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

Anonymous Authors[1]

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. These issues stem from inconsistent latent assignment across samples: without cross-sample constraints, per-sample optimization often allows a single underlying concept to be inconsistently distributed across multiple redundant or interfering latents. To address this, we introduce C^{2}R (**C**ross-sample **C**onsistency **R**egularization). C^{2}R explicitly encourages that each semantic feature is consistently represented by a unified latent across the batch by penalizing the co-activation of directionally similar latents. Comprehensive evaluation demonstrates that C^{2}R effectively mitigates both splitting and absorption while, crucially, preserving reconstruction fidelity, providing a principled solution that enhances latent interpretability without degrading model performance. Source code is available[*].

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 model's representation are more effectively captured by an overcomplete sparse basis (Olshausen & 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.

SAE Constraints	Theoretical Guarantee		Intuitively Solved		Reconstruction Preservation
	Splitting	Absorption	Splitting	Absorption	
ℓ_1	×	×	×	×	✓
TopK	×	×	×	×	✓
Batch TopK	×	×	×	×	✓
Matryoshka	×	×	✓	✓	×
Ort	×	×	×	✓	✓
Ours	✓	✓	✓	✓	✓

Table 1. Comparison of different SAE constraints in terms of theoretical guarantees, intuitive solutions, and reconstruction fidelity. Our cross-sample consistency regularization uniquely offers a theoretical guarantee against splitting and absorption and preserves reconstruction fidelity.

While SAEs are effective, they face significant challenges, specifically feature splitting (Bricken et al., 2023; Leask et al.) and feature absorption (Chanin et al., 2024), which undermine the reliability of learned latents. Feature splitting fragments coherent, high-level concepts into overly specific pieces. For example, a single "Mathematics" feature might break down into separate latents for "Algebra," "Geometry," and others (Chanin et al., 2024). Although these granular features are interpretable, this fragmentation is problematic because it obscures the true high-level concept the model functionally uses. Moreover, this is inefficient: the dictionary wastes capacity on redundant variations of known concepts instead of finding new ones. Feature absorption, on the other hand, creates "holes" in general features when specific latents capture their activations. A "starts with S" latent, for instance, might fail to activate on "short" or "small" because token-specific latents absorb the signal. This effectively changes the latent to "starts with S (except for short/small)," distorting the intended pattern. Leask et al. show that splitting scales with model size, while Chanin et al. (2024) find that absorption affects hundreds of LLM SAEs. These systematic failures undermine the utility of SAEs for critical tasks like causal analysis and circuit discovery.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

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

We argue that these failures arise from a mismatch between the hierarchical nature of language model features and the local scope of standard sparsity constraints. Real-world concepts are inherently hierarchical (Bussmann et al.), yet common objectives like ℓ_1 (Bricken et al., 2023) or TopK (Gao et al.) enforce sparsity on a per-sample basis, which potentially penalizes the hierarchical structure. activating both uses more of the sparsity budget than activating the child feature alone. Consequently, the optimizer suppresses the parent latent and forces the child latent to take over its role to keep the active count low. Similarly, regarding feature splitting, the objective does not distinguish between activating a general latent or a specific one. The SAE allows disjoint latents to handle different contexts of a single concept, as nothing ensures consistent latent assignment across samples. Solving these issues, therefore, requires looking beyond per-sample optimization to enforce cross-sample consistency in how latents are selected.

To address this issue, we propose C^{2}R (**C**ross-sample **C**onsistency **R**egularization). This objective builds on the geometry of the Minkowski inequality (Gruber, 1987) and the strict convexity of the ℓ_2 norm. It exploits the fact that the sum of the norms of separate vectors strictly exceeds the norm of their sum: $\|u\|_2 + \|v\|_2 > \|u + v\|_2$ for non-aligned vectors. By applying this constraint across the batch dimension, C^{2}R makes it expensive to split a concept into multiple disjoint latents. This formulation penalizes spreading semantic information across redundant latents, driving the SAEs to consolidate activations into a single, consistent latent without supervision.

Our contributions are threefold:

- **Theoretical diagnosis:** We identify the lack of cross-sample consistency in per-sample sparsity objectives as the root cause of feature splitting and absorption, providing a unified formal analysis of these phenomena.
- **Principled objective:** We propose C^{2}R, a novel regularization objective that utilizes decoder geometry and batch-level statistics to enforce consistent latent selection, effectively distinguishing between true polysemanticity and harmful redundancy.
- **Empirical validation:** We demonstrate that C^{2}R significantly mitigates splitting and absorption, achieving better feature hierarchy without compromising reconstruction fidelity compared to state-of-the-art baselines.

2. Related Work

2.1. Sparse Autoencoders

Sparse Autoencoders (SAEs) are grounded in the linear representation hypothesis, which posits that the dense activation space of a language model is constructed from the superposition of sparse, discernible concepts, referred to as *features*. The goal of an SAE is to recover these ground-truth features by learning a dictionary of *latents*. Ideally, there exists a one-to-one mapping where each learned latent corresponds precisely to a single meaningful feature.

