Formally, in addition to the standard reconstruction loss $\mathcal{L}_{\text{reconstruct}}(\mathbf{x})$ and sparsity loss $\mathcal{L}_{\text{sparsity}}(\mathbf{x})$, we introduce a contrastive robustness loss $\mathcal{L}_{\text{robust}}(\mathbf{x})$[3]:

$$\mathcal{L}_{\text{robust}}(\mathbf{x}) = \sum_i \lambda \|z_i\|_1, \tag{8}$$

where

$$\lambda = \begin{cases} 1, & \text{if } z_i > 0, z_i' = 0 \\ 0. & \text{otherwise} \end{cases} \tag{9}$$

Consequently, the total loss for RoSAE is formulated as:

$$\mathcal{L}(\mathbf{x}) = \mathcal{L}_{\text{reconstruct}}(\mathbf{x}) + \alpha \mathcal{L}_{\text{sparsity}}(\mathbf{x}) + \beta \mathcal{L}_{\text{robust}}(\mathbf{x}), \tag{10}$$

where α and β are coefficients controlling the trade-off among different constraints.

For TopK SAEs, which impose no L_1 regularization, we adopt an alternative strategy: during the TopK masking step, we additionally zero out any feature that is not co-activated across both sentences in the pair. This encourages the SAE to rely only on features consistently activated by semantically equivalent inputs, thereby reducing split features without an additional explicit sparsity penalty. The encoder of RoSAE based on TopK SAEs is defined as:

$$\mathbf{z} = \lambda \odot \text{TopK}(\mathbf{W}_{\text{enc}}\,\mathbf{x} + \mathbf{b}_{\text{enc}}) \tag{11}$$

4.2 Training Setup

Datasets. We train SAEs on OpenWebText (Gokaslan and Cohen, 2019). Since SAEs are trained at the token level, we construct contrastive training pairs via token-wise perturbations: for each original token, we sample a substitute whose embedding cosine similarity to the original exceeds a threshold θ in the LLM's embedding space, preserving total number of tokens after perturbation. We set $\theta = 0.6$ based on the ablation study in Sec. 5.1. All data are pre-tokenized prior to training to improve efficiency.

[3]Note that λ is defined asymmetrically between $\mathbf{z}$ and $\mathbf{z}'$, with detailed justification provided in Appendix A.

Following SAEBench (Karvonen et al., 2025), which finds that SAEs perform best when trained on roughly 50M tokens, we train each SAE on up to 200M tokens and retain the checkpoint closest to the Pareto frontier across evaluated trade-off configurations.

Base Models. We evaluate on two base LLMs: Gemma-2-2B and Qwen-2.5-1.5B. Following Gemma Scope (Lieberum et al., 2024), we select three representative layers of Gemma-2-2B for analysis: shallow (layer 5), middle (layer 12), and deep (layer 19), using the residual stream as the input source. For Qwen-2.5-1.5B, we conduct experiments on layer 12 as a cross-model validation of our approach.

SAE Variants. We train RoSAE on top of two SAE architectures: standard ReLU SAEs and the state-of-the-art BatchTopK SAEs. All SAEs use a dictionary size of 16384.

Implementation. Our training codebase is built on SAE-Lens (Bloom et al., 2024), and evaluation experiments are conducted within the SAEBench (Karvonen et al., 2025) framework, using two NVIDIA GeForce RTX 3090 GPUs.

4.3 Main Results

Figure 3 shows the experimental results on layer 12 of Gemma-2-2B.

Delta LM Loss. We first assess reconstruction quality by examining the trade-off between reconstruction fidelity and feature sparsity. As illustrated in Figure 3(a), RoSAE consistently achieves lower delta LM loss across all sparsity levels for ReLU SAEs, indicating that RoSAE enables ReLU SAEs to more effectively encode semantic information. For BatchTopK SAEs, which represent the state-of-the-art in reconstruction quality, RoSAE yields a modest reduction in delta LM loss. This may be attributed to the fact that RoSAE encourages the model to prioritize more generalizable features over highly specialized ones.

