DIAGNOSING AND IMPROVING LATENT RECOVERY IN SPARSE AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) are used to extract interpretable features from neural representations. Their codes assign feature amplitudes used to rank activating examples, attribute model behavior, and set intervention strengths. These applications make amplitude fidelity relevant once feature directions and scales have been fixed. However, reconstruction objectives do not directly measure coefficient accuracy, and interference between co-active features can cause sample-dependent errors that fixed feature-wise rescaling cannot generally remove. Understanding these errors is therefore important for interpreting the codes produced by SAE encoders. We study coefficient recovery in a blind sparse mixing model and characterize amplitude bias in one-pass tied encoders, even with an exact dictionary and correct support. We then compare each code with the code obtained after reconstruction and re-encoding. For tied and untied SAEs, we derive local bounds relating this self-consistency residual to coefficient error under correct and stable support, value-preserving activations, and suitable conditioning. This analysis motivates a self-consistency regularizer that requires one additional encoder evaluation during training and adds no parameters or inference cost. Controlled synthetic experiments show reduced latent recovery error, while CE-bench and SAEBench evaluations provide complementary evidence of improved feature quality.

1 INTRODUCTION

Large neural networks learn rich internal representations that generalize across many downstream tasks (Brown et al., 2020; Caron et al., 2021; Wiedemer et al., 2025; Bengio et al., 2013), yet these representations remain difficult to interpret: individual units and linear directions are typically polysemantic, encoding multiple overlapping concepts rather than clean semantic factors. A prominent hypothesis is that monosemantic features (features associated with a single concept) are sparsely superposed in distributed activations (Elhage et al., 2022), motivating sparse autoencoders (SAEs) (Ng et al., 2011) as a tool for disentangling such representations in language (Cunningham et al., 2023), vision (Fel et al., 2025a;b), and multimodal models (Pach et al., 2025).

Most SAE objectives minimize reconstruction error, yet their codes are also used to rank examples by feature activation, attribute model behavior to features, and set feature-intervention strengths (Cunningham et al., 2023; Bricken et al., 2023; Marks et al., 2025). These uses make amplitude fidelity relevant once feature directions and scales have been fixed. Interference between co-active features can vary across samples and even reverse the ordering of a feature's activations, as illustrated in Section 4.1. Such sample-dependent distortions cannot, in general, be removed by fixed feature-wise rescaling. This raises a basic question: how can we diagnose and reduce coefficient errors that remain after dictionary and support recovery? We study coefficients in a normalized, aligned generative dictionary, and assess feature utility separately on real-model benchmarks, where true latent codes are unobserved.

Classical sparse recovery studies support identification and coefficient estimation, including algorithms that exploit structured supports (Duarte & Eldar, 2011; Baraniuk et al., 2010). O'Neill et al. (2025) characterize limitations of amortized SAE inference and investigate more expressive encoders. Cui et al. (2025) show that latent recovery can fail even in idealized SAE models and propose weighted reconstruction objectives as a remedy. Building on these studies, we isolate coefficient

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

errors that remain even with a correct dictionary and support, and examine whether the SAE's own encoding and decoding operations can reveal these errors.

We analyze this question in a blind sparse mixing model. A tied encoder can distort feature amplitudes through interference between co-active features, even when the dictionary and active support are correct. Motivated by the study of latent-space dynamics (Fumero et al., 2025), we compare the encoder's original output with the code obtained by reconstructing and re-encoding the input. For tied SAEs, this self-consistency residual controls the distance to the least-squares code when the support is correct and stable, the activation preserves the selected values, and the active dictionary is well conditioned. We extend the analysis to untied SAEs and derive local latent recovery bounds that account for decoder mismatch, observation noise, and conditioning. These results motivate a self-consistency regularizer that supplies a coefficient-level training signal while retaining one-pass inference.

Our contributions are:

- We characterize the sample-dependent amplitude bias of one-pass tied encoders even when the decoder dictionary and active support are correct. This isolates coefficient errors from dictionary and support errors after accounting for permutation and scaling ambiguities.
- We establish conditions under which the change in a code after reconstruction and re-encoding controls its coefficient error. For tied and untied SAEs, we derive local latent recovery bounds with explicit dependence on decoder mismatch, observation noise, and conditioning.
- We propose a self-consistency regularizer that can be added to standard SAE training objectives. It requires one additional encoder evaluation during training, adds no parameters or inference cost, and admits a masked variant for TopK encoders.
- We demonstrate reduced latent recovery error in controlled tied and untied synthetic experiments, including a fixed-decoder control, while separately measuring dictionary, support, and amplitude recovery. CE-bench and selected SAEBench evaluations provide complementary evidence of improved feature quality at matched activation budgets.

Section 2 discusses the closest connections to prior work; Appendix A provides additional background.

2 Related Work

SAE architectures and coefficient estimation. SAEs extract sparse features from neural activations (Cunningham et al., 2023). TopK directly controls per-example sparsity (Gao et al., 2024), while BatchTopK allocates the activation budget across a batch (Bussmann et al., 2024). Gated SAEs separate feature selection from magnitude estimation to address shrinkage (Rajamanoharan et al., 2024). Our analysis isolates amplitude distortion caused by correlated atoms, which can persist even when the dictionary and support are correct.

Identifiability and amortized inference. Cui et al. (2025) characterize identifiability in idealized SAE models and propose weighted reconstruction to improve latent recovery. O'Neill et al. (2025) establish limitations of linear-nonlinear SAE inference and evaluate more expressive encoders. Paulo & Belrose (2025) find that independently trained SAEs can learn different features from the same data. We study coefficient refinement through self-consistency while retaining the original inference architecture.

Sparse coding and dictionary learning. Model-based compressed sensing exploits support structure (Baraniuk et al., 2010), while neural sparse-coding algorithms provide dictionary-learning guarantees under suitable assumptions (Arora et al., 2015). Nguyen et al. (2019) analyze autoencoder training dynamics, and Tolooshams & Ba (2022) study learning through unrolled sparse inference. Hindupur et al. (2025) connect SAE architectures to assumptions about concept geometry. These works contextualize our selected-support analysis of encoder-induced coefficient errors. Appendix A provides further background.