Formally, given an input activation vector $x \in \mathbb{R}^{d_{model}}$ (e.g., from a Transformer's residual stream), an SAE projects x into a higher-dimensional sparse latent code $f \in \mathbb{R}^{d_{dict}}$, where $d_{dict} \gg d_{model}$. The encoding process is parameterized by an encoder weight matrix $W_e \in \mathbb{R}^{d_{dict} \times d_{model}}$ and a bias b_e:

$$f = \phi(W_e x + b_e), \tag{1}$$

where ϕ is a non-linear activation function, typically ReLU, TopK, or JumpReLU, designed to induce sparsity. The input is then reconstructed via a linear decoder $W_d \in \mathbb{R}^{d_{model} \times d_{dict}}$:

$$\hat{x} = W_d f + b_d. \tag{2}$$

The training objective minimizes a combination of reconstruction error and a sparsity penalty:

$$\mathcal{L}_{\text{SAE}}(x) = \|x - \hat{x}\|_2^2 + \lambda \mathcal{R}(f). \tag{3}$$

Common choices for the regularizer $\mathcal{R}(f)$ include the ℓ_1 norm (Bricken et al., 2023) or the auxiliary loss associated with TopK constraints (Gao et al.). Various architectural improvements have been proposed to enhance SAE quality. Gated SAEs (Rajamanoharan et al., 2024a) and JumpReLU SAEs (Rajamanoharan et al., 2024b) introduce learnable thresholds to improve the fidelity-sparsity frontier. However, these methods focus on the per-sample activation sparsity rather than enforcing the hierarchical structure of the SAE.

2.2. Structural SAEs

More recently, approaches attempting to structure the latent space have emerged. Batch TopK SAEs (Leask et al.) relax the rigid per-sample TopK constraint to a batch-level aggregate, allowing for variable sparsity across samples. While this improves reconstruction, it lacks any mechanism to enforce hierarchical structure among the latents and still faces feature absorption and splitting challenges.

Matryoshka SAEs (Bussmann et al.) enforce a nested structure where subsets of latents are trained to approximate the input at different sparsity levels. While this creates a hierarchy, it compromises reconstruction fidelity and lacks a clear theoretical explanation for why it would fix the optimization issues of standard sparsity objectives. OrtSAE (Korznikov et al., 2025) addresses feature splitting and composition by penalizing the cosine similarity between decoder weights. However, this approach tries to handle absorption indirectly via the decoder geometry, rather than addressing the encoder activation patterns where absorption is formally defined. In

contrast, our method targets the latent activations directly to penalize redundancy, offering a theoretically guaranteed solution derived from the formal definitions of splitting and absorption.

2.3. 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

$$\|a + b\|_p \leq \|a\|_p + \|b\|_p. \tag{4}$$

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

3. Unified Problem Formulation

In this section, we propose a geometric framework that unifies feature splitting and absorption. Rather than viewing them as separate pathologies, we model them as instances of latent redundancy arising from perturbed basis directions. We introduce a **redundancy parameter** α to quantify the extent to which a semantic feature "leaks" into varying latents, allowing us to derive a single consistency condition that prevents both failure modes.

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:

$$\|L_1\| = \|L_2\| = 1, \quad L_1 \perp L_2. \tag{5}$$

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:

$$L_1' = L_1, \quad L_2' = \frac{(1-\alpha)L_1 + \alpha L_2}{\|(1-\alpha)L_1 + \alpha L_2\|}. \tag{6}$$

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 2. 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. 6. The $\sqrt{(\alpha z_1^{(i)})^2 + (z_2^{(i)})^2}$ term ensures the total reconstructed 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.

4. Theoretical Analysis on Two SAE Latents

4.1. Limitation of Per-sample Sparsity Objectives

We analyze the behavior of ℓ_1 and TopK objectives under the activation patterns defined in the unified problem formulation.

Lemma 4.1. *Per-sample sparsity constraints, specifically ℓ_1 regularization and TopK, strictly favor feature splitting and absorption ($\alpha \to 1$) over the ideal orthogonal decomposition ($\alpha = 0$) given equivalent reconstruction fidelity.*

Proof. **Case 1: The ℓ_1 Penalty.** We compare the cumulative ℓ_1 penalty for the ideal configuration versus the per-