Feature Absorption: Feature absorption refers to the phenomenon where overly specific features in an SAE suppress the activation of more general features (Chanin et al., 2024), as the sparsity penalty discourages redundant co-activation. We evaluate absorption using the Mean Absorption Fraction Score, where lower values indicate less absorption and thus better feature disentanglement.

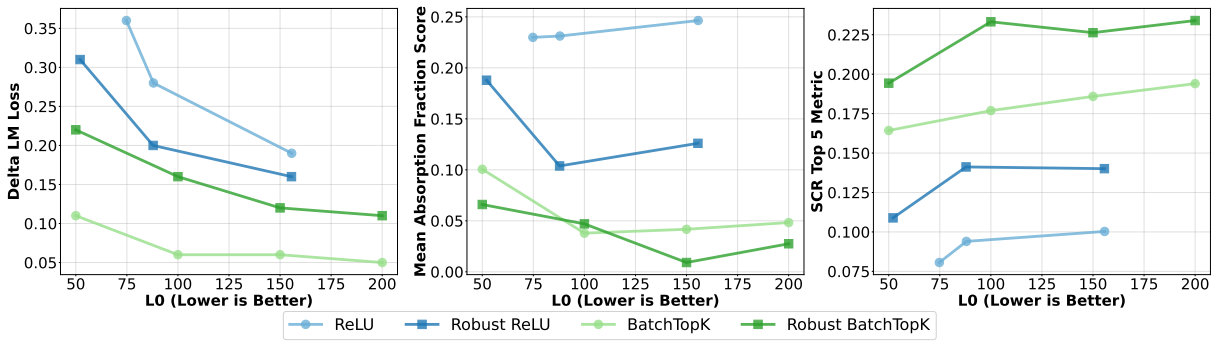


Figure 3: Main experimental results on layer 12 of Gemma-2-2B across varying L_0 sparsity levels. (a) Delta LM loss (lower is better), measuring reconstruction fidelity. (b) Mean Absorption Fraction Score (lower is better), measuring feature absorption. (c) SCR score (higher is better), measuring the effectiveness of targeted causal interventions. RoSAE is evaluated on top of both ReLU and BatchTopK SAE architectures. Results demonstrate that RoSAE consistently reduces feature absorption and improves causal intervention effectiveness, while maintaining competitive reconstruction quality.

As shown in Figure 3(b), RoSAEs achieve substantially lower absorption scores than baselines across most L_0 values, demonstrating that our training approach produces features that are more disentangled and less prone to absorbing general semantic information.

Spurious Correlation Removal (SCR): SCR quantifies an SAE's ability to isolate target signals by intervening in the latent space. In classification tasks, SCR measures the recovery of target prediction accuracy after performing zero-ablation on the latent features most correlated with the spurious attribute. When an SAE suffers from feature splitting, where a single semantic concept is distributed across multiple features, suppressing a limited subset of those features is insufficient to fully decouple the spurious correlation. As illustrated in Figure 3(c), RoSAE substantially outperforms the baseline on both SAE architectures. By encouraging more concentrated conceptual encoding, RoSAE enhances the effectiveness of targeted causal interventions.

For results on other layers and Qwen-2.5-1.5B, please see Appendix D.

5 Analysis

5.1 Ablation on Perturbation Threshold

To construct contrastive text pairs, we replace each token with a candidate whose embedding cosine similarity exceeds a threshold θ. In this section, we analyze how θ affects the proportion of replaceable tokens. Figure 4 shows the proportion of tokens eligible for replacement as θ varies. The proportion remains near 100% for $\theta \leq 0.3$, meaning almost

