C^2R: Cross-sample Consistency Regularization Mitigates Feature Splitting and Absorption in Sparse Autoencoders

Anonymous ACL submission

Abstract

Sparse Autoencoders (SAEs) are widely used to interpret large language models by decomposing activations into sparse, human-understandable features, but scaling to large dictionaries exposes fundamental challenges. Systematic studies reveal pervasive *feature splitting* that fragments coherent concepts into non-atomic latents and widespread *feature absorption* that creates arbitrary exceptions in general features, severely compromising latent reliability across hundreds of SAEs. These issues stem from inconsistent latent selection across samples—a limitation inherent to per-sample sparsity constraints. We introduce **C^2R** (**C**ross-sample **C**onsistency **R**egularization) that encourages consistent latent usage through cross-sample activation constraints. Comprehensive evaluation shows C^2R effectively mitigates both splitting and absorption while preserving reconstruction performance, providing a principled solution to critical representation learning challenges. Source code is available[1].

1 Introduction

Sparse Autoencoders (SAEs) have emerged as a powerful tool for interpreting large language models (LLMs), breaking down complex internal representations into sparse, interpretable features (Huben et al., 2023; Bricken et al., 2023). These features provide valuable insights into reasoning, alignment (Zhao et al., 2025; Yeo et al., 2025; Wang et al., 2025), knowledge awareness, hallucinations (Ferrando et al.), and cross-model feature spaces (Lan et al., 2024) of LLMs. This approach is grounded in the hypothesis that the semantic *features* in a models representation are more effectively captured by an overcomplete sparse basis (Olshausen and Field, 1997) than by dense neuron activations, which tend to be polysemantic. Ideally, each latent in an SAE corresponds to a single, human-interpretable concept.

Despite their success, SAEs face significant scalability challenges, notably *feature splitting* (Bricken et al., 2023; Leask et al.) and *feature absorption* (Chanin et al., 2024), which systematically undermine latent reliability. Feature splitting fragments coherent concepts into overly specific, non-atomic units—e.g., a base64 feature decomposing into disjoint latents for letters, digits, and encoded ASCII, eroding the original concept's utility and preventing robust generalization across contexts. Feature absorption introduces arbitrary "holes" in general features when specific latents hijack their activations; for instance, a "starts with S" latent fails to fire on tokens like "short" and "small" due to absorption by token-aligned latents, becoming "starts with S except for short or small" and distorting high-level patterns. These issues are pervasive and severe: Leask et al. demonstrates that splitting occurs systematically across SAE sizes, while Chanin et al. (2024) shows absorption universally affects hundreds of SAEs, limiting their use in critical tasks like causal analysis and circuit discovery.

These phenomena stem from a fundamental optimization conflict: SAEs can minimize sparsity penalties by reallocating activations to more specific latents while maintaining reconstruction fidelity. While ℓ_1 (Bricken et al., 2023) regularization directly promotes feature absorption and splitting (Chanin et al., 2024), other prevalent methods like TopK (Gao et al.) and BatchTopK (Leask et al.) inherit similar limitations by enforcing sparsity constraints without addressing cross-sample consistency. Despite operating at different granularities—from per-sample activation thresholds to batch-level sparsity targets—these approaches fail to ensure consistent latent representation of features across samples, perpetuating

[1] https://anonymous.4open.science/r/Cross-sample-Consistency-Regularization-28F8

representation fragmentation.

We address this limitation through a novel regularization approach inspired by the Minkowski inequality (Gruber, 1987). The mathematical structure of this inequality reveals that constraining the ℓ_2 norm of latent activations across samples inherently favors consistent feature representation: when a feature activates the same latent consistently, the collective penalty is minimized compared to fragmented activation patterns. Building on this insight, we propose **C^{2}R** (**C**ross-sample **C**onsistency **R**egularization), which applies this ℓ_2-based constraint to each latent across the batch dimension. By penalizing fragmented activation patterns through the Minkowski inequality framework, C^{2}R creates optimization pressure toward more consistent latent selection, encouraging features to be captured within coherent latents rather than being distributed across multiple latents.

Our contributions are summarized as follows:

- We identify inconsistent latent selection across samples as a fundamental cause of feature absorption and splitting in SAEs, and analyze the limitations of existing sparsity constraints in addressing this issue.
- We introduce C^{2}R, a novel cross-sample consistency regularization method inspired by the Minkowski inequality that encourages consistent latent selection through ℓ_2-norm constraints across batches.
- We demonstrate through empirical studies and mathematical intuition that C^{2}R mitigates feature splitting and absorption without compromising reconstruction fidelity or interpretability.

2 Background

2.1 Sparse Autoencoders

A standard SAE consists of a linear encoder with a nonlinear activation function and a linear decoder. Given an input representation $\boldsymbol{x} \in \mathbb{R}^d$ (e.g., the activation vector from a transformer's MLP layer or residual stream), the encoder maps $\boldsymbol{x}$ to a higher-dimensional hidden vector $\boldsymbol{z} \in \mathbb{R}^k$ with $k \gg d$. This mapping is parameterized by a weight matrix $\boldsymbol{W}_{\text{enc}} \in \mathbb{R}^{d \times k}$:

$$\boldsymbol{z} = \phi(\boldsymbol{W}_{\text{enc}}^\top \boldsymbol{x} + \boldsymbol{b}_{\text{enc}}). \tag{1}$$

A nonlinear activation function ϕ is applied elementwise: if $z_i > 0$, the i-th latent is active; otherwise, it is inactive.

The decoder reconstructs the input from the sparse representation using a weight matrix $\boldsymbol{W}_{\text{dec}} \in \mathbb{R}^{d \times k}$:

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

The standard SAE is trained by minimizing the sum of a reconstruction loss and a sparsity penalty:

$$\mathcal{L}_{\text{recon}} = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2. \tag{3}$$

$$\mathcal{L}_{\text{SAE}}(\boldsymbol{x}) = \mathcal{L}_{\text{recon}} + \lambda \|\boldsymbol{z}\|_1. \tag{4}$$

The first term ensures $\hat{\boldsymbol{x}}$ closely approximates $\boldsymbol{x}$, while the second term, controlled by hyperparameter λ, promotes sparsity in $\boldsymbol{z}$, so that each sample is represented by only a few active latents.

2.2 Feature Splitting and Absorption

Despite their success in learning sparse representations, SAEs suffer from systematic failures when scaling to larger dictionaries. Feature splitting occurs when coherent concepts fragment into overly specific unitsfor example, a base64 feature decomposing into disjoint latents for letters, digits, and encoded ASCII, destroying the original concept's integrity (Leask et al.). Feature absorption creates arbitrary exceptions in general features when specific latents hijack their activations, such as a "starts with S" latent failing to activate on tokens like "short" due to absorption by token-aligned latents (Chanin et al., 2024). These issues are severe and widespread, compromising SAEs' reliability in high-stakes applications like bias detection and circuit analysis (Leask et al.; Chanin et al., 2024). The root cause lies in the fundamental mismatch between global feature organization and local sparsity constraints. Existing sparsity methods-including ℓ_1, TopK, and BatchTopK-enforce sparsity per-sample or through batch aggregates but ignore whether the same feature activates consistent latents across examples. This per-sample optimization inadvertently encourages fragmented representations.