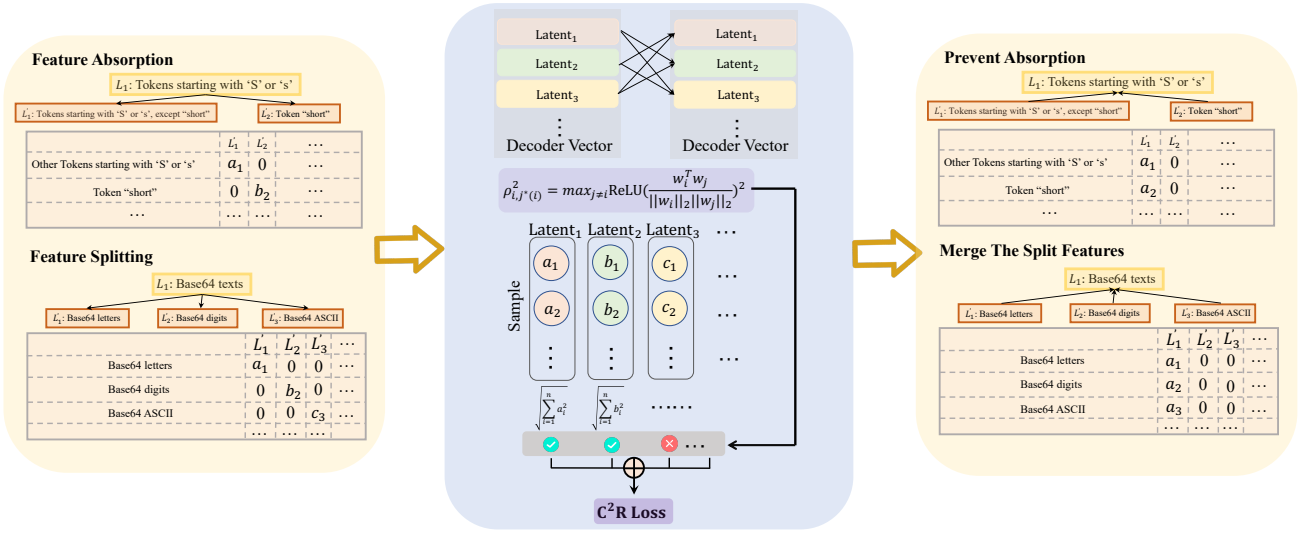


Figure 1. Overview of C^{2}R (Cross-sample Consistency Regularization). Each mini-batch contains activations of multiple samples encoded by a sparse autoencoder. C^{2}R enforces consistency of latent usage across samples by constraining activation patterns along the batch dimension. This encourages each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

turbed configuration, assuming equivalent SAE reconstruction.

(1) **Ideal Decomposition** ($\alpha = 0$)**:** In this state, the feature L_1 and the residual L_2 are orthogonal. The total ℓ_1 norm over the batch is:

$$\ell_1(\alpha = 0) = \sum_{i=1}^{m+n} z_1^{(i)} + \sum_{i=m+1}^{m+n} z_2^{(i)}. \tag{7}$$

(2) **Perturbed Decomposition** ($\alpha > 0$)**:** Under splitting or absorption, the activation energy is redistributed. The total ℓ_1 norm becomes:

$$\begin{aligned}\ell_1(\alpha > 0) = &\sum_{i=1}^{m} z_1^{(i)} + \sum_{i=m+1}^{m+n} (1-\alpha) z_1^{(i)} \\ &+ \sum_{i=m+1}^{m+n} \sqrt{(\alpha z_1^{(i)})^2 + (z_2^{(i)})^2}.\end{aligned} \tag{8}$$

We apply the Minkowski inequality to the terms summing over indices $i \in \{m+1, \ldots, m+n\}$. For any $\alpha > 0$:

$$\sum_{i=m+1}^{m+n} \alpha z_1^{(i)} + \sum_{i=m+1}^{m+n} z_2^{(i)} > \sum_{i=m+1}^{m+n} \sqrt{(\alpha z_1^{(i)})^2 + (z_2^{(i)})^2}. \tag{9}$$

Subtracting the shared terms from Eq. 7 and Eq. 8, it follows that $\ell_1(\alpha > 0) < \ell_1(\alpha = 0)$. The strict inequality holds for all $\alpha \in (0, 1]$. Consequently, the optimization process minimizes the global objective by maximizing α, thereby driving the solution toward splitting or absorption to reduce the total norm while maintaining reconstruction fidelity.

Case 2: The TopK Constraint. The TopK objective enforces a hard constraint on the number of active latents, effectively minimizing the ℓ_0 norm of the activation vector for a fixed reconstruction error tolerance. We evaluate the cardinality of the active set for the intersection samples $i \in \{m+1, \ldots, m+n\}$.

(1) **Ideal Decomposition** ($\alpha = 0$)**:** The signal consists of two orthogonal non-zero components: the feature activation $z_1^{(i)}$ and the residual $z_2^{(i)}$. Since the basis vectors L_1 and L_2 are orthogonal, exact representation requires both to be active. Thus, the sparsity consumption is:

$$\|z^{(i)}\|_0 = 2 \quad \text{for } i \in \{m+1, \ldots, m+n\}. \tag{10}$$

(2) **Full Splitting or Absorption** ($\alpha = 1$)**:** Substituting $\alpha = 1$ into the activation pattern defined in Table 2, the coefficient for the first latent becomes zero: $(1-\alpha)z_1^{(i)} = 0$. The second latent, L_2', captures the entire vector magnitude $\sqrt{(z_1^{(i)})^2 + (z_2^{(i)})^2}$. As a result, the SAE represents the same vector space using a single active latent:

$$\|z^{(i)}\|_0 = 1 \quad \text{for } i \in \{m+1, \ldots, m+n\}. \tag{11}$$

By reducing the active set from 2 latents to 1, the split configuration ($\alpha = 1$) saves the sparsity budget. This creates a strong pressure on the optimization: the TopK constraint pushes the SAEs to use latents to represent mixed feature directions rather than maintaining atomic features, as this saves capacity within the k-latent budget to reconstruct other features and lower the global reconstruction loss. □

4.2. Mitigating Splitting via Cross-Sample Consistency

Previous analysis shows that per-sample objectives (ℓ_1 and TopK) cannot distinguish between atomic and split features. In contrast, the Minkowski inequality (Gruber, 1987) (Eq.4) for $p = 2$ offers a strict convexity condition to reverse this preference. Since the sum of norms for separate vectors is always greater than the norm of their sum, minimizing the sum of ℓ_2 norms across the batch dimension, i.e. $\|Z_{:,1}\|_2 + \|Z_{:,2}\|_2$, encourages SAEs to consolidate shared semantic feature into a single latent.