3 PROBLEM SETTING & SAE BACKGROUND

3.1 OBSERVATION MODEL: BLIND SPARSE MIXING

We study a blind sparse mixing model in which an unobserved sparse latent code $x \in \mathbb{R}^{d_m}$ is observed only through a linear mixture

$$x_p = W_p x, \qquad W_p \in \mathbb{R}^{d_p \times d_m}, \tag{1}$$

where W_p is the ground-truth mixing dictionary. The training data contain samples of x_p, but not of x. We allow overcomplete dictionaries with $d_m > d_p$. The selected-support analysis below also applies to square and undercomplete dictionaries.

Assumption 1 (Blind sparse mixing)**.** The latent code and ground-truth dictionary satisfy the following conditions:

1. **Sparse latent code.** The latent code is k-sparse, with support
$$S = \mathrm{supp}(x), \qquad |S| = k \ll d_m, \tag{2}$$
and $k \le d_p$.

2. **Column normalization.** The ground-truth atoms are normalized to remove scaling ambiguity:
$$\|(W_p)_i\|_2 = 1, \qquad i = 1, \dots, d_m. \tag{3}$$
Dictionary recovery is understood after the usual column alignment.

We write $W_{p,S}$ for the columns of W_p indexed by S and $G_S = W_{p,S}^\top W_{p,S}$ for the corresponding Gram matrix. When a coherence-based bound is invoked, we use

$$\mu \triangleq \max_{i \neq j} |\langle (W_p)_i, (W_p)_j \rangle| .$$

Rank and conditioning requirements for the selected subdictionaries are stated explicitly in each result.

3.2 SPARSE AUTOENCODER WITH TIED WEIGHTS

We consider a tied-weight sparse autoencoder with encoder and decoder

$$E_{W_m, b_m}(x_p) = \sigma(W_m x_p + b_m), \qquad D_{W_m}(z) = W_m^\top z, \tag{4}$$

where $W_m \in \mathbb{R}^{d_m \times d_p}$, $b_m \in \mathbb{R}^{d_m}$, and σ is a sparsity-inducing activation such as ReLU, TopK, JumpReLU, or soft-thresholding. For an input x_p, we write

$$x_m \triangleq \sigma(W_m x_p + b_m), \qquad \hat{x}_p \triangleq W_m^\top x_m. \tag{5}$$

The latent self-map induced by one decode-encode cycle is

$$x_t \triangleq E_{W_m, b_m}(\hat{x}_p) = \sigma(W_m \hat{x}_p + b_m) = \sigma(W_m W_m^\top x_m + b_m). \tag{6}$$

We denote the first-pass support by

$$\hat{S} = \mathrm{supp}(x_m). \tag{7}$$

For $U \subseteq [d_m]$, let $W_{m,U} \in \mathbb{R}^{|U| \times d_p}$ be the row submatrix of W_m, so $W_{m,U}^\top$ is the learned decoder subdictionary on U. Define

$$G_{m,U} \triangleq W_{m,U} W_{m,U}^\top \in \mathbb{R}^{|U| \times |U|}. \tag{8}$$

Whenever $G_{m,U}$ is invertible, the orthogonal projection onto the selected decoder subspace and the corresponding least-squares coefficient are

$$P_U \triangleq W_{m,U}^\top G_{m,U}^{-1} W_{m,U}, \qquad x_{\mathrm{LS}}^U \triangleq G_{m,U}^{-1} W_{m,U} x_p. \tag{9}$$

Thus,

$$P_U x_p = W_{m,U}^\top x_{\mathrm{LS}}^U. \tag{10}$$

For a fixed support U, full column rank of $W_{m,U}^\top$ ensures that x_{LS}^U is unique. Uniqueness among representations with different supports requires additional assumptions.

Value-preserving activations. We say that σ is value-preserving on a support U for a pre-activation vector a if

$$\mathrm{supp}(\sigma(a)) = U, \qquad \sigma(a)_U = a_U. \tag{11}$$

This property means retaining the selected pre-activation values, which may still differ from the true latent coefficients. ReLU preserves positive pre-activations, while signed TopK preserves the selected values with their signs. Soft-thresholding changes the retained values by a threshold-dependent shift.

Correct support recovery means $\hat{S} = S$, whereas cycle support stability means $\mathrm{supp}(x_t) = \hat{S}$. Appendix D.1 gives sufficient margin conditions for correct first-pass support and its stability under small pre-activation perturbations.

3.3 Training objective and recovery notions

A standard SAE minimizes reconstruction error, optionally with a sparsity or structure-inducing penalty:

$$\min_{W_m, b_m} \ \mathbb{E}_{x_p}\left[\|x_p - \hat{x}_p\|_2^2\right] + \beta \mathbb{E}_{x_p}\left[\Omega(x_m)\right], \tag{12}$$

where Ω promotes sparsity and $\beta \geq 0$ controls its strength. For hard sparsity mechanisms such as TopK, Ω may be omitted.

The blind sparse mixing model distinguishes observation reconstruction $\hat{x}_p \approx x_p$, dictionary recovery $W_m^\top \approx W_p$, support recovery $\mathrm{supp}(x_m) \approx S$, and latent-code recovery $x_m \approx x$. Latent-code recovery is more fine-grained than support recovery: it requires not only identifying the active coordinates, but also assigning the correct amplitudes to them. The theoretical results below show that, for tied-weight SAEs, dictionary and support recovery alone do not generally certify faithful latent-code recovery of the encoder output.

4 Theoretical Results

4.1 A latent recovery gap beyond dictionary and support recovery

We begin with a simple tied-weight setting to illustrate the relation between dictionary, support, and coefficient recovery. Assuming an exact decoder dictionary and correct support recovery for a given input isolates the amplitude distortion introduced by the encoder. The following lemma shows that correlations among active atoms can produce coefficient errors even under these conditions.

Lemma 2 (Amplitude bias in tied SAE encoders)**.** *Consider the tied-weight SAE in equation 4. Suppose the decoder dictionary is exactly recovered, $W_m^\top = W_p$, and let $S = \mathrm{supp}(x)$. Assume that the activation is value-preserving on the recovered support and that $\mathrm{supp}(x_m) = S$. Then*

$$(x_m)_S = W_{p,S}^\top W_{p,S} x_S + (b_m)_S.$$

Consequently,

$$(x_m)_S - x_S = \left(W_{p,S}^\top W_{p,S} - I\right) x_S + (b_m)_S. \tag{13}$$

Thus, even exact decoder dictionary recovery and exact support recovery do not generally certify latent-code recovery by the tied encoder.