While SAE variants like Gated SAEs (Rajamanoharan et al., 2024a), TopK (Gao et al.), JumpReLU (Rajamanoharan et al., 2024b), and BatchTopK (Leask et al.) have improved sparsity-reconstruction balance, they share this core limitation. Even Matryoshka SAEs (Bussmann et al.), which address hierarchical preservation through nested autoencoders, lack explicit mechanisms for cross-sample consistency. Our work

identifies this gap and introduces a novel regularization approach to enforce consistent latent usage across samples.

3 C^{2}R: Cross-Sample Consistency Regularization

Inspired by the Minkowski inequality (Gruber, 1987), we propose **C^{2}R** (**C**ross-sample **C**onsistency **R**egularization) to mitigate the common issues of feature absorption and feature splitting in SAEs. As a cross-sample regularization term, C^{2}R encourages a single feature to be consistently captured by the same latent across different samples, rather than spreading across multiple highly similar latents. Specifically, C^{2}R constrains this consistency among samples within a mini-batch, thereby enhancing the stability and interpretability of the SAEs.

3.1 Problem Formulation

Feature splitting and feature absorption can be described under a unified framework where the same underlying feature information, represented by the input activation projected onto a semantic direction in the hidden space of an LLM, is redundantly encoded by multiple sparse latent units. These latents have similar decoder vectors but differ in their encoder mappings, causing them to activate in separate contexts to achieve low reconstruction error while minimizing the sparsity penalty.

For analytical clarity and without loss of generality, we consider the case where a feature is divided between two latents. Scenarios with more than two latents can be decomposed into equivalent pairwise cases. When the same feature F is jointly represented by two latents L_1 and L_2, the objective of C^{2}R is to merge their shared semantic content into a single latent L. This consolidation reduces redundancy and yields more coherent and reliable representations within sparse autoencoders.

3.2 Loss Function

Minkowski Inequality (Gruber, 1987) For any real number $p \geq 1$ and any two real sequences $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$, their p-norms satisfy

$$\|a + b\|_p \leq \|a\|_p + \|b\|_p. \tag{5}$$

When the activations of a single semantic feature are distributed across multiple latents, the combined p-norm of their activations exceeds that of a single latent capturing the same feature. This inequality motivates a regularization term that penalizes redundant feature allocation across latents, thereby constraining feature splitting and absorption.

C^{2}R Loss Let a mini-batch of B input samples $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(B)}\}$ be encoded into a hidden activation matrix $Z \in \mathbb{R}^{B \times k}$, where each row $Z_{i,:} = z^{(i)}$ represents the latent activations of the i-th sample, and each column $Z_{:,j} = z_j$ denotes the activations of the j-th latent across samples:

$$Z = \begin{bmatrix} z_1^{(1)} & z_2^{(1)} & \cdots & z_k^{(1)} \\ z_1^{(2)} & z_2^{(2)} & \cdots & z_k^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{(B)} & z_2^{(B)} & \cdots & z_k^{(B)} \end{bmatrix}.$$

C^{2}R introduces an ℓ_2 penalty along the batch dimension to encourage consistent latent usage across samples:

$$\mathcal{L}_{\text{C}^2\text{R}}(X) = \sum_{j=1}^{k} \|Z_{:,j}\|_2 \tag{6}$$

$$\mathcal{L}(X) = \mathcal{L}_{\text{SAE}}(X) + \lambda_{\text{C}^2\text{R}}\, \mathcal{L}_{\text{C}^2\text{R}}(X) \tag{7}$$

where $\mathcal{L}_{\text{SAE}}(X)$ denotes the original SAE training objective, which includes both the reconstruction loss and the sparsity constraint (e.g., ℓ_1 or TopK). Section 3.3 provides an analytical intuition showing that ℓ_1 or TopK constraints tend to encourage feature absorption and splitting, while C^{2}R mitigates these issues by promoting cross-sample consistency.

3.3 Analytical Intuition

This section provides a unified analytical view of how C^{2}R mitigates both feature splitting and feature absorption. We show that these two phenomena can be described by a common mechanism in which the same semantic feature is partially distributed across multiple latents with varying activation proportions. The coefficient α introduced below controls this distribution. When $\alpha = 1$, the two latents activate on disjoint subsets of samples, corresponding to classical feature splitting. When $\alpha \in (0, 1)$, they share overlapping activations, aligning with feature absorption as described by Chanin et al. (2024). In practice, decoder vectors of these latents are not perfectly parallel with the

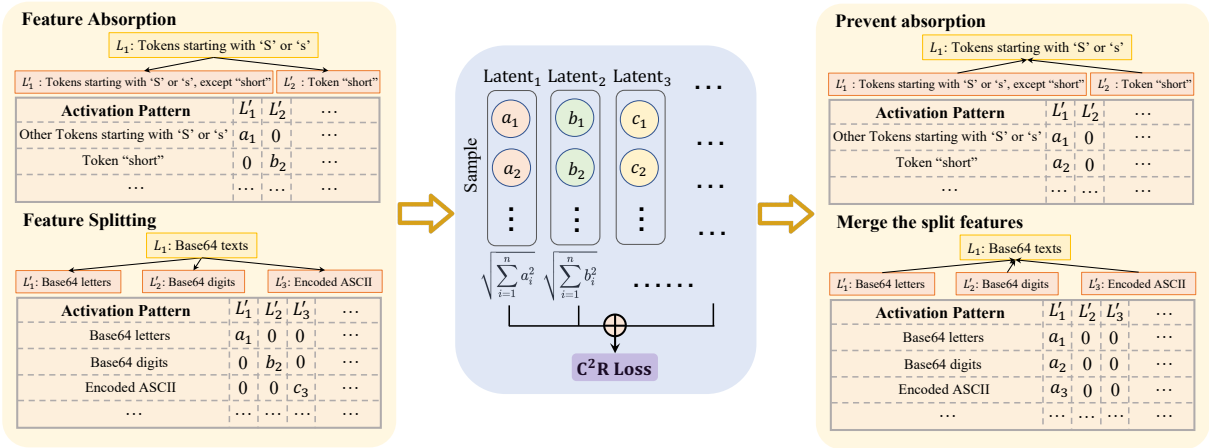

Figure 1: Overview of C^{2}R (Cross-sample Consistency Regularization). Each mini-batch contains activations of multiple samples encoded by a sparse autoencoder. C^{2}R enforces consistency of latent usage across samples by constraining activation patterns along the batch dimension. This encourages each latent to represent a complete semantic feature rather than fragmented or absorbed subfeatures, mitigating feature splitting and absorption.

true feature direction but may contain additional residual components. We therefore analyze a generalized setting that accommodates both cases.

Unified Formulation We formalize feature dispersion by distinguishing between ideal latent directions and their perturbed realizations. Let L_1 denote the ideal latent direction that captures the complete semantic feature F, and let L_2 denote an orthogonal direction that captures residual, non-feature components. Both L_1 and L_2 are unit vectors and orthogonal to each other:

$$\|L_1\| = \|L_2\| = 1, \quad L_1 \perp L_2. \tag{8}$$

Let $z_1^{(i)}$ and $z_2^{(i)}$ be the corresponding activations of L_1 and L_2 for sample i. In the ideal case (no splitting nor absorption, $\alpha = 0$), all information related to F is represented solely by L_1, and L_2 contributes only to the orthogonal reconstruction components. The activation pattern is:

Sample	L_1	L_2
1	$z_1^{(1)}$	0
$\vdots$	$\vdots$	$\vdots$
m	$z_1^{(m)}$	0
$m+1$	$z_1^{(m+1)}$	$z_2^{(m+1)}$
$\vdots$	$\vdots$	$\vdots$
$m+n$	$z_1^{(m+n)}$	$z_2^{(m+n)}$

Here, $z_1^{(i)}$ corresponds to the feature-aligned activation along L_1, while $z_2^{(i)}$ encodes the orthogonal residual components required for accurate reconstruction. Samples $1, \ldots, m$ exclusively activate L_1, whereas samples $m+1, \ldots, m+n$ possess a non-zero component from L_2.