We analyze this mechanism using the unified formulation in Table 2. We define a regularization term $\mathcal{L}_{\text{pair}}$ as the sum of the batch-norms for the two split latents:

$$\begin{aligned}\mathcal{L}_{\text{pair}}(\alpha) &= \|Z_{:,1}\|_2 + \|Z_{:,2}\|_2 \\ &= \sqrt{\sum_{i=1}^{m}(z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}((1-\alpha)z_1^{(i)})^2} \\ &+ \sqrt{\sum_{i=m+1}^{m+n}(\alpha z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}(z_2^{(i)})^2}.\end{aligned} \tag{12}$$

To check if minimizing this term suppresses splitting, we calculate its gradient with respect to the splitting factor α. The derivative $\frac{\partial \mathcal{L}_{\text{pair}}}{\partial \alpha}$ shows that the loss increases monotonically with α (meaning the penalty reduces splitting) as long as the following condition holds (derivation in Appendix A):

$$\alpha \geq \frac{1}{\sqrt{\frac{\sum_{i=1}^{m}(z_1^{(i)})^2}{\sum_{i=m+1}^{m+n}(z_2^{(i)})^2}} + 1}. \tag{13}$$

Empirical results from recent literature support this condition for practical SAE training. Leask et al. find that split latents retain high cosine similarity with their parent latents, implying the feature component magnitude far exceeds the residual ($\|z_1^{(i)}\| \gg \|z_2^{(i)}\|$). Additionally, Chanin et al. (2024) note large frequency gaps between parent and child features (e.g., $P(f_0) = 0.25$ vs $P(f_1) = 0.05$), which means the cumulative energy of the primary feature dominates the residual:

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

Under these settings, the right-hand side of Eq. 13 approaches zero. As a result, $\mathcal{L}_{\text{pair}}$ increases monotonically with α in the relevant domain. This confirms that a cross-sample ℓ_2 penalty on redundant pairs theoretically ensures the consolidation of semantically related activations into a single consistent latent, mitigating both splitting and absorption.

5. Cross-Sample Consistency Regularization

Following the analysis in Section 4, we introduce **C^{2}R** (**C**ross-sample **C**onsistency **R**egularization). Minkowski inequality helps merge redundant features, but applying it to a large dictionary requires a careful approach. Here, we extend the pairwise analysis to the full SAE dictionary and examine the gradient dynamics that enable C^{2}R to enforce both cross-sample consistency and latent orthogonality.

5.1. Generalizing from Pairwise to Multiple Latents

The derivation in Section 4 used a simple system with two latents for one parent feature and a child feature. However, standard SAEs have thousands of latents, and most represent distinct concepts. If we minimize the sum of norms across random pairs without selection, we would force independent features to merge, which causes feature collapse and reduces the SAEs' ability to resolve semantic differences.

We therefore need the regularization to be selective. It should only penalize latent pairs that look like split fragments or absorbed variations, while leaving independent features alone. Previous work on feature splitting (Chanin et al., 2024) and orthogonal constraints (Korznikov et al., 2025) shows that redundant latents usually have similar decoder weight directions. In contrast, distinct features tend to be nearly orthogonal in the high-dimensional space.

Based on this, we use the cosine similarity of decoder weights to detect redundancy. For each latent i, we find its nearest neighbor $j^*(i)$ in the decoder space:

$$j^*(i) = \arg\max_{j \neq i} \langle \hat{w}_i, \hat{w}_j \rangle, \quad \text{where } \hat{w} = \frac{w}{\|w\|_2}. \tag{15}$$

We then define the C^{2}R loss by weighting the pairwise norm penalty with the squared rectified cosine similarity $\rho^2_{i,j^*(i)} = \text{ReLU}(\langle \hat{w}_i, \hat{w}_{j^*(i)} \rangle)^2$:

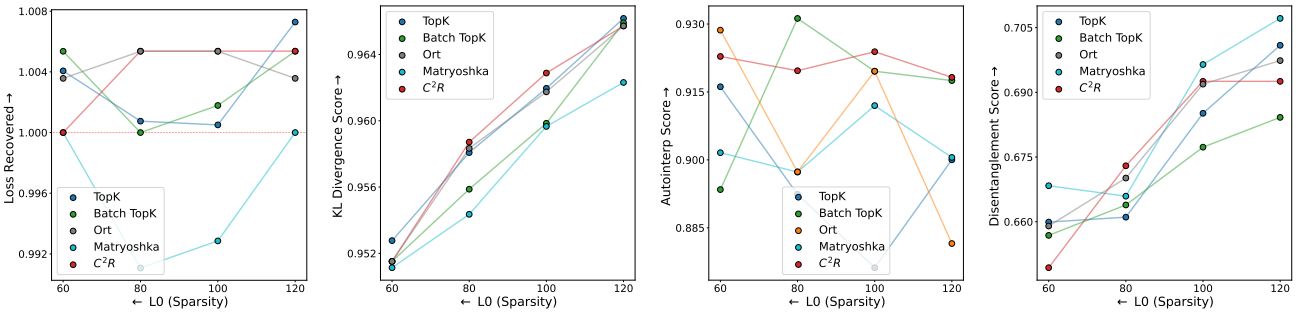
$$\mathcal{L}_{\text{C}^2\text{R}}(X) = \frac{1}{k}\sum_{i=1}^{k} \rho^2_{i,j^*(i)} \cdot \underbrace{\left(\|Z_{:,i}\|_2 + \|Z_{:,j^*(i)}\|_2\right)}_{S_{i,j^*(i)}}. \tag{16}$$

This weight acts as a gate. For distinct features where $\hat{w}_i \perp \hat{w}_j$, ρ^2 is zero, so the consistency constraint does not apply. For redundant features with high alignment, ρ^2 is large, which fully applies the regularization.