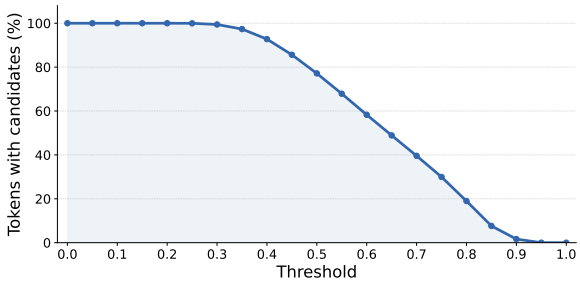


Figure 4: Proportion of tokens eligible for replacement under varying perturbation threshold θ. Higher thresholds impose stricter similarity requirements, reducing the fraction of tokens that have a valid substitute.

every token can be perturbed at low thresholds. As θ increases, the proportion drops steeply and approaches near 0 at $\theta = 0.9$, indicating that very high thresholds yield negligible perturbation signal.

We therefore train RoSAE on ReLU SAEs under thresholds θ ranging from 0.4 to 0.8 with a step size of 0.05 to empirically examine the downstream performance of RoSAE. As shown in Figure 5, $\theta = 0.6$ achieves the best overall performance across the configurations evaluated. We therefore adopt $\theta = 0.6$ as the default threshold in all other experiments.

Table 1 shows a representative case for the token `check`: candidates with similarity ≥ 0.6 are predominantly typographic variants (`Check`, `CHECK`), whereas candidates below 0.6 include morphological variants (`checks`, `checking`, `checked`) and cross-lingual equivalents (チェック, 检查). This pattern suggests that $\theta = 0.6$ marks a natural boundary: perturbations above this threshold are surface-level and preserve lexical identity, while

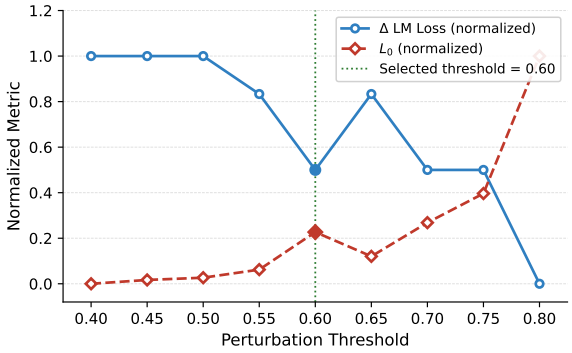


Figure 5: RoSAE performance under varying perturbation threshold θ, evaluated on Delta LM Loss and L_0. $\theta = 0.6$ achieves the best overall performance and is adopted as the default in all experiments.

Table 1: Token similarity candidates for token ID 3534 ("check"). Candidates are ranked by cosine similarity.

Rank	Token ID	Token	Similarity
1	4417	'Check'	0.8605
2	2701	' check'	0.8347
3	6958	' Check'	0.7854
4	17910	'CHECK'	0.6784
5	30116	' CHECK'	0.6439
6	19483	' checks'	0.6065
7	18604	' checking'	0.5943
8	15062	' checked'	0.5720
9	35389	'チェック'	0.5580
10	37251	'检查'	0.5048

those below introduce changes in inflection or language, which may introduce stronger distributional shift into the training signal.

5.2 Ablation on Loss Balance

In Section 4.1, we introduce two coefficients, α and β, balance competing objectives in the RoSAE training loss for ReLU SAE variants: α weights the standard L_1 sparsity penalty $\mathcal{L}_{\text{sparsity}}$, and β weights our proposed auxiliary objective $\mathcal{L}_{\text{Robust}}$. We conduct ablation studies to systematically examine how reallocating the total penalty budget between these two terms affects SAE behavior.

Since $\mathcal{L}_{\text{Robust}}$ applies selective penalization to a subset of latent features, its effective penalty magnitude is bounded between zero and an upper limit functionally equivalent to a full L_1 constraint scaled by β. To isolate the effect of this selective penalization, we design a series of experiments in which a varying proportion of the global penalty budget is transferred from the uniform L_1 term (α) to the selective term (β), while keeping the total penalty sum $\alpha + \beta$ constant within each group. All experiments are conducted on a SAE trained on the residual stream at layer 12 of Gemma-2-2B. Results are summarized in Table 2.