The proof is deferred to Appendix B.1.

Lemma 2 characterizes the amplitude error on samples for which dictionary and support recovery have succeeded. Exact coefficient recovery on such a sample requires $(G_S - I)x_S + (b_m)_S = 0$. Across samples with the same support, the bias contributes a constant shift, whereas $(G_S - I)x_S$ can vary with the latent coefficients. Appendix B.2 quantifies the population error that remains after optimizing a constant shift in this fixed-support affine model. For normalized, μ-coherent atoms,

$$\|(G_S - I)x_S\|_2 \leq (k-1)\mu \|x_S\|_2,$$

as shown in Appendix B.3.

In the noiseless model, exact reconstruction on the correct full-rank subdictionary uniquely determines the true coefficients. The remaining question is whether the one-pass tied encoder produces those coefficients. Sample-dependent coefficient errors can reverse the ranking of examples by a feature's activation, even when dictionary and support recovery are correct. Appendix D.2 gives an explicit example in which a fixed positive rescaling cannot correct the reversed ranking.

4.2 Reconstruction error separates subspace and coefficient errors

We use the standard least-squares decomposition to separate reconstruction error into a subspace residual and a coefficient discrepancy on the selected support.

Let $\hat{S} = \mathrm{supp}(x_m)$ and assume that $G_{m,\hat{S}}$ is invertible. The standard least-squares decomposition gives

$$x_p - W_{m,\hat{S}}^\top (x_m)_{\hat{S}} = \underbrace{(I - P_{\hat{S}})x_p}_{\text{subspace error}} + \underbrace{W_{m,\hat{S}}^\top \left(x_{\mathrm{LS}}^{\hat{S}} - (x_m)_{\hat{S}} \right)}_{\text{coefficient error}}. \tag{14}$$

Consequently,

$$\left\| x_p - W_{m,\hat{S}}^\top (x_m)_{\hat{S}} \right\|_2^2 = \|(I - P_{\hat{S}})x_p\|_2^2 + \left\| W_{m,\hat{S}}^\top \left(x_{\mathrm{LS}}^{\hat{S}} - (x_m)_{\hat{S}} \right) \right\|_2^2. \tag{15}$$

A derivation is given in Appendix B.4.

The first term in equation 15 measures how well the selected decoder atoms span the observation. The second term measures the LS coefficient discrepancy through the selected decoder. If the observation lies in the selected span, the first term vanishes and reconstruction error consists entirely of the decoder-weighted coefficient term.

Under exact dictionary and support recovery in the noiseless model,

$$W_m^\top = W_p, \qquad \hat{S} = S, \qquad x_p = W_{p,S} x_S,$$

the subspace error vanishes and $x_{\mathrm{LS}}^S = x_S$. More generally, on the correct support $\hat{S} = S$, suppose

$$W_{m,S}^\top = W_{p,S} + \Delta_S, \qquad \|\Delta_S\|_{\mathrm{op}} \le \delta, \qquad x_p = W_{p,S} x_S + \xi.$$

Then

$$\|(I - P_S)x_p\|_2 \le \delta \|x_S\|_2 + \|\xi\|_2. \tag{16}$$

The derivation is given in Appendix B.4. This bound controls the subspace contribution through decoder mismatch and observation noise. The relation between the decoder-weighted coefficient term and Euclidean coefficient error depends on the conditioning of the selected decoder. Proposition 4 further accounts for decoder mismatch and noise when bounding error relative to the true code.

4.3 Self-consistency as coefficient-level latent refinement

We now show that the latent self-consistency residual directly targets the coefficient-level gap identified above. Recall the decode-encode code x_t from equation 6, and define

$$\mathcal{L}_{\mathrm{sc}} = \|x_t - x_m\|_2^2. \tag{17}$$

We consider the stable-support regime $\mathrm{supp}(x_t) = \hat{S}$ and assume that the activation is value-preserving on $\hat{S}$ for both the first encoder pass and the decode-encode pass.

Proposition 3 (Self-consistency characterizes the selected-support LS gap)**.** *Assume that $G_{m,\hat{S}}$ is invertible. Under the stable-support and value-preserving conditions above,*

$$(x_t - x_m)_{\hat{S}} = G_{m,\hat{S}} \left((x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}} \right). \tag{18}$$

Consequently, if $\alpha I \preceq G_{m,\hat{S}} \preceq \beta I$, then

$$\frac{1}{\beta} \left\| (x_t - x_m)_{\hat{S}} \right\|_2 \le \left\| (x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}} \right\|_2 \le \frac{1}{\alpha} \left\| (x_t - x_m)_{\hat{S}} \right\|_2. \tag{19}$$

The proof is deferred to Appendix B.5.

Relation to the reconstruction objective. Let $e = (x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}}$ and $G = G_{m,\hat{S}}$. Under Proposition 3, the coefficient term in reconstruction loss and the self-consistency loss satisfy

$$E_{\mathrm{coef}} = e^\top G e, \qquad \mathcal{L}_{\mathrm{sc}} = e^\top G^2 e.$$

If $0 < \alpha I \preceq G \preceq \beta I$, then

$$\alpha E_{\mathrm{coef}} \le \mathcal{L}_{\mathrm{sc}} \le \beta E_{\mathrm{coef}}.$$

Adding self-consistency with weight $\lambda \ge 0$ changes the coefficient weighting from G to $G + \lambda G^2$, while leaving the orthogonal subspace term unchanged. For a fixed decoder and support, these quadratic forms have the same LS minimizer. During training, a shared encoder couples the codes across samples, so the additional term changes how their coefficient discrepancies are weighted. Appendix D.3 gives the conditioning comparison and the corresponding masked formulation.

Proposition 4 (Local latent recovery bound)**.** *Suppose the inferred and decode-encode supports are correct, $\hat{S} = \mathrm{supp}(x_t) = S$, and the activation is value-preserving on S for both encoder passes. Assume that $G_{m,S}$ is invertible. Assume $W_{m,S}^\top = W_{p,S} + \Delta_S$ with $\|\Delta_S\|_{\mathrm{op}} \le \delta$, and $x_p = W_{p,S} x_S + \xi$. Then*

$$(x_m)_S - x_S = G_{m,S}^{-1}(x_t - x_m)_S + G_{m,S}^{-1} W_{m,S}(\xi - \Delta_S x_S). \tag{20}$$

Consequently, if the learned active Gram matrix is (α, β)-well-conditioned, $\alpha I \preceq G_{m,S} \preceq \beta I$, then

$$\|(x_m)_S - x_S\|_2 \le \frac{1}{\alpha}\|(x_t - x_m)_S\|_2 + \frac{\sqrt{\beta}}{\alpha}\big(\|\xi\|_2 + \delta\|x_S\|_2\big). \tag{21}$$

The proof is deferred to Appendix B.6.