The final training objective adds this regularization to the standard SAE loss, which includes reconstruction and sparsity terms:

$$\mathcal{L}(X) = \mathcal{L}_{\text{SAE}}(X) + \lambda_{\text{C}^2\text{R}}\mathcal{L}_{\text{C}^2\text{R}}(X), \tag{17}$$

where $\lambda_{\text{C}^2\text{R}}$ controls the weight of the cross-sample consistency term.



(a) Loss Recovered (b) KL Divergence Score (c) Autointerp (d) Disentanglement Score

Figure 2. **Quantitative comparison of SAE performance across different sparsity levels.** (a) and (b) evaluate **reconstruction fidelity** using Cross-Entropy Loss and KL Divergence at the LLM output layer, respectively, (c) assesses SAE latent **interpretability** using the Autointerp score, and (d) measures the extent to which real-world features are disentangled into steerable latents. Notably, C^{2}R-enhanced Batch TopK SAE does not exhibit performance degradation compared to the vanilla Batch TopK baseline. Instead, it maintains competitive or superior results, particularly in KL Divergence and Autointerp scores, demonstrating that our proposed C^{2}R constraint preserves these important capabilities.

5.2. Gradient Analysis and Implicit Orthogonality

We can better understand C^{2}R by looking at its gradient dynamics. The loss function is the product of a geometric alignment term (ρ^2) and an activation magnitude term ($S_{i,j^*(i)}$). The gradient with respect to the model parameters θ follows the product rule $\nabla(AB) = B\nabla A + A\nabla B$:

$$\nabla_\theta \mathcal{L}_{\text{C}^2\text{R}} \propto \underbrace{\rho^2 \cdot \nabla_\theta S_{i,j^*(i)}}_{\text{Consistency Gradient}} + \underbrace{S_{i,j^*(i)} \cdot \nabla_\theta(\rho^2)}_{\text{Orthogonality Gradient}}. \quad (18)$$

This shows that C^{2}R creates two simultaneous forces during optimization.

1. Consistency Pressure ($\rho^2 \nabla S$). The first term minimizes the sum of norms, scaled by the similarity weight ρ^2. As shown in Section 4, this uses the Minkowski inequality to push the splitting factor α toward 0. It merges the activation energy of redundant latents into a single one, which helps fix feature splitting and absorption.

2. Implicit Orthogonality Pressure ($S \nabla \rho^2$). The second term minimizes cosine similarity between decoder weights to encourage feature orthogonality, similar to the goals of OrtSAE (Korznikov et al., 2025). Unlike OrtSAE that typically applies a uniform penalty to all selected max-cosine pairs, our method scales the penalty by $S_{i,j^*(i)}$ (the sum of feature activations). This allows C^{2}R to adjust regularization based on feature frequency and magnitude. Strong, high-frequency features (large S) are subject to stricter orthogonality pressure to prevent redundancy. Conversely, for newly initialized or rare latents (small S), aggressive orthogonality enforcement can be counterproductive, potentially pushing them away from valid directions before they stabilize. By scaling the gradient with activation magnitude, C^{2}R prevents such disruption, allowing developing latents to converge naturally. This mechanism promotes orthogonality while adapting the regularization strength to each feature's convergence state.

6. EXPERIMENTS

6.1. Baselines

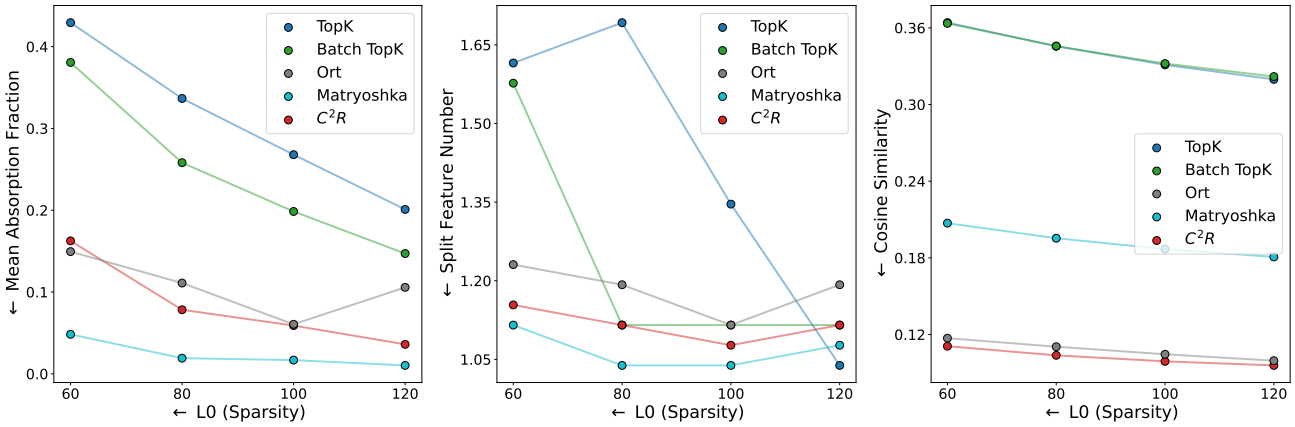
We integrate C^{2}R into four different SAE architectures: TopK SAEs (Gao et al.), Batch TopK SAEs (Leask et al.), Matryoshka SAEs (Bussmann et al.), and Ort-SAEs (Korznikov et al., 2025), as introduced in 2. We evaluate performance by iterating over sparsity levels of $k \in \{60, 80, 100, 120\}$, ensuring a fair comparison between baselines and C^{2}R-enhanced Batch TopK SAE under equivalent sparsity conditions.. We run baselines and implement C^{2}R 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.