Table 2: Ablation study of SAE on Gemma-2-2B (Layer 12). We compare the effects of the traditional L_1 penalty (α) and the proposed redundancy constraint (β) under equivalent total sparsity constraints ($\alpha + \beta$). The results demonstrate that substituting a portion of α with β improves both reconstruction fidelity (Delta Loss) and feature sparsity (L_0).

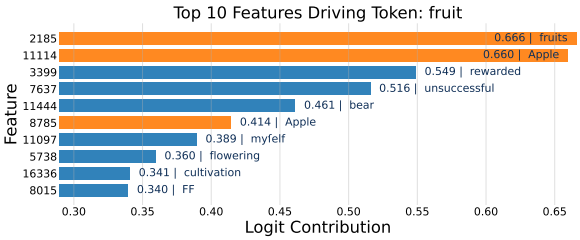
$\alpha+\beta$	α	β	Δ Loss↓	L0↓	L1↑
2.0	2.0	-	0.30	139.16	884.45
	1.0	1.0	**0.23**	**126.21**	**914.16**
3.0	3.0	-	0.39	85.79	682.69
	2.0	1.0	0.36	73.20	685.92
	1.0	2.0	**0.33**	**73.11**	**713.05**
4.0	4.0	-	0.50	63.39	580.92
	3.0	1.0	0.47	53.19	579.82
	2.0	2.0	**0.42**	**50.62**	**583.01**

Quantitative analysis reveals that, under a fixed $\alpha + \beta$, progressively substituting $\mathcal{L}_{\text{sparsity}}$ with $\mathcal{L}_{\text{Robust}}$ yields consistent improvements in reconstruction fidelity, as evidenced by decreasing Delta LM Loss. This improvement can be attributed to the reduced effective sparsity pressure on high-activation features. However, despite the overall relaxation of the sparsity constraint, the L_0 norm does not increase; rather, it exhibits a downward trend, while the L_1 norm remains stable or marginally increases. This divergence between L_0 and L_1 trends suggests that $\mathcal{L}_{\text{Robust}}$ encourages the model to consolidate representational capacity into fewer but higher-magnitude latents, rather than distributing activation across a larger number of low-magnitude features. This behavior is consistent with a reduction in feature splitting, wherein semantically coherent concepts are encoded by dedicated latent directions rather than fragmented across multiple weakly activating features.

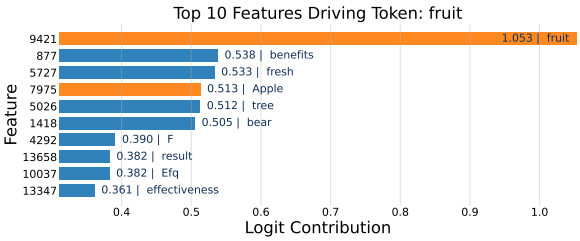
5.3 Case Study

To qualitatively examine the effect of robust training on feature organization, we conduct a case study on the token "fruit" using SAEs trained on layer 19 of Gemma-2-2B. Figure 6 presents the top-10 features by logit contribution for this token under the standard ReLU SAE and the proposed RoSAE, respectively.

In the ReLU SAE (Figure 6(a)), the two highest-



(a) ReLU SAE



(b) Robust ReLU SAE

Figure 6: Top-10 features by logit contribution to the token "fruit" for SAEs trained on layer 19 of Gemma-2-2B. (a) The standard ReLU SAE splits the representation across two near-equal "fruits"/"Apple" features (contributions 0.666 and 0.660) and a second Apple-related feature (0.414), with no single dominant latent. (b) RoSAE consolidates the representation into a single dominant "fruit" feature (contribution 1.053), with remaining features clustered around 0.5, indicating reduced feature splitting.

contributing features are Feature 2185 ("fruits", contribution 0.666) and Feature 11114 ("Apple", contribution 0.660), which are nearly equal in magnitude. This suggests a splitting pattern: rather than consolidating the semantic content of "fruits" into a single dominant feature, the model distributes logit contribution across both the general concept and a more specific hyponym. The splitting is further compounded by the presence of another Apple-related feature, Feature 8785 (contribution 0.414), indicating that even the subordinate concept "Apple" is itself split into two distinct latents. These three features account for a substantial share of the total contribution, with no single feature assuming clear dominance.