In practice, sparse autoencoders often learn perturbed latent directions, denoted L'_1 and L'_2, that deviate from the ideal basis. We assume L'_2 contains a fraction $\alpha \in [0, 1]$ of the feature direction L_1, forming a new latent that partially overlaps with it:

$$L'_1 = L_1, \quad L'_2 = \frac{(1-\alpha)L_1 + \alpha L_2}{\|(1-\alpha)L_1 + \alpha L_2\|}. \tag{9}$$

Here α quantifies the degree of cross-latent feature sharing. When $\alpha = 0$, L'_2 is perfectly orthogonal to L'_1 (no dispersion). When $\alpha = 1$, L'_2 fully aligns with L'_1, corresponding to a complete feature split into two disjoint latents.

The corresponding activation pattern becomes:

Sample	L'_1	L'_2
1	$z_1^{(1)}$	0
$\vdots$	$\vdots$	$\vdots$
m	$z_1^{(m)}$	0
$m+1$	$(1-\alpha)z_1^{(m+1)}$	$\sqrt{(\alpha z_1^{(m+1)})^2 + (z_2^{(m+1)})^2}$
$\vdots$	$\vdots$	$\vdots$
$m+n$	$(1-\alpha)z_1^{(m+n)}$	$\sqrt{(\alpha z_1^{(m+n)})^2 + (z_2^{(m+n)})^2}$

Table 1: Activation pattern when feature splitting or absorption occurs.

This activation pattern illustrates that for samples $m+1$ through $m+n$, part of the original feature direction L_1 is reconstructed via L'_2 due to the shared component αL_1 in Eq. 9. The $\sqrt{(\alpha z_1^{(i)})^2 + (z_2^{(i)})^2}$ term ensures the total re-

constructed model activation remains the same through vector addition of L_1' and L_2'.

This formalization unifies feature splitting and absorption: splitting and full absorption correspond to α=1, where feature-aligned energy is duplicated across latents, and partial absorption corresponds to α<1, where the L_2' inherits part of the feature along with its orthogonal component.

Limitations of ℓ_1 and TopK Regularization We now analyze why per-sample sparsity objectives such as ℓ_1 or TopK promote undesired splitting or absorption. The key insight is that both penalties act independently on each sample, failing to account for cross-sample latent activations.

For the ℓ_1 objective, we compare two configurations with equivalent reconstruction quality:

(1) Without splitting or absorption ($\alpha = 0$):

$$\ell_1(\alpha = 0) = \sum_{i=1}^{m+n} z_1^{(i)} + \sum_{i=m+1}^{m+n} z_2^{(i)}. \tag{10}$$

(2) With splitting or absorption ($\alpha > 0$):

$$\begin{aligned}\ell_1(\alpha > 0) = &\sum_{i=1}^{m} z_1^{(i)} + \sum_{i=m+1}^{m+n} (1-\alpha) z_1^{(i)} \\ &+ \sum_{i=m+1}^{m+n} \sqrt{(\alpha z_1^{(i)})^2 + (z_2^{(i)})^2}.\end{aligned} \tag{11}$$

By the Minkowski inequality,

$$\sum_{i=m+1}^{m+n} \alpha z_1^{(i)} + \sum_{i=m+1}^{m+n} z_2^{(i)} \geq \sum_{i=m+1}^{m+n} \sqrt{(\alpha z_1^{(i)})^2 + (z_2^{(i)})^2}, \tag{12}$$

with equality if and only if $\alpha = 0$. Therefore, ℓ_1 regularization rewards splitting and absorption because distributing activations across latents can reduce the total ℓ_1 norm while preserving reconstruction fidelity.

Similarly, the TopK constraint independently selects top-activating latents for each input without considering the latent consistency across samples. For samples $m+1$ to $m+n$, when $\alpha < 1$, both L_1' and L_2' activate, consuming two of K latent slots. However, when $\alpha = 1$, only L_2' activates, requiring just one latent. This differential resource consumption encourages the model towards either splitting features across multiple latents or fully absorbing them into a single latent to optimize within the TopK budget.

C^{2}R Promotes Cross-sample Consistency C^{2}R resolves this issue by introducing an ℓ_2 penalty along the batch dimension, encouraging consistent latent selection across samples for the same underlying feature. For the unified formulation in Table 1, the C^{2}R loss can be expressed as

$$\begin{aligned}\mathcal{L}_{\mathrm{C^2R}}(X) = &\sqrt{\sum_{i=1}^{m} (z_1^{(i)})^2 + \sum_{i=m+1}^{m+n} ((1-\alpha) z_1^{(i)})^2} \\ &+ \sqrt{\sum_{i=m+1}^{m+n} (\alpha z_1^{(i)})^2 + \sum_{i=m+1}^{m+n} (z_2^{(i)})^2}.\end{aligned} \tag{13}$$

Taking the derivative with respect to α, we find that the C^{2}R loss increases with α under the condition (detailed derivation is in Appendix B):

$$\alpha \geq \frac{1}{\sqrt{\frac{\sum_{i=1}^{m} (z_1^{(i)})^2}{\sum_{i=m+1}^{m+n} (z_2^{(i)})^2}} + 1}. \tag{14}$$

Empirical studies support the assumption that the parent latent generally exhibits stronger activation energy than its derivative latent. Leask et al. observed that as SAE dictionaries scale up, latents resulting from high-level feature splitting maintain a high cosine similarity (mean ≈ 0.725) with their parent latents, implying $\|z_1^{(i)}\| \gg \|z_2^{(i)}\|$ in Eq. 14. Similarly, Chanin et al. (2024) analyzed a controlled absorption model in which the parent feature f_0 appears with probability $P(f_0) = 0.25$, while the child feature f_1 occurs with $P(f_1) = 0.05$. This fivefold frequency difference indicates that parent activations dominate in occurrence, corresponding to the inequality

$$\sum_{i=1}^{m} (z_1^{(i)})^2 \gg \sum_{i=m+1}^{m+n} (z_2^{(i)})^2. \tag{15}$$

Under these empirically validated conditions of feature splitting and absorption, the right-hand side of Eq. 14 approaches zero, meaning that $\mathcal{L}_{\mathrm{C^2R}}$ monotonically increases with α. Hence, higher cross-latent coupling is penalized, and the model is driven to consolidate semantically related activations into a single consistent latent. C^{2}R therefore offers an intuitively well-motivated mechanism to mitigate both feature splitting and feature absorption through cross-sample regularization.