6.2. Experiment Settings

We conduct systematic experiments on Gemma-2-2B (Team, 2024)[†] to validate the effectiveness of C^{2}R. Specifically, we sample a 500M-token subset from the OpenWebText dataset (Gokaslan et al., 2019)[‡] 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.). We performed a hyperparameter sweep for $\lambda_{\text{C}^2\text{R}}$ over the set $\{0.1, 0.5, 1, 5, 10\}$. We selected $\lambda_{\text{C}^2\text{R}} = 5$, as it represents the maximal regularization strength that does not degrade reconstruction fidelity. All SAEs are optimized with Adam using a learning rate of

[†]Gemma Terms of Use

[‡]Creative Commons Zero v1.0 Universal



(a) Feature absorption (b) Feature splitting (c) Feature composition

Figure 3. **Analysis of feature structural metrics.** We compare feature absorption, splitting, and composition across different sparsity levels. Among methods that maintain high reconstruction fidelity (i.e., excluding Matryoshka SAEs), the proposed C^{2}R constraint achieves the lowest rates of feature absorption and splitting. Furthermore, it achieves the optimal performance in feature decomposition, consistently yielding the lowest cosine similarity to ensure more atomic features.

2×10^{-4}, batch size of 2,048, and context length of 1,024.

Computational Efficiency. Computing the pairwise cosine similarity for a large dictionary (e.g., $k = 65,536$) imposes a quadratic $O(k^2)$ computational complexity and substantial memory overhead. To ensure computational feasibility and maintain a fair comparison with the state-of-the-art baseline, we adopt the efficient optimization strategy used in OrtSAEs (Korznikov et al., 2025). Specifically, we employ a block-wise computation strategy with a chunk size of 8,192 and compute the consistency regularization term every 5 training steps, scaling the coefficient $\lambda_{\text{C}^2\text{R}}$ accordingly. This approach reduces the overhead to negligible levels while preserving performance (detailed in Appendix B).

6.3. Metrics

To holistically assess the relative performance of SAEs when integrating C^{2}R, our evaluation utilizes a comprehensive set of seven key metrics. These metrics, implemented using the code framework from SAEBench (Karvonen et al.), cover four areas: **reconstruction fidelity**, **feature hierarchy**, **interpretability**, and **disentanglement**. Specifically, our evaluation consists of six key metrics: **Loss Recovered**, **KL Div. Score**, **AutoInterp**, **Split Num**, **Absorption Rate**, **Composition**, and **Disentanglement**. Detailed description of these metrics is in E.

6.4. Reconstruction Fidelity

As shown in Figure 2, integrating C^{2}R preserves reconstruction fidelity of the backbone SAEs. Figure 2a reports the Loss Recovery metric (details in Appendix E), where a value ≥ 1 indicates that the SAE-reconstructed LLM activations yield a cross-entropy loss lower than or equal to the original LLM. We observe that across the tested sparsity levels, only Matryoshka SAEs exhibit performance loss, while other methods maintain full recovery. Figure 2b displays the KL Divergence Score, which measures the shift in the LLM's output logit distribution. Higher scores correspond to smaller deviations from the original distribution. The results demonstrate that SAEs trained with the C^{2}R constraint maintain high reconstruction fidelity, strictly adhering to the original model's behavior.

6.5. Interpretability

Following the evaluation framework of (Paulo et al.), we assess the interpretability of all SAEs using the *AutoInterp* metric. The results in Figure 2c show that adding C^{2}R does not significantly affect interpretability. For each SAE, we sampled 128 latents and constructed prompts based on their activations over a 2M-token input. We prompt an LLM explainer to generate concise and comprehensive textual descriptions for each latent from 15 observed samples, and prompt 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.

6.6. Disentanglement

We evaluate disentanglement using the RAVEL benchmark (Huang et al., 2024), which employs interchange interventions to determine if specific SAE latents causally control individual attributes such as continent or gender. The metric averages cause and isolation scores to measure how well the SAE governs a target concept without affecting unrelated ones. As illustrated in Figure 2d, Batch TopK SAEs trained with the C^2R constraint achieve performance comparable to the vanilla baseline. This demonstrates that our regularization effectively preserves a disentangled and manipulable latent space.

6.7. Feature Splitting and Feature Absorption

Figures 3a and 3b illustrate the evaluation of feature absorption and splitting. Our proposed C^2R constraint effectively mitigates both issues compared to the vanilla Batch TopK baseline and outperforms the orthogonal constraint in OrtSAE. Although Matryoshka SAEs achieve lower absorption and splitting scores, this improvement comes at the expense of reconstruction fidelity, as evidenced in Figure 2a and 2b. Therefore, among methods that maintain the original model's performance, the C^2R constraint strikes the best balance, achieving the lowest levels of absorption and splitting while preserving latent quality and reconstruction capabilities.

6.8. Feature Composition

Following Bussmann et al., we quantify feature composition by measuring the average maximum cosine similarity between latent vectors. High similarity implies that multiple latents represent overlapping information, indicating a lack of atomicity. As shown in Figure 3c, our approach achieves the optimal performance on this metric. The C^2R constraint yields significantly lower cosine similarity compared to vanilla Batch TopK and Matryoshka SAEs, and marginally outperforms OrtSAE. This result confirms that our method effectively minimizes redundancy, producing a set of highly distinct and atomic features.