In contrast, RoSAE (Figure 6(b)) exhibits a markedly different activation pattern. Feature 9421 emerges as a dominant "fruits" with a substantially higher contribution of 1.053, while the remaining top features cluster around 0.5, leaving a pronounced gap between the top feature and the rest. An Apple-related feature (Feature 7975, contribution 0.513) is still present, but it appears as a single instance. This indicates that the robust training objective encourages the model to consolidate semantically related representations, effectively alleviating the feature splitting phenomenon.

6 Related Work

Several recent SAE architectures have been proposed to address feature splitting. OrtSAE (Korznikov et al., 2025) imposes an orthogonality constraint on the decoder, reducing cosine similarity between features. Matryoshka SAE (Bussmann et al., 2025) learns hierarchical representations across nested latent spaces, encouraging the model to retain coarse-grained summary features alongside finer-grained ones. Hierarchical SAE (Luo et al., 2026) jointly trains a series of SAEs with varying dictionary sizes by explicitly enforcing a tree-structured hierarchy, using a structural constraint loss and a random feature perturbation mechanism to align parent-child features into a consistent conceptual taxonomy.

In contrast, our approach addresses feature splitting from a data-centric perspective, constructing contrastive text pairs to encourage feature separation without modifying the underlying SAE architecture. This positions our method as largely orthogonal to aforementioned architecture-level solutions, and potentially complementary to them.

7 Conclusion

We propose RoSAE, a training strategy that encourages semantically equivalent inputs to activate consistent SAE features. We show that lexical perturbations can substantially change SAE activation patterns while leaving LLM behavior largely unchanged, indicating that many SAE latents correspond to unstable, surface-sensitive split features rather than functionally necessary concepts. Building on this, RoSAE suppresses features activated for an original input but not its perturbed counterpart, via a selective robustness penalty for ReLU SAEs or co-activation filtering for TopK SAEs. Experiments on Gemma-2-2B show that RoSAE improves the reconstruction–sparsity trade-off for ReLU SAEs, reduces feature absorption, and substantially improves spurious correlation removal for both ReLU and BatchTopK architectures. Ablations and a qualitative case study further suggest that RoSAE consolidates fragmented semantic representations into fewer, more stable latents, indicating that contrastive text pairs offer a simple and effective signal for mitigating feature splitting.

8 Limitations

When constructing perturbed text pairs, we employed synonym substitution, leaving more perturbation methods remain to be explored. For example, (Li et al., 2025) demonstrated the robustness vulnerabilities of SAE features through adversarial construction (specifically, appending a few specific tokens after the original input sequence). Furthermore, conceptualizing perturbations as a form of noise, alternative data augmentation techniques such as masking also worth exploration.

References

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

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, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. 2024. Batchtopk sparse autoencoders. *CoRR*, abs/2412.06410.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. 2025. Learning multi-level features with matryoshka sparse autoencoders. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

David Chanin, James Wilken-Smith, Tomás Dulka, Hardik Bhatnagar, and Joseph Bloom. 2024. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *CoRR*, abs/2409.14507.