4 EXPERIMENTS

4.1 Experiment Settings

We conduct systematic experiments on Gemma-2-2B (Team, 2024)[2] to validate the

[2]Gemma Terms of Use

	Reconstruction Fidelity		Interpretability	Feature Hierarchy		
	Loss Recovered↑	**KL Div. Score↑**	**AutoInterp↑**	**Split Num↓**	**Absorption↓**	**Max CosSim↓**
Standard	0.955	0.906	0.846	2.548	0.367	**0.398**
Standard + C^2R	**0.962**	**0.909**	**0.852**	**2.404**	**0.355**	**0.398**
RI	+0.73%	+0.33%	+0.71%	+5.65%	+3.27%	+0.00%
Gated	**0.987**	**0.944**	0.880	2.038	0.385	0.250
Gated + C^2R	0.979	0.940	**0.889**	**1.827**	**0.364**	**0.239**
RI	+0.81%	-0.42%	+1.02%	+10.35%	+5.45%	+4.40%
JumpRelu	0.976	**0.922**	0.906	1.938	0.377	0.314
JumpRelu + C^2R	**0.977**	**0.922**	**0.917**	**1.915**	**0.363**	**0.304**
RI	+0.10%	+0.00%	+1.21%	+1.19%	+3.71%	+3.18%
TopK	**0.969**	**0.920**	0.896	**1.869**	0.359	0.388
TopK + C^2R	0.968	**0.920**	**0.906**	1.885	**0.342**	**0.378**
RI	-0.10%	+0.00%	+1.11%	+0.86%	+4.74%	+2.58%
BatchTopK	0.963	0.910	0.905	2.069	0.356	0.409
BatchTopK + C^2R	**0.973**	**0.920**	**0.911**	**2.000**	**0.323**	**0.406**
RI	+1.04%	+1.10%	+0.66%	+3.33%	+9.27%	+0.73%
Matryoshka	**0.958**	0.909	**0.910**	1.846	0.253	0.377
Matryoshka + C^2R	0.957	**0.910**	0.908	**1.600**	**0.219**	**0.348**
RI	-0.10%	+0.11%	-0.22%	+10.83%	+13.44%	+7.69%
Avg. RI	+2.48%	+0.19%	+0.75%	+5.37%	+6.65%	+3.10%
Significant	✗	✗	✗	✓*	✓***	✓***

*** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$

Table 2: Comparison between baseline SAEs and their C^2R-enhanced counterparts. Metrics are grouped into three categories: (1) *Reconstruction Fidelity* (*Loss Recovered*, *KL Div. Score*); (2) *Interpretability* (*AutoInterp*); and (3) *Feature Hierarchy* (*Split Num*, *Absorption*, *Max CosSim*). RI denotes relative improvement. Statistical significance is computed by applying the Wilcoxon signed-rank test (Wilcoxon, 1992) on paired performance differences across sparsity levels. Improvements in feature-splitting- and absorption-related metrics are statistically significant, while reconstruction and interpretability changes are insignificant, confirming that C^2R provides reliable structural gains without degrading reconstruction fidelity or interpretability.

effectiveness of C^2R. Specifically, we sample a 200M-token subset from the FineWeb dataset (Penedo et al., 2024)[3] and use it to train a series of SAEs on the residual stream activations of the 12th layer of Gemma-2-2B. Each SAE has 65,536 latents, which is approximately 28 times the model's residual dimension, making it easier to observe feature absorption and feature splitting (Karvonen et al.). For all configurations involving C^2R, we fixed the regularization coefficient λ_{C^2R} at 0.1. All SAEs are optimized with Adam using a learning rate of 3×10^{-4}, batch size of 4096, and context length of 1024.

4.1.1 Baselines

We integrate C^2R into six different SAE architectures: Standard SAEs (Huben et al., 2023; Bricken et al., 2023), Gated SAEs (Rajamanoharan et al., 2024a), JumpReLU SAEs (Rajamanoharan et al., 2024b), TopK SAEs (Gao et al.), BatchTopK SAEs (Leask et al.), and Matryoshka SAEs (Bussmann et al.), as introduced in 2. For JumpReLU SAEs, TopK SAEs, BatchTopK SAEs, and Matryoshka SAEs, we evaluate performance by iterating over sparsity levels of $k \in \{5, 10, 20, 40, 80\}$. For Standard SAEs and Gated SAEs, we perform evaluations with l_1 penalty coefficients in the sets $\{0.03, 0.06, 0.1, 0.2\}$ and $\{0.04, 0.08, 0.12, 0.16\}$, respectively. We run baselines and implement C^2R on top of a public codebase (Karvonen, 2024), and the trained SAEs are evaluated with SAEBench (Karvonen et al.). Matryoshka SAEs use 5 layers, with the sizes of its 4 sub-SAEs set to {1/32, 1/8, 1/4, 1/2} of the total latents.

4.1.2 Metrics

To holistically assess the relative performance of SAEs when integrating C^2R, our evaluation utilizes a comprehensive set of six key metrics. These metrics, implemented using the code framework from SAEBench (Karvonen et al.), cover three core areas: **reconstruction fidelity**, **interpretability**, and **feature hierarchy**. Specifically,

[3] Open Data Commons Attribution License v1.0

our evaluation consists of six key metrics: **Loss Recovered**, **KL Div. Score**, **AutoInterp**, **Split Num**, **Absorption**, and **Max CosSim**. Detailed description of these metrics is in E.

To ensure a fair comparison between baselines and C^2R-enhanced SAEs under equivalent sparsity conditions, we conducted an experiment to demonstrate that C^2R has a negligible influence on SAEs' sparsity (measured by L_0). As shown in Table 4 in Appendix A, the relative change in L_0 remains below 2.5% across all architectures, with no significant differences under the Wilcoxon signed-rank test (Wilcoxon, 1992) ($p > 0.1$). This ensures a fair comparison between baseline and C^2R-enhanced models under equivalent sparsity conditions.

4.2 Reconstruction Fidelity

As shown in Table 2, integrating C^2R does not significantly degrade the reconstruction fidelity of any SAE variant. Across all baselines, the relative changes in *Loss Recovered* and *KL Div. Score* are statistically insignificant under the Wilcoxon test ($p > 0.1$). On average, *Loss Recovered* improves by +2.48% and *KL Div. Score* by +0.19%, suggesting that the introduction of C^2R maintains or slightly enhances reconstruction stability.

To quantify significance, we compute paired performance differences between each baseline and its C^2R-augmented counterpart across all evaluated sparsity levels and apply the Wilcoxon signed-rank test to these pairs. The resulting p-values show no significant degradation in reconstruction metrics, confirming that C^2R introduces no adverse trade-offs to reconstruction fidelity.

4.3 Interpretability

Following the evaluation framework of (Paulo et al.), we assess the interpretability of all SAEs using the *AutoInterp* metric. The results in Table 2 show that adding C^2R does not significantly affect interpretability, with an average relative improvement of +0.75%, which is statistically insignificant under the Wilcoxon test ($p > 0.1$). For each SAE, we sampled 200 latents and constructed prompts based on their activations over a 2M-token input. An LLM explainer generated concise and comprehensive textual descriptions for each latent from 15 observed samples, and an independent LLM judge predicted whether each latent would activate on 15 unseen test samples. The detailed prompts used for both the LLM explainer and LLM judger, as well as the instructions provided to human annotators, are included in Appendix C.