The decomposition in equation 15 requires only a full-rank selected subdictionary, while Proposition 3 additionally requires value preservation and the same support on both encoder passes. Under correct support and these activation conditions, Proposition 4 bounds true-code error in terms of the self-consistency residual, decoder mismatch, observation noise, and active-Gram conditioning. This is a bound at fixed model parameters; the effect of regularized training is evaluated empirically. For hard TopK, the score-level masked residual in Section 5 retains the selected-support LS identity without requiring the second pass to select the same support (Appendix C.1).

4.4 Untied encoders and local inverse consistency

We next allow the encoder and decoder to have distinct weights. Let $E \in \mathbb{R}^{d_m \times d_p}$ and $D \in \mathbb{R}^{d_p \times d_m}$ denote the encoder and decoder matrices. For a support U, let E_U contain the corresponding rows of E and D_U the corresponding columns of D.

Proposition 5 (Untied self-consistency and local recovery)**.** *Let*

$$x_m = \sigma(E x_p + b_m), \qquad x_t = \sigma(E D x_m + b_m)$$

have the same support U, with value-preserving activations on both passes. Suppose $H_U = E_U D_U$ is invertible, and define

$$s_U = (x_t - x_m)_U, \qquad x_{\mathrm{inv}}^U = H_U^{-1} E_U x_p.$$

Then

$$\begin{aligned} s_U &= H_U\big((x_m)_U - x_{\mathrm{inv}}^U\big), \\ \|(x_m)_U - x_{\mathrm{inv}}^U\|_2 &\le \frac{\|s_U\|_2}{\sigma_{\min}(H_U)}. \end{aligned} \tag{22}$$

If $U = S$ is the true support and $x_p = D_S x_S + q$, then

$$\|(x_m)_S - x_S\|_2 \le \frac{\|s_S\|_2 + \|E_S\|_{\mathrm{op}}\|q\|_2}{\sigma_{\min}(H_S)}. \tag{23}$$

The proof follows from the identities in Appendix C.2. In the sparse mixing model, $D_S = W_{p,S} + \Delta_S$ and $x_p = W_{p,S} x_S + \xi$ give $q = \xi - \Delta_S x_S$. When $E = D^\top$, the inverse code equals the decoder LS code. For general untied weights, the inverse code also depends on the encoder and need not coincide with the decoder LS code. The true-code bound depends on both $\sigma_{\min}(H_S)$ and the perturbation amplification factor $\|E_S\|_{\mathrm{op}}/\sigma_{\min}(H_S)$.

5 ALGORITHM

Motivated by the local coefficient-error bounds in Section 4, we add a self-consistency penalty to the SAE training objective.

$$x_t \triangleq \sigma(W_m \hat{x}_p + b_m) = \sigma(W_m W_m^\top x_m + b_m) =: F_{W_m,b_m}(x_m), \tag{24}$$

where F_{W_m,b_m} is the latent self-map induced by the tied SAE. For activations with stable support, we define the latent self-consistency loss as

$$\mathcal{L}_{\text{sc}} \triangleq \mathbb{E}_{x_p}\left[\|x_t - x_m\|_2^2\right]. \tag{25}$$

This unmasked form is used in the controlled tied TopK-ReLU study to test the stable-support mechanism in Propositions 3 and 4.

Masked self-consistency for hard TopK encoders. For the signed hard TopK encoder used in our real-model experiments, we compare pre-TopK scores on the first-pass support to accommodate support changes after re-encoding. Let

$$a = W_m x_p + b_m, \qquad x_m = \text{TopK}_{|\cdot|,K}(a), \qquad a_t = W_m \hat{x}_p + b_m,$$

where $\text{TopK}_{|\cdot|,K}$ retains the K largest-magnitude entries and preserves their signs. Writing $\hat{S} = \text{supp}(x_m)$, we have $(x_m)_{\hat{S}} = a_{\hat{S}}$ and define

$$\mathcal{L}_{\text{sc}}^{\hat{S}} = \mathbb{E}_{x_p}\left[|\hat{S}|^{-1}\|(a_t - a)_{\hat{S}}\|_2^2\right]. \tag{26}$$

When $G_{m,\hat{S}}$ is invertible, the tied model satisfies

$$(a_t - a)_{\hat{S}} = G_{m,\hat{S}}\left((x_m)_{\hat{S}} - x_{\text{LS}}^{\hat{S}}\right).$$

Thus, the masked loss penalizes a Gram-weighted coefficient discrepancy on the selected support without requiring the re-encoded TopK support to equal $\hat{S}$ (Appendix C.1). Scores outside $\hat{S}$ do not contribute directly to this loss. Since TopK enforces sparsity, we omit an additional sparsity penalty and train with

$$\min_{W_m,b_m} \mathbb{E}_{x_p}\left[\|x_p - \hat{x}_p\|_2^2\right] + \lambda \mathcal{L}_{\text{sc}}^{\hat{S}}. \tag{27}$$

For non-TopK activations, the unmasked term $\mathcal{L}_{\text{sc}}$ can be added to the standard SAE objective, together with a sparsity penalty when applicable.

Untied SAEs. The same construction applies to the untied SAEs analyzed in Section 4. With encoder E, decoder D, and bias b_m, let

$$x_m = \sigma(E x_p + b_m), \qquad \hat{x}_p = D x_m, \qquad x_t = \sigma(E \hat{x}_p + b_m).$$

The unmasked loss uses $x_t - x_m$, while the masked TopK loss uses $a = E x_p + b_m$ and $a_t = E \hat{x}_p + b_m$ in equation 26. The corresponding training objective is optimized over (E, D, b_m). The untied analysis relates the residual to an encoder-induced inverse code through the active encoder-decoder cross-Gram matrix (Appendix C.2). Both variants require one additional encoder evaluation during training, introduce no auxiliary parameters, and leave inference unchanged.