7. Conclusion

In this work, we introduced C^2R, a theoretically grounded constraint aimed at improving feature hierarchy in sparse autoencoders. By leveraging the Minkowski inequality, our approach provides a rigorous guarantee against feature splitting and absorption. Extensive empirical evaluations show that incorporating the C^2R constraint significantly enhances the structural quality of the learned dictionary, reducing feature composition and minimizing redundancy. Importantly, our approach preserves reconstruction fidelity, latent interpretability, and disentanglement capabilities on par with the baselines. These results highlight the effectiveness of the cross-sample consistency regularization in addressing the trade-off between feature atomicity and model fidelity, offering a robust solution for large language model analysis.

8. Limitations

While our theoretical analysis and empirical results demonstrate that the C^2R constraint effectively enhances SAE latent hierarchy, our study has some potential limitations. First, our experiments primarily focus on the residual stream of gemma-2-2b. Investigating the scalability of C^2R to larger foundation models and diverse architectures, such as Mixture-of-Experts, remains a priority for future research. Second, although we establish strong performance on feature hierarchy metrics, applying C^2R-enhanced SAEs to practical downstream tasks like model steering in open-ended generation would further validate their utility.

Impact Statement

This work advances the methodology of sparse dictionary learning for neural networks. By improving the fidelity of feature extraction, we aim to enable a more granular understanding of large language model internals. Such interpretability is essential for auditing model behavior, identifying latent failure modes, and verifying safety properties prior to deployment. However, we acknowledge that deeper insights into model representations can also be leveraged to improve model efficiency or steerability, potentially accelerating the development of powerful systems and amplifying the societal risks associated with their deployment.

References

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bussmann, B., Nabeshima, N., Karvonen, A., and Nanda, N. Learning multi-level features with matryoshka sparse autoencoders. In *Forty-second International Conference on Machine Learning*.

Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., Golechha, S., and Bloom, J. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv preprint arXiv:2409.14507*, 2024.

Ferrando, J., Obeso, O. B., Rajamanoharan, S., and Nanda, N. Do i know this entity? knowledge awareness and hallucinations in language models. In *The Thirteenth International Conference on Learning Representations*.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Gokaslan, A., Cohen, V., Pavlick, E., and Tellex, S. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

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

Huang, J., Wu, Z., Potts, C., Geva, M., and Geiger, A. RAVEL: Evaluating interpretability methods on disentangling language model representations. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8669–8687, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.470. URL https://aclanthology.org/2024.acl-long.470/.

Huben, R., Cunningham, H., Smith, L. R., Ewart, A., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Karvonen, A. dictionary_learning_demo. https://github.com/adamkarvonen/dictionary_learning_demo, 2024.

Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J. I., Chanin, D., Lau, Y.-T., Farrell, E., McDougall, C. S., Ayonrinde, K., et al. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Forty-second International Conference on Machine Learning*.

Korznikov, A., Galichin, A., Dontsov, A., Rogov, O., Tutubalina, E., and Oseledets, I. Ortsae: Orthogonal sparse autoencoders uncover atomic features, 2025. URL https://arxiv.org/abs/2509.22033.

Lan, M., Torr, P., Meek, A., Khakzar, A., Krueger, D., and Barez, F. Sparse autoencoders reveal universal feature spaces across large language models. *arXiv preprint arXiv:2410.06981*, 2024.

Leask, P., Bussmann, B., Pearce, M. T., Bloom, J. I., Tigges, C., Al Moubayed, N., Sharkey, L., and Nanda, N. Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations*.

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

OpenAI. Gpt-5 system card, 2025. URL https://openai.com/index/gpt-5-system-card/.

Paulo, G. S., Mallen, A. T., Juang, C., and Belrose, N. Automatically interpreting millions of features in large language models. In *Forty-second International Conference on Machine Learning*.

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

Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramár, J., Shah, R., and Nanda, N. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024a.

Rajamanoharan, S., Lieberum, T., Sonnerat, N., Conmy, A., Varma, V., Kramár, J., and Nanda, N. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024b.

Team, G. Gemma. 2024. doi: 10.34740/KAGGLE/M/3301. URL https://www.kaggle.com/m/3301.

Wang, S., Asilis, J., Akgül, Ö. F., Bilgin, E. B., Liu, O., Fu, D., and Neiswanger, W. Resa: Transparent reasoning models via saes. *arXiv preprint arXiv:2506.09967*, 2025.

Yeo, W. J., Prakash, N., Neo, C., Lee, R. K.-W., Cambria, E., and Satapathy, R. Understanding refusal in language models with sparse autoencoders. *arXiv preprint arXiv:2505.23556*, 2025.

Zhao, Y., Devoto, A., Hong, G., Du, X., Gema, A. P., Wang, H., He, X., Wong, K.-F., and Minervini, P. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *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)*, pp. 5117–5136, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.264. URL https://aclanthology.org/2025.naacl-long.264/.