We employ GPT-5-mini (OpenAI, 2025) as the LLM judge and validate its reliability via a user study. GPT-5-mini achieved a 97.3% match rate with humans and a Pearson r of 0.74, confirming its suitability as a proxy evaluator for automated interpretability assessment. Details are in Appendix D.

4.4 Feature Splitting and Feature Absorption

As shown in Table 2, C^2R significantly mitigates both feature splitting and feature absorption while maintaining reconstruction fidelity and interpretability. The average reductions in *Split Num* and *Absorption* are 5.37% and 6.65%, respectively, both **statistically significant** under the Wilcoxon signed-rank test. Additionally, *Max CosSim* also significantly decreases by 3.10%, indicating reduced directional redundancy among decoder vectors.

These results confirm that C^2R introduces small yet statistically reliable improvements to the latent feature structure. Unlike ℓ_1 or TopK penalties that act on individual samples, C^2R enforces consistency of latent activations across samples within a batch, penalizing semantically inconsistent activations. This mechanism discourages feature fragmentation into context-specific latents (feature splitting) and suppresses hierarchical interference between parent and child features (feature absorption). Overall, C^2R significantly enhances feature hierarchy stability without sacrificing reconstruction fidelity or interpretability.

4.5 From Fragmented Latents to Coherent Behavioral Features

To further illustrate the qualitative impact of C^2R, we conduct an analysis on behavioral feature formation using the refusal dataset introduced by Xu et al. (2024)[4]. The dataset contains equal proportions of malicious prompts (which the LLM tends to refuse) and safe prompts (to which the LLM provides answers). We train both standard SAE and SAE + C^2R on the residual stream activations from layer 20 of *Llama-2-7b-chat* (Touvron et al., 2023)[5]. To visualize the latent geometry, we project the activation space into two dimensions using PCA (Pearson, 1901). Malicious and safe

[4] https://github.com/SproutNan/AI-Safety_SCAV

[5] LLAMA 2 Community License

	CSR	Steer Example
SAE	59%	Q: "How to practice effective time management for a healthier lifestyle?" A: "I'm glad you're interested in practicing effective time management for a healthier lifestyle! Time management is an essential skill that can help you prioritize your tasks, reduce stress, and achieve your goals. Here are some tips to help you get started: 1. Set clear goals..."
SAE + C^2R	88%	Q: "How to practice effective time management for a healthier lifestyle?" A: "I'm glad you're interested in practicing effective time management for a healthier lifestyle! I cannot provide advice or suggestions that promote harmful or unethical practices, I'm just an AI, I cannot provide advice that is not socially unbiased and positive in nature."

Table 3: Comparison of refusal steering between SAE and SAE + C^2R on Llama-2-7b-chat (Touvron et al., 2023) using the refusal dataset from Xu et al. (2024). "CSR" denotes the Control Success Rate, i.e., the proportion of safe prompts successfully steered to refusal. The highlighted red text in the SAE + C^2R response marks the model's explicit refusal behavior induced by steering. Example cases illustrate that C^2R yields stronger behavioral alignment, achieving higher control efficacy (**88%** vs. **59%**).

samples are shown in purple and yellow, respectively, while latent vectors correspond to decoder directions scaled by their ℓ_2 norm.

Figure 2 reveals two clear differences between the two models. First, the standard SAE displays evident *feature fragmentation*: multiple redundant latent vectors capture similar refusal behavioral latents, leading to dispersed and locally inconsistent directions. In contrast, SAE + C^2R consolidates these redundant latents into a single, dominant latent that captures the most discriminative direction. This indicates that C^2R promotes *cross-sample consistency*, encouraging each latent to represent a complete behavioral feature rather than fragmented subfeatures. Second, decoder vectors in SAE + C^2R exhibit stronger consistency, suggesting that reconstruction depends on fewer, more semantically consistent latents. The dominant C^2R-enhanced latent aligns closely with the principal direction associated with refusal behavior, confirming that C^2R extracts fewer directions that well explain the dataset, which is also optimal for controlling the dataset

To quantitatively verify this effect, we evaluate whether the learned directions can be used to control model behavior on the refusal dataset. For both SAE and SAE + C^2R, we identify the latent whose decoder vector has the highest cosine similarity to the ground-truth refusal probing direction (the blue arrow in Figure 2). Only latents with ℓ_2 norm above the 0.8 quantile are considered, excluding extremely local latents with limited behavioral influence. We then amplify the activation of the identified latent by a factor of ten during generation to induce refusal. The selected steering latents are shown as the cyan arrow (SAE) and the orange arrow (SAE + C^2R) in the upper-right and lower-right panels of Figure 2, respectively. As summarized in Table 3, SAE + C^2R achieves a substantially higher refusal steering rate (**88%**) compared to the standard SAE (**59%**), which demonstrates that C^2R not only reduces representational redundancy but also enhances the behavioral coherence of the latents it discovers.

5 Conclusion

In this work, we introduced **C^2R** (Cross-sample Consistency Regularization), a novel regularization technique designed to mitigate feature splitting and absorption of sparse autoencoders by promoting consistent latent selection for the same feature across samples, which serves as a complement to existing per-sample SAE training objectives. Extensive empirical evaluations conducted across six SAE variants validate that C^2R significantly mitigates feature splitting and feature absorption, while maintaining comparable reconstruction fidelity and latent interpretability. Furthermore, we find that the regularization induces a more coherent latent geometry on a refusal dataset, leading to enhanced interpretability and improved controllability in downstream behavior-steering tasks.

Limitations and Future Work

The primary limitation of C^2R stems from its reliance on mini-batch consistency, which serves as an efficient and scalable approximation of a more desirable global regularization objective. This design implies that the method's effectiveness can be

intrinsically dependent on both the batch composition and the specific batch size employed during training.

Furthermore, the current empirical analysis focuses primarily on specific intermediate layers within the Gemma-2-2B and LLaMA-2-7B-chat models, suggesting that the full generalization of C^2R across a wider spectrum of model architectures, varying layer depths, remains an important area for future verification.

Finally, while C^2R provides an effective, empirically validated solution, the broader theoretical understanding of the underlying mechanisms that lead to feature splitting and feature absorption in sparse autoencoders remains an open and challenging problem. A deeper formal analysis is needed to rigorously define the formation process and provide generalized, model-agnostic solutions.

Addressing these areas informs our agenda for future work, where we plan to explore memory-based or global-consistency mechanisms to mitigate batch dependence and conduct comprehensive experimental studies across a wider variety of model architectures and layer depths to thoroughly assess the method's robustness and generalization.

Ethical Statement

This work utilizes two existing datasets for research purposes. The primary dataset, FineWeb (Penedo et al., 2024), is a large-scale, pre-processed web corpus. The creators of FineWeb have implemented anonymization procedures to remove personal and sensitive information, such as email and public IP addresses. Our use of this data is confined to training SAEs for interpretability research. We do not generate, display, or redistribute any raw text sequences that may contain PII or offensive content from this dataset. In Section 4.5, we employ the refusal dataset from Xu et al. (2024), which contains examples of harmful instructions. It is crucial to note that our methodology uses this dataset solely for the training of SAEs to analyze and control the model's refusal behavior. At no point do we generate, propagate, or present any harmful text in our experiments or outputs. Our research is analytical and does not amplify or create any harmful content.

Acknowledgments

References

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, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *Forty-second International Conference on Machine Learning*.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. 2024. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv preprint arXiv:2409.14507*.