6 EXPERIMENTS

6.1 EVALUATION PROTOCOL

On synthetic mixtures, we evaluate latent L_2 error after dictionary matching and scale correction, together with support F1 and mean matched decoder cosine (MCC). We also measure the selected-support coefficient error E_{coef} from the reconstruction decomposition and amplitude MSE on samples with exactly recovered support. MCC and support F1 track dictionary and support recovery, while E_{coef} measures the decoder-weighted gap to the least-squares code on the inferred support. Real-model evaluations use CE-bench (Gulko et al., 2025) and selected SAEBench evaluations (Karvonen et al., 2025) to assess feature utility, where ground-truth latent codes are unavailable. Appendix E gives the settings, alignment procedure, metrics, and weight-selection protocol; Appendix F reports the full results.

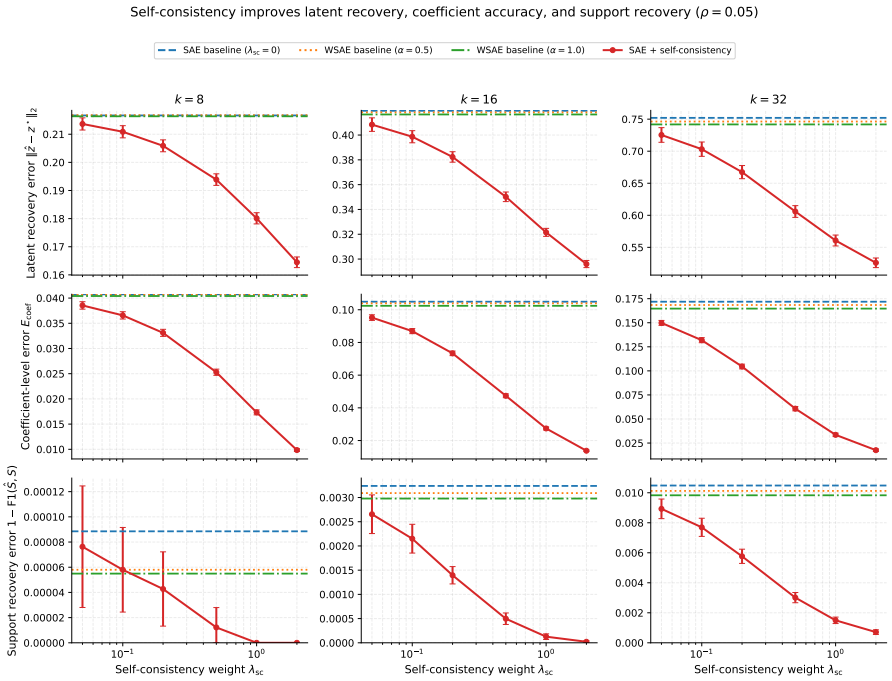

Figure 1: **Self-consistency in tied TopK-ReLU SAEs.** We sweep λ_{sc} for $k \in \{8, 16, 32\}$. Rows show latent L_2 error, E_{coef}, and support error $1 - \text{F1}$. In the plot, $\hat{z}$ and $z^\star$ denote the inferred and true codes. Results average ten seeds, with standard-deviation error bars. Horizontal lines show reconstruction-only and WSAE baselines.

6.2 Synthetic recovery and coefficient refinement

Tied encoders. We generate observations from a column-normalized dictionary with controllable correlation and exactly k-sparse nonnegative codes, whose active coefficients are sampled uniformly from $[0.5, 1.0]$. A tied TopK-ReLU SAE is initialized near the generating dictionary and trained with reconstruction loss plus the unmasked self-consistency penalty. This initialization places the experiment in the local refinement regime of the analysis, with the decoder still learned during training. We sweep λ_{sc} and compare with reconstruction-only training and our WSAE reimplementation (Cui et al., 2025). Figure 1 shows lower latent recovery error and coefficient error with self-consistency than with reconstruction-only training. Support recovery also improves, with larger changes at higher sparsities; at $k = 8$, support recovery is already nearly saturated. The reduction in E_{coef} indicates a smaller discrepancy from the least-squares code on the selected support, consistent with the mechanism in Section 4. Together with the nearly saturated support scores at $k = 8$, this supports coefficient refinement as a source of the latent-error improvement.

Untied and fixed-decoder controls. Table 1 summarizes coefficient recovery under three settings. In the controlled untied experiments, decoder MCC and support F1 remain approximately 1.000, while the normalized active cross-map error $\|H_S - I\|_F^2/k$ decreases by 96.1–99.6% across the tested sparsities. True latent error and the selected-support coefficient error decrease alongside this change. A separate control freezes the decoder at the ground-truth dictionary and has support F1 equal to one for both methods. Self-consistency reduces latent L_2 error by 69.62% in this setting, demonstrating coefficient improvement with the decoder fixed and support already correct. For the tied comparisons in the table, coefficient improvements accompany a small decrease in decoder MCC and a 1.43–4.66% increase in reconstruction MSE. Appendix F provides these diagnostics, correlation sweeps, and the explicitly overcomplete control.

Setting	Latent L_2 reduction	E_{coef} reduction
Tied, near-correct ($k = 8, 16, 32$)	1.97–16.36%	10.24–63.83%
Untied, controlled ($k = 8, 16, 32$)	55.2–77.6%	59.3–88.8%
Fixed ground-truth decoder	69.62%	90.70%

Table 1: **Synthetic recovery controls.** Relative error reductions against reconstruction-only training, averaged over ten paired seeds. Ranges span the tested sparsities and are not uncertainty intervals. The fixed-decoder row is a separate control. Dictionary, support, amplitude, and reconstruction measurements are reported in Appendix F.

Evaluation	Metric	Mean change (pp)
SAE-Probes	Accuracy, $k = 1/2/5$	$+0.118$ / $+0.101$ / $+0.024$
	AUC, $k = 1/2/5$	$+0.121$ / $+0.101$ / $+0.101$
	F1, $k = 1/2/5$	$+0.175$ / $+0.089$ / $+0.239$
AutoInterp (custom judge)	Score	$+0.505$

Table 2: **Feature utility at the common weight $\lambda = 10^{-5}$.** Entries are mean percentage-point changes from paired unregularized checkpoints. For SAE-Probes, k denotes the number of features used by the probe. AutoInterp uses the fixed custom-judge protocol described in Appendix E.