Clark Christopher, Lee Kenton, Chang Ming-Wei, Kwiatkowski Tom, Collins Michael, and Toutanova Kristina. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Ed Hovy, Hinrich Schutze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *ArXiv*, abs/2102.01017.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*. https://transformer-circuits.pub/2022/toy_model/index.html.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *CoRR*, abs/2405.00208.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Agam Goyal, Vedant Rathi, William Yeh, Yian Wang, Yuen Chen, and Hari Sundaram. 2025. Breaking bad tokens: Detoxification of llms using sparse autoencoders. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 12691–12709. Association for Computational Linguistics.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Anton Korznikov, Andrey V. Galichin, Alexey Dontsov, Oleg Rogov, Elena Tutubalina, and Ivan V. Oseledets. 2025. Ortsae: Orthogonal sparse autoencoders uncover atomic features. *CoRR*, abs/2509.22033.

Patrick Leask, Bart Bussmann, Michael T. Pearce, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. 2025. Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Aaron J. Li, Suraj Srinivas, Usha Bhalla, and Himabindu Lakkaraju. 2025. Interpretability illusions with sparse autoencoders: Evaluating robustness of concept representations. *CoRR*, abs/2505.16004.

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

Yifan Luo, Yang Zhan, Jiedong Jiang, Tianyang Liu, Mingrui Wu, Zhennan Zhou, and Bin Dong. 2026. From atoms to trees: Building a structured feature forest with hierarchical sparse autoencoders. *CoRR*, abs/2602.11881.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to wordnet: An on-line lexical database*. *International Journal of Lexicography*, 3(4):235–244.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 119–126. Association for Computational Linguistics.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature visualization. *Distill*. Https://distill.pub/2017/feature-visualization.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2024. A practical review of mechanistic interpretability for transformer-based language models. *CoRR*, abs/2407.02646.

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

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 80 others. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2):20:1–20:38.

A Discussion on the Asymmetric Robust Loss

In Equation 9, we employ an asymmetric penalty targeting features activated in $\mathbf{z}$ but suppressed in $\mathbf{z}'$, as the perturbed input $\mathbf{x}'$ serves as a reference to identify split features. The reconstruction and sparsity losses are still computed using the original latent representation $\mathbf{z}$. A symmetric loss (penalizing cases where $z_i = 0$ but $z'_i > 0$) would drive gradient descent to force z_i nonzero, inadvertently encouraging the SAE to activate features on the original input that should remain dormant and undermining feature precision. By setting $\lambda = 0$ when $z_i = 0$, the Robust loss acts solely as a pruning force on existing activations rather than an incentive for new ones.

B Perturbation Examples

To provide a qualitative understanding of the perturbations used in our experiments, we present examples of the contrastive text pairs in Table 3. Character-level perturbations involve stochastic character replacement, which simulates common typos or OCR errors while preserving the overall string topology. Word-level perturbations utilize synonym substitution via WordNet. The resulting text often exhibits slight grammatical shifts or domain-specific nuances (e.g., "Undergraduate education" $\rightarrow$ "Baccalaureate teaching"), thereby generating lexically diverse variants to expose split features within SAEs.

Perturbation Level	Texts
Original	Undergraduate education is the post-secondary education previous to the post-graduate education. It includes all the academic programs up to the level of a bachelor's degree. For example, in the United States, an entry level university student is known as an undergraduate, while students of higher degrees are known as graduates. In some other educational systems and subjects, under-graduate education is post-secondary education up to the level of a master's degree; this is the case for some science courses in Britain and some medicine courses in Europe.
Character-level	Undergra**c**duate **r**ducation is the post-seconda**y** educat**P**ion previou**T**s to the postgrad**a**ate ed**V**ucation. **tI** includ**s** all the aca**y**demic **l**rograms up to the lev**le** of a bac**F**elor's **e**dgree. **F**or examp**le**, in the Un**ti**ed Sta**N**es, an e**n**try leve**S** universi**r**y stude**t** is kn**T**own as an undergra**ud**ate, while stu**p**dents of **i**hgher degree**F** are kno**n**wn as gra**V**duates. **Y**n some other ed**A**ucational sy**I**stems and subje**ck**s, und**re**graduate e**V**ducation is p**W**ost-secondary e**k**ucation up to the le**k**vel of a master**s'** deg**q**ree; this is the cas**k**e for some scienc**Z**e **o**curses in Britai**Y** and some **e**mdicine co**r**uses in **e**Europe.
Word-level	**Baccalaureate teaching** is the post-secondary **educational former** to the **grad teaching**. It **consists** all the **universities agendas** up to the **tier** of a bachelor's **degrees**. **Per examples**, in the United States, an **inlet echelon academia schoolchildren** is **familiar** as **an baccalaureate**, while **student** of **highest diplomas** are **renowned** as **graduation**. **At** some other **teaching schemes** and **topic**, **baccalaureate schooling** is post-secondary **tuition** up to the **echelon** of a master's **degrees**; this is the **lawsuit** for some **veda cours** in **British** and some **pharmaceuticals cours** in **Eu**.