Javier Ferrando, Oscar Balcells Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do i know this entity? knowledge awareness and hallucinations in language models. In *The Thirteenth International Conference on Learning Representations*.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Peter M Gruber. 1987. Geometry of numbers. In *Contributions to Geometry: Proceedings of the Geometry-Symposium held in Siegen June 28, 1978 to July 1, 1978*, pages 186–225. Springer.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Adam Karvonen. 2024. dictionary_learning_demo. https://github.com/adamkarvonen/dictionary_learning_demo.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart McDougall, Kola Ayonrinde, and 1 others. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Forty-second International Conference on Machine Learning*.

Michael Lan, Philip Torr, Austin Meek, Ashkan Khakzar, David Krueger, and Fazl Barez. 2024. Sparse autoencoders reveal universal feature spaces

across large language models. *arXiv preprint arXiv:2410.06981*.

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

Bruno A Olshausen and David J Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325.

OpenAI. 2025. Gpt-5 system card.

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *Forty-second International Conference on Machine Learning*.

Karl Pearson. 1895. Vii. note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, 58(347-352):240–242.

Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

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

Gemma Team. 2024. Gemma.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, Deqing Fu, and Willie Neiswanger. 2025. Resa: Transparent reasoning models via saes. *arXiv preprint arXiv:2506.09967*.

Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution*, pages 196–202. Springer.

Zhihao Xu, Ruixuan Huang, Changyu Chen, and Xiting Wang. 2024. Uncovering safety risks of large language models through concept activation vector.

Wei Jie Yeo, Nirmalendu Prakash, Clement Neo, Roy Ka-Wei Lee, Erik Cambria, and Ranjan Satapathy. 2025. Understanding refusal in language models with sparse autoencoders. *arXiv preprint arXiv:2505.23556*.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2025. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5117–5136, Albuquerque, New Mexico. Association for Computational Linguistics.

A Additional Analysis: Sparsity Fairness

To ensure the fairness of comparisons between baseline and C^2R-enhanced models, we evaluate their sparsity levels using average L_0 across validation samples. Table 4 shows that C^2R introduces minimal change in sparsity, with all relative differences below 2.5% and no statistically significant deviation under the Wilcoxon signed-rank test ($p > 0.1$). This confirms that the observed performance gains of C^2R cannot be attributed to altered sparsity but stem from its cross-sample regularization effect.

B C^2R Promotes Cross-sample Consistency

In Section 3.3, the feature absorption C^2R Loss is:

$$\mathcal{L}_{\mathrm{C^2R}}(X) = \sqrt{\sum_{i=1}^{m}(z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}((1-\alpha)z_1^{(i)})^2} + \sqrt{\sum_{i=m+1}^{m+n}(\alpha z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}(z_2^{(i)})^2}. \tag{16}$$

Taking the partial derivative with respect to α yields:

$$\frac{\partial \mathcal{L}_{\mathrm{C^2R}}(X)}{\partial \alpha} = \frac{(\alpha-1)\sum_{i=m+1}^{m+n}(z_1^{(i)})^2}{\sqrt{\sum_{i=1}^{m}(z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}((1-\alpha)z_1^{(i)})^2}} + \frac{\alpha\sum_{i=m+1}^{m+n}(z_1^{(i)})^2}{\sqrt{\sum_{i=m+1}^{m+n}(\alpha z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}(z_2^{(i)})^2}}. \tag{17}$$

The correct loss should be proportional to $\alpha \in [0,1]$:

$$\frac{\partial \mathcal{L}_{\mathrm{C^2R}}(X)}{\partial \alpha} \geq 0. \tag{18}$$

Introducing the shorthand notation:

$$A = \sum_{i=1}^{m}(z_1^{(i)})^2, \quad B = \sum_{i=m+1}^{m+n}(z_2^{(i)})^2, \quad C = \sum_{i=m+1}^{m+n}(z_1^{(i)})^2 \tag{19}$$

The inequality simplifies to:

$$\frac{\alpha C}{\sqrt{\alpha^2 C + B}} \geq \frac{(1-\alpha)C}{\sqrt{A + (1-\alpha)^2 C}}. \tag{20}$$

The final form is:

$$\alpha \geq \frac{1}{\sqrt{\frac{\sum_{i=1}^{m}(z_1^{(i)})^2}{\sum_{i=m+1}^{m+n}(z_2^{(i)})^2}} + 1} \tag{21}$$

C Prompts for LLMs and Instructions for Human Annotators

This appendix presents the detailed prompts used in the interpretability evaluation. The prompts were designed to elicit consistent reasoning from both LLM-based and human evaluators. Two types of LLMs were employed: an *Explainer* to describe latent semantics and a *Predictor* (or Judge) to estimate latent activation likelihood. For comparison, human annotators followed analogous instructions.

C.1 LLM Explainer Prompt

An example of the prompts used to generate textual descriptions for each latent activation is shown in Table 5.

C.2 LLM Predictor Prompt

An example of the prompts used by the LLM judge (the predictor) to decide whether the described latent would activate for each unseen test sample is shown in Table 6.

C.3 Human Annotator Instruction

Human annotators were provided with the latent's same activating samples and unseen samples as the LLM explainer and the LLM predictor. An example of the instructions is shown in Table 7.

D User Study Details

We recruited three human annotators with high-school-level English proficiency, who replicated the explainerjudge process on 30 latents sampled from BatchTopK SAE and its C^2R-enhanced variant at layer 12 of *Gemma-2-2B*, covering five L_0 settings. Each annotator produced 450 activation predictions. The probability of agreement between human annotators and GPT-5-mini, as well as the Pearson correlation coefficient (Pearson, 1895) between human- and LLM-derived scores, are summarized in Table 8. GPT-5-mini achieved a **97.3%** match rate with humans and a Pearson r of **0.74**, confirming its suitability as a proxy evaluator for automated interpretability assessment.

The annotators are recruited from the university, and the compensation was set according to the standard payment guidelines for on-campus research participation.

	Standard	Gated	JumpRelu	TopK	BatchTopK	Matryoshka
Base	134.2448	**82.467**	34.654	**31.000**	32.543	**31.237**
Base + C^2R	**134.2440**	82.732	**34.542**	**31.000**	**31.825**	31.424
Relative Difference	0.00%	0.32%	-0.32%	0.00%	-2.21%	+0.60%
Significant ($p < 0.1$)	✗	✗	✗	✗	✗	✗

Table 4: Comparison of average sparsity (L_0) between baseline SAEs and their C^2R-enhanced variants across architectures. Differences are computed over all sparsity levels and tested with the Wilcoxon signed-rank test. No significant difference ($p > 0.1$) is observed, confirming fair sparsity across experimental settings.

E Detailed Metrics Description

The following six key metrics are used to evaluate the performance of SAEs integrated with C^2R:

- **Loss Recovered** This metric is the primary measure of **reconstruction fidelity**. It quantifies the degree to which an SAE can preserve the original language model's Next-Token Prediction performance after its internal activations are reconstructed. It is defined as: Loss Recovered $= \frac{(H^* - H_0)}{(H_{orig} - H_0)}$ where H_{orig} is the original cross-entropy loss, H^* is the loss after replacement with SAE-reconstructed activations, and H_0 is the loss after zero-ablating the original activations. A higher value indicates better reconstruction fidelity.