6.3 Real-model feature quality

CE-bench. We train untied signed TopK SAEs of width 2048 and $K = 32$ on GPT-2 residual-stream activations, using five matched seeds. Contrastive measures view-specific signal, while Independence measures separation of shared and contrasting components. With the masked self-consistency penalty at $\lambda = 0.5$, Contrastive and Independence improve by 3.54% and 2.59%, respectively, yielding a 3.07% increase in CE total. The dead-feature fraction decreases from 0.1599 to 0.0286, usage entropy increases by 4.34%, and coherence RMS decreases by 3.57%, indicating broader, more balanced feature usage and less correlated decoder atoms. The active fraction is fixed across methods, so these changes occur at the same activation budget. Additional evaluations at GPT-2 layers 2 and 6 yield CE-total gains of 1.55% and 1.07%, respectively; full results appear in Appendix F.

Probing and automated interpretability. For the SAEBench evaluations, we use untied signed TopK SAEs of width 2048 and $K = 8$. We select $\lambda = 10^{-5}$ using Legacy Sparse Probing and keep it fixed for SAE-Probes, AutoInterp, and CORE. Sparse probing evaluates task-relevant information accessible through a few features, while AutoInterp assesses how well feature descriptions predict activation patterns. Table 2 reports positive mean changes in accuracy, AUC, and F1 across 113 SAE-Probes tasks, together with a positive mean AutoInterp change. SAE-Probes uses four paired training seeds; AutoInterp uses three paired seeds and a fixed DeepSeek-V4-Flash judge. The same operating point retains near-baseline CORE scores. The complete probing and AutoInterp weight sweeps, CORE results, and seed-aggregation details appear in Appendices E and F.

7 Conclusion

We identify sample-dependent amplitude errors in one-pass tied SAE encoders, even when dictionary and support recovery are exact. For tied and untied SAEs, we establish local bounds relating the decode-encode self-consistency residual to coefficient error, accounting for dictionary mismatch, observation noise, and conditioning. These results motivate a self-consistency regularizer that adds one encoder evaluation during training and leaves inference unchanged. Controlled synthetic experiments demonstrate improved coefficient recovery, while CE-bench and SAEBench evaluations provide complementary evidence of improved feature quality.

AI USE STATEMENT

We used generative AI tools as coding assistants when implementing and modifying the codebase, and for language editing. They were not used to produce research results on their own or to replace the authors' scientific judgment. The authors designed the methodology, derived and verified the theoretical results, designed the experiments, and reviewed and validated all AI-assisted code and text. Separately, a large language model (DeepSeek-V4-Flash) serves as a fixed judge inside the AutoInterp evaluation of SAEBench (Karvonen et al., 2025). In that role it is part of the scoring protocol, is held fixed across all compared checkpoints, and is reported as a custom-judge evaluation (Appendix E).

ETHICS STATEMENT

This work is primarily theoretical and methodological, focusing on coefficient recovery in sparse autoencoders and on a self-consistency regularizer for SAE training. Our controlled experiments use synthetic data generated from a sparse mixing model. Our real-model experiments use activations of the publicly available GPT-2 model (Radford et al., 2019) and the public CE-bench (Gulko et al., 2025) and SAEBench (Karvonen et al., 2025) evaluation suites. No human subjects are involved, and we introduce no new pretrained models or datasets. Our goal is to make interpretability tools more faithful. As discussed in Appendix H, however, more reliable SAE codes could create unwarranted confidence if our local, near-correct-regime guarantees are applied outside their assumptions. We state these assumptions explicitly with each result (Appendix D) and discuss them further in Appendix G.

REPRODUCIBILITY STATEMENT

Proofs of the main tied-weight results are given in Appendix B, with masked TopK and untied extensions in Appendices C.1 and C.2. Local conditions and supporting examples are collected in Appendix D. Synthetic settings, alignment procedures, and recovery metrics are described in Appendices E.1 and E.2. Our WSAE baseline is a best-effort reimplementation of the published algorithmic description in Cui et al. (2025). Appendix E.3 describes the GPT-2 activation and benchmark evaluation protocols. Appendix E.4 gives the CE-bench regularization schedule, SAEBench weight selection, and reporting conventions; additional results are provided in Appendix F. We will release the experimental code upon acceptance.

REFERENCES

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding, 2015. URL https://arxiv.org/abs/1503.00778.

Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010. doi: 10.1109/TIT.2010.2040894.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2(5):6, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

Emmanuel J. Candès. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*, volume 3, pp. 1433–1452. European Mathematical Society, 2006. doi: 10.4171/022-3/69. URL https://ems.press/books/standalone/24/561.

Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes rendus. Mathematique*, 346(9-10):589–592, 2008.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Albert Cohen, Ingrid Daubechies, and Jean-Christophe Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992. doi: 10.1002/cpa.3160450502.

Jingyi Cui, Qi Zhang, Yifei Wang, and Yisen Wang. On the theoretical understanding of identifiable sparse autoencoders and beyond. *arXiv preprint arXiv:2506.15963*, 2025.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

Marco F Duarte and Yonina C Eldar. Structured compressed sensing: From theory to applications. *IEEE Transactions on signal processing*, 59(9):4053–4085, 2011.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Thomas Fel, Ekdeep Singh Lubana, Jacob S Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba Ba, and Talia Konkle. Archetypal sae: Adaptive and stable dictionary learning for concept extraction in large vision models. *arXiv preprint arXiv:2502.12892*, 2025a.

Thomas Fel, Binxu Wang, Michael A Lepori, Matthew Kowal, Andrew Lee, Randall Balestriero, Sonia Joseph, Ekdeep S Lubana, Talia Konkle, Demba Ba, et al. Into the rabbit hull: From task-relevant concepts in dino to minkowski geometry. *arXiv preprint arXiv:2510.08638*, 2025b.

Marco Fumero, Luca Moschella, Emanuele Rodolà, and Francesco Locatello. Navigating the latent space dynamics of neural models. *arXiv preprint arXiv:2505.22785*, 2025.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Rémi Gribonval and Karin Schnass. Dictionary identification—sparse matrix-factorization via l1-minimization. *IEEE Transactions on Information Theory*, 56(7):3523–3539, 2010.