Table 3: Example perturbed text pairs.

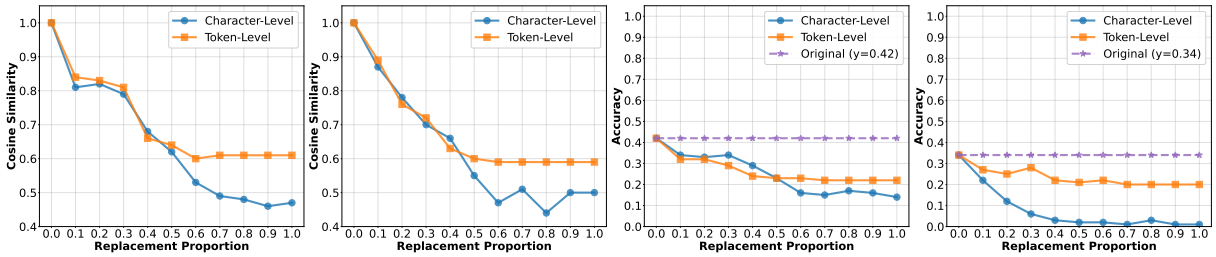
C Additional Analysis Results

We present supplementary analysis results on two additional datasets: ParaRel (Elazar et al., 2021) and GSM8K (Cobbe et al., 2021), following the same experimental setup described in Section 3. As shown in Figure 7, the observations are consistent with those reported in the main text, suggesting that our findings generalize across a broader range of downstream tasks.

D Additional Experimental Results

We provide additional experimental results on Gemma-2-2B layers 5 and 19, and Qwen-0.5B layer 12 to assess the generalizability of our findings across different network depths and model families. Figure 8 and Figure 9 present results on layers 5 and 19 of ReLU SAEs trained on Gemma-2-2B, corresponding to earlier and later representational stages of the model, respectively. The sustained performance gains observed across both shallow and deep layers underscore the robustness and cross-layer generalizability of our strategy.

Figure 10 reports results on layer 12 of Qwen-0.5B. Despite the architectural and scale differences between Qwen-0.5B and Gemma-2-2B, RoSAE exhibits qualitatively similar behavior, suggesting that the benefit of contrastive training in reducing feature splitting is not specific to a single model family.



(a) Similarity on ParaRel (b) Similarity on GSM8K (c) Accuracy on ParaRel (d) Accuracy on GSM8K

Figure 7: Analysis results on ParaRel and GSM8K.