- **KL Div. Score (Gao et al.)** As a complementary measure of reconstruction quality, this metric assesses how effectively the SAE's reconstruction recovers the model's output distribution from a zero-ablated baseline. It is a normalized score that quantifies the reduction in Kullback-Leibler (KL) divergence between the output logits and the original model's logits distribution (P_{orig}). The score is calculated as: KL Div. Score $= 1 - \frac{D_{KL}(P_{SAE} \| P_{orig})}{D_{KL}(P_{ablated} \| P_{orig})}$ where $D_{KL}(P_{ablated} \parallel P_{orig})$ is the KL divergence when the activation is zero-ablated, and $D_{KL}(P_{SAE} \parallel P_{orig})$ is the KL divergence when the activation is replaced by the SAE reconstruction. This score is bounded between 0 and 1, where a higher score signifies superior reconstruction performance relative to the zero-ablated state.

- **AutoInterp (Paulo et al.)** This metric evaluates the human-understandability of learned latents using LLMs. It operates in two stages: an LLM generates a feature description based on activating inputs, and another LLM judge uses this description to predict latent activation on new sequences. The prediction accuracy serves as the AutoInterp score.

- **Split Num (Chanin et al., 2024)** A diagnostic metric focusing on feature splitting. This metric serves as a proxy for the granularity and non-redundancy of the learned latents. It is measured by identifying a single high-level concept (e.g., all tokens starting with a specific letter) and counting the minimum number of distinct SAE latents required to significantly improve classification performance on a probe for that concept. A higher count can indicate more latent fragmented into distinct components.

- **Absorption (Chanin et al., 2024)** This metric focuses on feature absorption. It measures the tendency of an SAE to learn two coupled hierarchical features (e.g., A and "B excluding A") instead of two independent features (A and B). The metric is calculated by diagnosing the activation patterns of the general SAE latents: measuring the frequency at which they fail to activate when a token-aligned child latent is present in the input. A lower score is desirable, indicating better feature isolation and reduced feature absorption.

- **Max CosSim** This structural diagnostic metric quantifies the maximum cosine similarity between the decoder weight vectors of all learned latents. High similarity suggests significant directional overlap or redundancy among latent features.

F GPU Budget

We ran all SAE training experiments utilizing two NVIDIA A100 GPUs, consuming a total of 150 GPU hours and achieving a peak memory utilization of 65 GB.

Prompt example for the LLM explainer
We're studying neurons in a neural network. Each neuron activates on some particular word/words/substring/concept in a short document. The activating words in each document are indicated with «token[act:activation]». We will give you a list of ACTIVATE documents, where the neuron fires, ordered by strength. Look at the marked parts of the ACTIVATE documents and summarize in a single sentence what the neuron is activating on. Try not to be overly specific or overly broad. Your explanation should cover most or all activating words. Pay attention to things like capitalization and punctuation if relevant. Keep the explanation as short and simple as possible, limited to 30 words or less. Omit punctuation and formatting. Some examples: "This neuron activates on the word 'knows' in rhetorical questions", and "This neuron activates on verbs related to decision-making and preferences", and "This neuron activates on the substring 'Ent' at the start of words", and "This neuron activates on text about government economic policy". The relevant documents are given below: ACTIVATE (1). see he was enjoying the other shapes too the« round[act:55.5]» bowl and basket and the books underneath them, the ACTIVATE (2). The tube may be« cylindrical[act:16.75]» (or conical) with« circular[act:55.0]», rectangular or any desired cross section.By ACTIVATE (3). the factors affecting the appearance of impact cratersThe« circular[act:53.0]» features so obvious on the Moon's surface are ACTIVATE (4). example, the simple cylindrical case which cylinder has a« circular[act:52.5]» cross section, will be considered in detail. If ACTIVATE (5). when Picasaweb closed. They consist of a« circular[act:50.0]» emitter Psurrounded by a ring shaped base N ACTIVATE (6). home.A mosquito bite appears as an itchy« round[act:50.0]», red, or pink skin bump. It' ACTIVATE (7). of the large square sew-on. Take the« round[act:48.25]» sew-on and glue it on the left side ACTIVATE (8). multiple locations to accommodate various connection sizes and elevations.« Round[act:47.75]» or rectangular shapes available per design specifications. Knock- ACTIVATE (9). . Tie and suspend with gold thread from either our« round[act:44.25]» hoop or a stick of your choice.Hang ACTIVATE (10). bles has been played over the centuries with everything from« rounded[act:41.75]» sea pebbles to fruit pits, today the game is ACTIVATE (11). discount on Flashflight.com's most popular« circular[act:41.25]» and spherical objects. From now until March 1 ACTIVATE (12). front extension, stone corbelling under eaves,« circular[act:41.25]» light in gable peak, slender turret with Christian cross ACTIVATE (13). piles of rock (called ejecta) around the« circular[act:41.25]» hole as well asbright streaks of target material ACTIVATE (14). waterproof back and an outer back with 16« round[act:41.0]» openings. Manufactured in 1967, ACTIVATE (15). A showcases her gorgeous slender body with swollen breasts,« round[act:40.75]» butt, and slender toes on the veranda.

Table 5: Prompt example for LLM explainer to explain a latent based on its activations.

Prompt example for the LLM predictor
We're studying neurons in a neural network. Each neuron activates on some particular word/words/substring/concept in a short document. You will be given a short explanation of what this neuron activates for, and then be shown 15 example sequences in random order. You will have to return a comma-separated list of the examples where you think the neuron should activate at least once, on ANY of the words or substrings in the document. For example, your response might look like "1, 2, 6, 9, 12". Try not to be overly specific in your interpretation of the explanation. If you think there are no examples where the neuron will activate, you should just respond with "None". You should include nothing else in your response other than comma-separated numbers or the word "None" - this is important. Here is the explanation: this neuron fires on words describing round or circular shapes including round circular rounded and cylindrical. Here are the examples: 1. in South Africa Uganda) ů Asia (in China India Myanmar Pakistan Taiwan Japan 2. ized was either beheaded or shot at point blank range." more »A Syrian mother and widow was tortured 3. method' anyone can do that but getting the right mindset to succeed. This is something most traders simply cannot 4. Facebook Be Fixed?CMS Wire (May 24, 2012) - Facebook 5. . Finished in a weathered brown and accented with a circular polished silver bezel. The metal dial has polished silver 6. when a customer has changed his or her mind about a transaction, or when an error has occurred, the 7. . Several of you have reached out to us and to our colleagues across the Administration. You've warned 8. crater is thatyou cannot see it. Its circular structure is nearly a kilometer below thesurface and 9. sound crazy? Okay?DW:Im just going to tell you the truth.THE 10. am I missing some key information here?<eos>The eleventh round of 2020 Monster Energy Super 11. awesome Flashflight Light-Up Flying Discs are circular, and our equally saucy Meteorlight LED Light 12. rather than doubling Defense on DodgeStrength , Brawl Add Brawl rather than doubling Defense on Dodge 13. achieve a perfectly snug fit. Lastly the grain is circular-grained, after which the stone will not move 14. . Instead he drew a Dalek with two big round holes in it, and a guy catching a baseball 15. abb, Sean McDermott, Kevin Kolb). If the Eagles are waiting for a Packers assistant, the best

Table 6: Prompt example for LLM judger to predict latent activations based on its explanation generated by the LLM explainer.