Alex Gulko, Yusen Peng, and Sachin Kumar. Ce-bench: Towards a reliable contrastive evaluation benchmark of interpretability of sparse autoencoders. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 1–15, 2025.

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

Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry. *arXiv preprint arXiv:2503.01822*, 2025. URL https://arxiv.org/abs/2503.01822.

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

Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19, 2006.

Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. *arXiv preprint arXiv:2412.05276*, 2024.

Miles Lopes. Estimating unknown sparsity in compressed sensing. In *International Conference on Machine Learning*, pp. 217–225. PMLR, 2013.

Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models, 2025. URL https://arxiv.org/abs/2403.19647.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

Thanh V. Nguyen, Raymond K. W. Wong, and Chinmay Hegde. On the dynamics of gradient descent for autoencoders. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 2019.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004.

Charles O'Neill, Alim Gumran, and David Klindt. Compute optimal inference and provable amortisation gap in sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46877–46896. PMLR, 2025. URL https://proceedings.mlr.press/v267/o-neill25a.html.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. *arXiv preprint arXiv:2504.02821*, 2025.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Akshay Rangamani, Anirbit Mukherjee, Amitabh Basu, Tejaswini Ganapathy, Ashish Arora, Sang Chin, and Trac D. Tran. Sparse coding and autoencoders, 2017. URL https://arxiv.org/abs/1708.03735.

Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

Adam Scherlis, Kshitij Sachan, Adam S Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks. *arXiv preprint arXiv:2210.01892*, 2022.

Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Conference on Learning Theory*, pp. 37–1. JMLR Workshop and Conference Proceedings, 2012.

Pranay Tankala, Abiy Tasissa, James M. Murphy, and Demba Ba. K-Deep Simplex: Deep manifold learning via local dictionaries. *arXiv preprint arXiv:2012.02134v4*, 2024. URL https://arxiv.org/abs/2012.02134v4.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Bahareh Tolooshams and Demba Ba. Stable and interpretable unrolled dictionary learning, 2022. URL https://arxiv.org/abs/2106.00058.

Ivana Tošić and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2): 27–38, 2011.

J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004. doi: 10.1109/TIT.2004.834793.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025.

A Related Work

Polysemanticity and superposition. A recent line of work argues that neural networks encode far more features than they have neurons or dimensions by representing concepts in superposition. In this view, individual neurons or directions are generally *polysemantic*, participating in multiple unrelated features rather than corresponding to single interpretable concepts (Olah et al., 2018; Elhage et al., 2022; Scherlis et al., 2022; Gurnee et al., 2023; Mu & Andreas, 2020). This phenomenon has been observed across modalities and architectures, and poses a central challenge for mechanistic interpretability: understanding model behavior requires disentangling many overlapping features that are not explicitly represented in the basis of the network.

Sparse autoencoders for interpretability. Sparse autoencoders (SAEs) (Ng et al., 2011; Makhzani & Frey, 2013) have recently re-emerged as a practical tool for interpreting learned representations by attempting to recover approximately monosemantic features from polysemantic activations. Applied to large language models (LLMs) (Cunningham et al., 2023), vision models (Fel et al., 2025a), and multimodal systems (Pach et al., 2025; Lim et al., 2024), SAEs have been shown to extract human-interpretable directions corresponding to concepts, circuits, or attributes (Cunningham et al., 2023; Bricken et al., 2023; Gao et al., 2024; Fel et al., 2025a). A variety of encoder parameterizations and sparsity mechanisms have been explored, including ReLU activations with ℓ_1 penalties (Cunningham et al., 2023), hard Top-K activations (Gao et al., 2024; Bussmann et al., 2024), and gated or structured units (Rajamanoharan et al., 2024). Most prior work evaluates SAEs primarily through reconstruction fidelity, sparsity statistics, or downstream interpretability benchmarks (Karvonen et al., 2025). These evaluations are essential for assessing representation quality, but they do not directly certify whether the SAE encoder recovers the amplitudes of the underlying latent features.

Identifiability and latent recovery. Despite empirical success, recent theoretical work has highlighted fundamental limitations of SAE-based interpretation. In particular, even under idealized conditions, such as perfect reconstruction of observations and recovery of the ground-truth dictionary, there is no general guarantee that the learned latent codes correspond to the true underlying factors (Cui et al., 2025). Related empirical work has also observed instability of SAE solutions under retraining (Paulo & Belrose, 2025; Fel et al., 2025a). These results emphasize a distinction between *dictionary recovery* and *latent-code recovery*, and show that reconstruction-based objectives alone are insufficient to certify that an SAE has learned the intended concepts. Our work builds on this line of inquiry, but focuses on a complementary mechanism: coefficient-level amplitude bias induced by the SAE encoder. We show that even when the decoder dictionary and active support are recovered, the one-pass tied encoder can return biased amplitudes, and we propose a latent self-consistency regularizer that targets the selected-support least-squares coefficient gap.

Sparse coding and dictionary learning. SAEs are closely related to classical sparse coding and dictionary learning methods (Rubinstein et al., 2010; Tibshirani, 1996; Olshausen & Field, 2004; Lee et al., 2006; Tošić & Frossard, 2011; Kreutz-Delgado et al., 2003). These approaches aim to represent data as linear combinations of a small number of dictionary atoms and have been extensively studied in signal processing and machine learning. In linear settings, strong theoretical guarantees exist for sparse approximation, support recovery, and stability under assumptions such as sparsity, incoherence, or restricted isometry (Tropp, 2004; Spielman et al., 2012; Gribonval & Schnass, 2010; Candes, 2008; Donoho, 2006; Candès, 2006; Duarte & Eldar, 2011; Lopes, 2013). More recent analyses connect these ideas to neural or autoencoder-style inference, studying dictionary recovery, support recovery, or convergence of sparse-coding autoencoders and unrolled dictionary-learning algorithms under suitable initialization, thresholding, or model assumptions (Arora et al., 2015; Rangamani et al., 2017; Nguyen et al., 2019; Tolooshams & Ba, 2022). However, many of these guarantees concern explicit inference procedures, such as greedy selection, convex sparse recovery, or least-squares refitting on a recovered support. Our focus is different: we analyze the code produced by the SAE encoder itself. In tied-weight SAEs, this encoder returns projection coefficients, which can remain Gram-distorted even when the decoder dictionary and active support are correct. Thus, our results do not contradict classical sparse recovery guarantees; rather, they identify an encoder-induced gap between support recovery and faithful latent-code recovery in SAE-style inference.