A. C^{2}R Promotes Cross-sample Consistency

When feature absorption occurs, the C^{2}R Loss is:

$$\begin{aligned}\mathcal{L}_{\mathrm{C^2R}}(X) = &\sqrt{\sum_{i=1}^{m}(z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}((1-\alpha)z_1^{(i)})^2} \\ &+ \sqrt{\sum_{i=m+1}^{m+n}(\alpha z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}(z_2^{(i)})^2}.\end{aligned} \tag{19}$$

Taking the partial derivative with respect to α yields:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathrm{C^2R}}(X)}{\partial \alpha} = &\frac{(\alpha-1)\sum_{i=m+1}^{m+n}(z_1^{(i)})^2}{\sqrt{\sum_{i=1}^{m}(z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}((1-\alpha)z_1^{(i)})^2}} \\ &+ \frac{\alpha\sum_{i=m+1}^{m+n}(z_1^{(i)})^2}{\sqrt{\sum_{i=m+1}^{m+n}(\alpha z_1^{(i)})^2 + \sum_{i=m+1}^{m+n}(z_2^{(i)})^2}}.\end{aligned} \tag{20}$$

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

$$\frac{\partial \mathcal{L}_{\mathrm{C^2R}}(X)}{\partial \alpha} \geq 0. \tag{21}$$

Introducing the shorthand notation:

$$\begin{aligned}A &= \sum_{i=1}^{m}(z_1^{(i)})^2 \\ B &= \sum_{i=m+1}^{m+n}(z_2^{(i)})^2 \\ C &= \sum_{i=m+1}^{m+n}(z_1^{(i)})^2\end{aligned} \tag{22}$$

The inequality simplifies to:

$$\frac{\alpha C}{\sqrt{\alpha^2 C + B}} \geq \frac{(1-\alpha)C}{\sqrt{A + (1-\alpha)^2 C}}. \tag{23}$$

The final form is:

$$\alpha \geq \frac{1}{\sqrt{\frac{\sum_{i=1}^{m}(z_1^{(i)})^2}{\sum_{i=m+1}^{m+n}(z_2^{(i)})^2}} + 1} \tag{24}$$

B. Efficient Implementation Details

Applying the C^{2}R constraint naively requires computing the cosine similarity between all pairs of decoder weight vectors, resulting in a $k \times k$ similarity matrix. For a dictionary size of $k = 65,536$, this operation requires $O(k^2)$ memory and computation, which significantly slows down training and increases VRAM usage.

To address this, we implement two engineering optimizations following the methodology of OrtSAE (Korznikov et al., 2025):

Chunk-wise Approximation. Instead of searching for the nearest neighbor $j^*(i)$ across the entire dictionary, we randomly permute the feature indices at each step and partition the dictionary into smaller blocks (chunks). For a dictionary size k and chunk size C, we partition the latents into $N = k/C$ chunks. The nearest neighbor search and loss computation are then restricted to within each chunk.

$$j^*(i) \approx \underset{j \in \mathrm{Chunk}(i), j \neq i}{\arg\max} \langle \hat{w}_i, \hat{w}_j \rangle. \tag{25}$$

In our experiments, we use a chunk size of $C = 8,192$. This reduces the complexity from $O(k^2)$ to $O(C \cdot C)$. Since high-cosine similarity features (redundant pairs) are rare and the permutation is randomized at every step, the probability of a redundant pair falling into the same chunk accumulates rapidly over training steps, ensuring the regularization remains effective.

Periodic Updates. To further reduce the computational overhead, we compute the C^{2}R loss and its gradients only every T training steps rather than at every iteration. To maintain the same effective regularization strength over time, we scale the regularization coefficient $\lambda_{\text{C}^2\text{R}}$ by the period T:

$$\mathcal{L}_{\text{step } t}(X) = \mathcal{L}_{\text{SAE}}(X) + (\mathbb{1}_{t \bmod T=0} \cdot T \cdot \lambda_{\text{C}^2\text{R}})\mathcal{L}_{\text{C}^2\text{R}}(X). \tag{26}$$

We set $T = 5$ for all experiments of OrtSAEs and our approach. This configuration aligns our training cost with other SAEs' training times, adding only negligible overhead (slightly more than the findings in OrtSAE, where overhead was reduced to $< 10\%$).

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 3.

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 4.

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 5.

D. User Study Details

We recruited three human annotators with high-school-level English proficiency, who replicated the explainer–judge process on 30 latents sampled from Batch TopK SAE and its C^{2}R-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 6. 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.

E. Detailed Metrics Description

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

- **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

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 craters↩The« 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 as↩bright 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 3. 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) · 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 that↩you cannot see it. Its circular structure is nearly a kilometer below the↩surface and
9. sound crazy? Okay?↩DW:I'm 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 Dodge↩Strength ••, 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 4. 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 5. Instruction example for human annotators to predict latent activation.

	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 6. 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.

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 that more latent are 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 metric quantifies the maximum cosine similarity between the decoder weight vectors of all learned latents, reflecting the feature composition level of SAEs (Bussmann et al.). High similarity suggests significant directional overlap or redundancy among SAE latents.

- **Disentanglement (Huang et al., 2024)** This benchmark evaluates the ability of interpretability methods to disentangle independent attributes within language model representations. It utilizes interchange interventions to test whether a targeted feature (e.g., city country) can be modified without affecting other related attributes (e.g., city language). The performance is summarized by the disentangle score: Disentangle Score $= \frac{1}{2}(\text{Cause} + \text{Iso})$ where **Cause** measures the success rate of changing the target attribute's value through intervention, and **Iso** (Isolation) measures the frequency with which non-target attributes remain unchanged. A high score indicates that the SAE has successfully localized individual concepts into independent, causal units.

F. GPU Budget

We ran all SAE training experiments utilizing an NVIDIA H800 GPU, consuming a total of 300 GPU hours and achieving a peak memory utilization of 65 GB.

Evaluating a trained SAE across all metrics required approximately 1 GPU hour. The breakdown of evaluation time per SAE is as follows:

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

In terms of memory footprint, the reconstruction fidelity evaluation was the most demanding, requiring up to 50 GB of VRAM. The interpretability and feature hierarchy analyses were less memory-intensive, each requiring approximately 15 GB.