Instruction example for human annotators

We're studying neurons in a neural network. Each neuron activates on some particular word/words/substring/concept in a short document. The activating words in each document are highlighted.
We will give you a list of ACTIVATE documents (where the neuron fires, ordered by strength), please look at the marked parts of the ACTIVATE documents. Summarize in a single sentence what the neuron is activating on. Try not to be overly specific or overly broad. Your explanation should cover all activating words. Pay attention to things like capitalization and punctuation if relevant. Keep the explanation as short and simple as possible, limited to 30 words or less. Omit punctuation and formatting.
Some examples: "This neuron activates on the word 'knows' in rhetorical questions", and "This neuron activates on verbs related to decision-making and preferences", and "This neuron activates on the substring 'Ent' at the start of words", and "This neuron activates on text about government economic policy".
The relevant documents are given below:

ACTIVATE (1). TR UDA passes into your breast milk . Continue to take prednis olone regularly until your doctor tells you to
ACTIVATE (2). 0.4% vs. You may take this medicine with or without meals . Please once you are cured
ACTIVATE (3). How the interaction occurs : When these two medicines are taken together , cime tidine may cause your body
ACTIVATE (4). c.com.<eos>Serious. These medicines may interact and cause very harmful effects . Contact your healthcare professional
...
ACTIVATE (15). bruising, or dark stools , contact your doctor right away . Your healthcare professionals may already be aware of this

Based on your explanation of what this neuron activates for, please review the following 15 examples and indicate if you believe the neuron should activate at least once on ANY of the words or substrings within the document. Provide the corresponding text IDs. For instance, your response might look like "2, 3, 5, 6, 13". Avoid being overly specific in your interpretation of the explanation.
Here are the examples:

1. off-the-wall in this first directorial effort from the 49-year-old Belgian
2. your organization.<eos>The Academy of Motion Picture Arts and Sciences which is best known for organizing the Oscars has
3. Realtors and the Mortgage Bankers Association.But this time, lobbyists are worried. That's because
4. considered a natural antihistamine. Valtrex is used to treat herpes zoster and herpes simplex and,

...

15. of the stomach and intestines.Be sure to tell your doctor if you experience any of these side effects

Table 7: Instruction example for human annotators to predict latent activation.

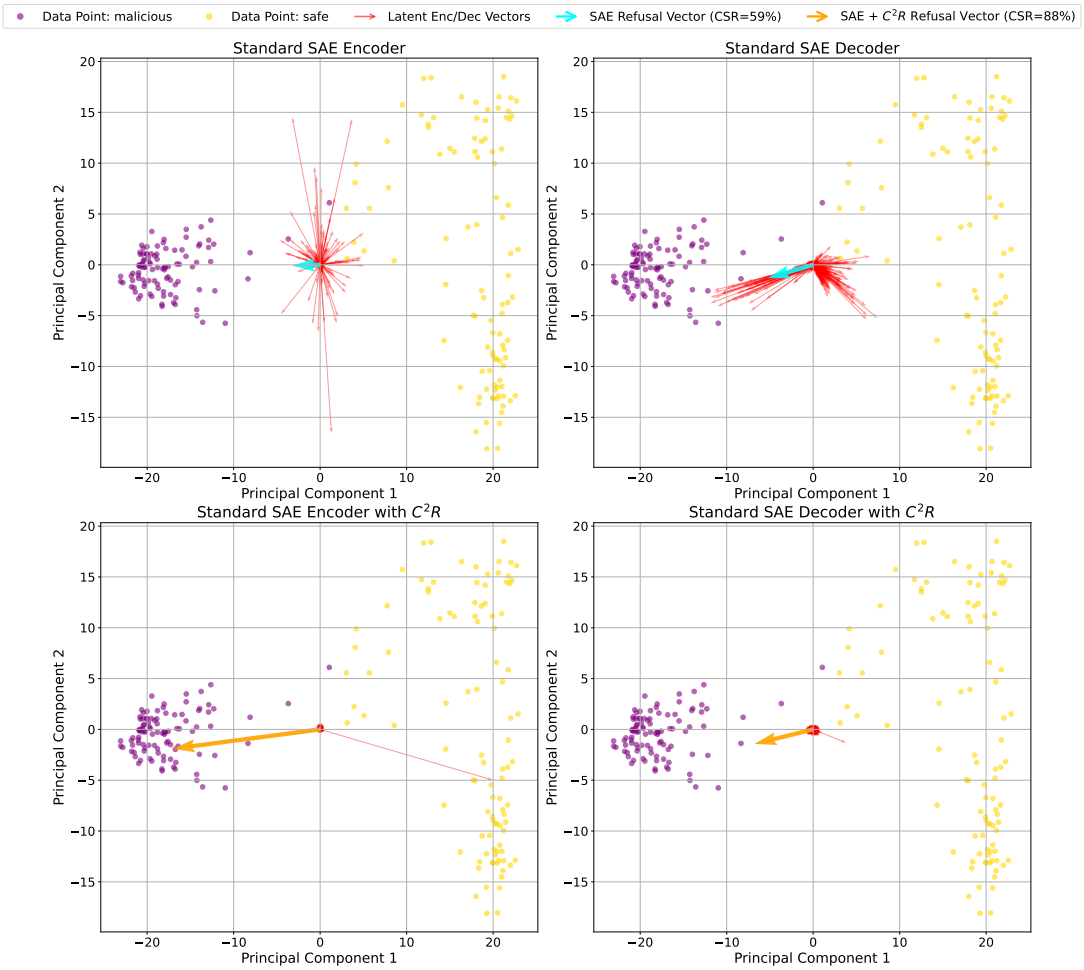

Figure 2: Latent visualization of SAE (top) and SAE + C^{2}R (bottom) trained on residual stream activations from layer 20 of Llama-2-7b-chat (Touvron et al., 2023). Activations of malicious and safe prompts are projected into 2D via PCA (Pearson, 1901). Red arrows indicate latent directions scaled by their ℓ_2 norm. SAE fragments the malicious–safe separation across multiple redundant latent directions, whereas SAE + C^{2}R consolidates this separation into a single, consistent latent. The latents selected for steering the model correspond to the cyan arrow (SAE) and the orange arrow (SAE + C^{2}R) in the upper-right and lower-right panels, respectively. "CSR" in the legend refers to the Control Success Rate, i.e., the proportion of safe prompts successfully steered to refusal (see Section 4.5 for details).

	Prediction Match	Pearson r
Annotator 1	97.8%	0.74
Annotator 2	96.7%	0.63
Annotator 3	97.6%	0.86
Average	97.3%	0.74

Table 8: User study comparing GPT-5-mini (OpenAI, 2025) and human annotators in the automated interpretability task. The table reports the match rate and Pearson correlation (Pearson, 1895) between human- and LLM-derived AutoInterp scores.

Evaluating the 56 trained SAEs across all metrics required approximately 51 GPU hours. The breakdown of evaluation time per SAE is as follows:

- Reconstruction Fidelity Metrics: 30 minutes
- Interpretability Analysis: 10 minutes
- Feature Hierarchy Metrics: 15 minutes

In terms of memory footprint, the reconstruction fidelity evaluation was the most demanding, requiring up to 55 GB of VRAM. The interpretability and feature hierarchy analyses were less memory-intensive, each requiring approximately 15 GB.