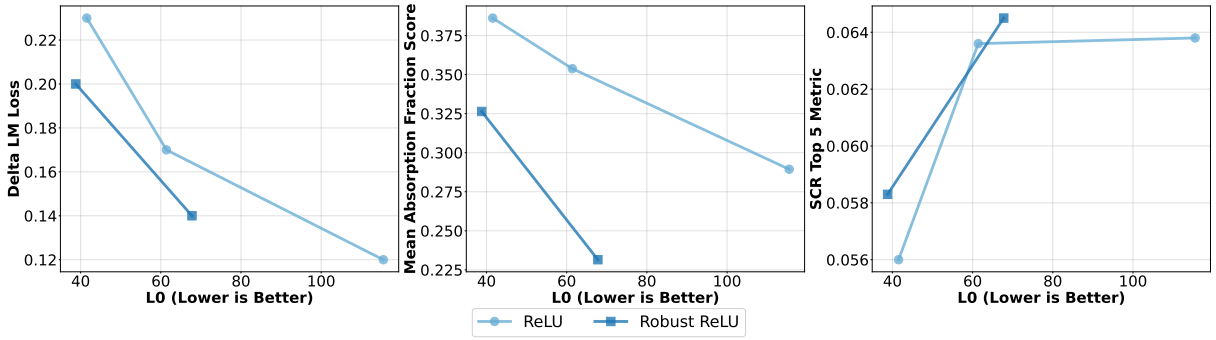


Figure 8: Main experimental results on layer 5 of Gemma-2-2B across varying L_0 sparsity levels.

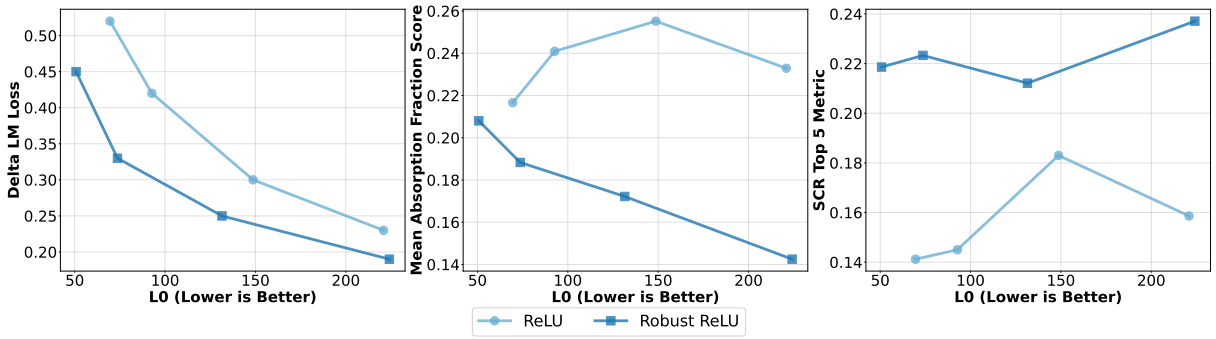


Figure 9: Main experimental results on layer 19 of Gemma-2-2B across varying L_0 sparsity levels.

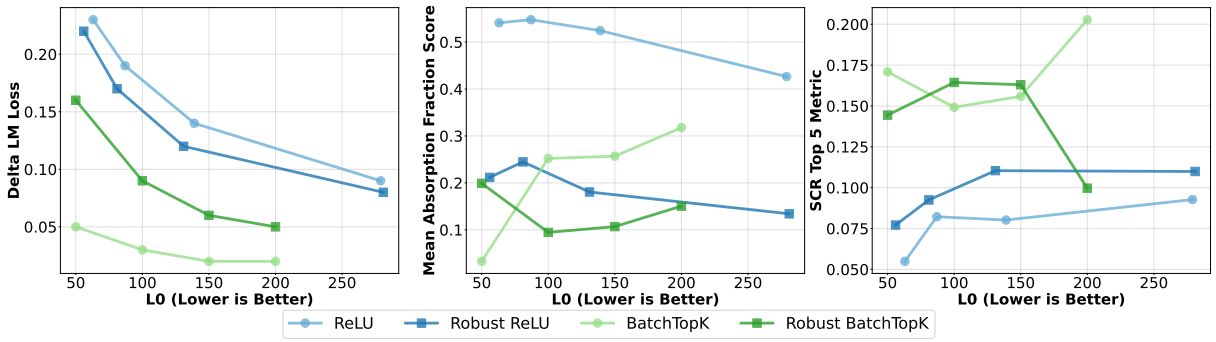


Figure 10: Main experimental results on layer 12 of Qwen2.5-1.5B across varying L_0 sparsity levels.