Structured sparse inference and amortization. Model-based compressed sensing uses support structure to refine classical sparse-recovery guarantees (Baraniuk et al., 2010; Duarte & Eldar, 2011). These results concern a generative model together with an inference procedure; they do not imply that

a tied transpose encoder computes the recovered coefficients. O'Neill et al. (2025) show limitations of linear-nonlinear SAE inference and evaluate alternatives with greater inference expressivity. Our analysis characterizes the relation between the SAE cycle residual and coefficient error on the selected support, motivating a training regularizer that retains one-pass inference.

Analysis-synthesis geometry. Distinct analysis and synthesis operators are classical, including in biorthogonal wavelets (Cohen et al., 1992). K-Deep Simplex learns local dictionaries with simplex-supported representations and an unrolled inference architecture (Tankala et al., 2024). Work on SAE concept geometry further emphasizes that architecture determines which latent structures can be represented (Hindupur et al., 2025). Building on these analysis-synthesis perspectives, we relate the untied SAE cycle residual to coefficient estimation through the active cross-map $E_U D_U$.

B Proofs and additional details

B.1 Proof of the latent recovery gap and compatibility with exact reconstruction

Proof of Lemma 2. Since $W_m^\top = W_p$, we have $W_m = W_p^\top$. Therefore,

$$x_m = \sigma(W_m x_p + b_m) = \sigma(W_p^\top W_p x + b_m).$$

Because x is supported on S, we have

$$W_p x = W_{p,S} x_S,$$

and hence

$$W_p^\top W_p x = W_p^\top W_{p,S} x_S.$$

Restricting to coordinates in S, and using the value-preserving assumption on the recovered support, gives

$$(x_m)_S = W_{p,S}^\top W_{p,S} x_S + (b_m)_S.$$

Subtracting x_S from both sides gives

$$(x_m)_S - x_S = (W_{p,S}^\top W_{p,S} - I) x_S + (b_m)_S,$$

which proves the claim. □

Compatibility with exact reconstruction. If exact reconstruction is additionally imposed together with exact dictionary and support recovery, then

$$x_p = \hat{x}_p = W_{p,S}(x_m)_S, \qquad x_p = W_{p,S} x_S.$$

Hence

$$W_{p,S}\big((x_m)_S - x_S\big) = 0.$$

If $W_{p,S}$ has full column rank, then $(x_m)_S = x_S$. Thus, Lemma 2 does not contradict coefficient uniqueness under exact reconstruction; it shows that the tied encoder output generally need not satisfy this compatibility condition.

B.2 Population-level amplitude bias on a fixed support

The single-sample obstruction in Lemma 2 cannot generally be removed by choosing a fixed encoder bias when the latent amplitudes vary across samples. We formalize this on a fixed support S.

The infimum is taken within the fixed-support, value-preserving affine model; it does not assert that arbitrary changes of the bias preserve the recovered support.

Proposition 6 (Population amplitude bias on a fixed support)**.** *Consider the setting of Lemma 2 on a fixed support S, so that the support is recovered as S and the activation is value-preserving on S for all samples under consideration. Let $G_S = W_{p,S}^\top W_{p,S}$, and suppose x_S has mean $m_S = \mathbb{E}[x_S]$ and covariance $\Sigma_S = \mathbb{E}[(x_S - m_S)(x_S - m_S)^\top]$. Optimizing over any fixed on-support encoder bias $\bar{b}_S$, we have*

$$\inf_{\bar{b}_S} \mathbb{E}\left[\left\|G_S x_S + \bar{b}_S - x_S\right\|_2^2\right] = \mathrm{Tr}\big((G_S - I)\Sigma_S(G_S - I)^\top\big). \tag{28}$$

The minimizer is $\bar{b}_S^\star = -(G_S - I)m_S$. *Equivalently, under the tied encoder in Lemma 2,*

$$\inf_{\bar{b}_S} \mathbb{E}\|(x_m)_S - x_S\|_2^2 = \mathrm{Tr}\big((G_S - I)\Sigma_S(G_S - I)^\top\big)\,. \tag{29}$$

Moreover, if $\lambda_- I \preceq \Sigma_S \preceq \lambda_+ I$, *then*

$$\lambda_-\|G_S - I\|_F^2 \le \inf_{\bar{b}_S} \mathbb{E}\|(x_m)_S - x_S\|_2^2 \le \lambda_+\|G_S - I\|_F^2. \tag{30}$$

Proof. Let $B_S = G_S - I$. By Lemma 2, allowing an arbitrary fixed on-support bias $\bar{b}_S$, the on-support latent error is

$$(x_m)_S - x_S = B_S x_S + \bar{b}_S.$$

Therefore, for any fixed bias $\bar{b}_S$,

$$\mathbb{E}\|(x_m)_S - x_S\|_2^2 = \mathbb{E}\|B_S x_S + \bar{b}_S\|_2^2.$$

Write $x_S = m_S + \varepsilon_S$, where $\mathbb{E}[\varepsilon_S] = 0$ and $\mathbb{E}[\varepsilon_S\varepsilon_S^\top] = \Sigma_S$. Then

$$B_S x_S + \bar{b}_S = B_S\varepsilon_S + (B_S m_S + \bar{b}_S).$$

Since $\mathbb{E}[B_S\varepsilon_S] = 0$, the cross term vanishes, and hence

$$\mathbb{E}\|B_S x_S + \bar{b}_S\|_2^2 = \mathbb{E}\|B_S\varepsilon_S\|_2^2 + \|B_S m_S + \bar{b}_S\|_2^2.$$

The second term is minimized by $\bar{b}_S^\star = -B_S m_S = -(G_S - I)m_S$. At this optimum,

$$\inf_{\bar{b}_S} \mathbb{E}\|B_S x_S + \bar{b}_S\|_2^2 = \mathbb{E}\|B_S\varepsilon_S\|_2^2 = \mathrm{Tr}\big(B_S\Sigma_S B_S^\top\big)\,.$$

Substituting $B_S = G_S - I$ proves equation 28. If $\lambda_- I \preceq \Sigma_S \preceq \lambda_+ I$, then

$$\lambda_-\,\mathrm{Tr}(B_S B_S^\top) \le \mathrm{Tr}(B_S\Sigma_S B_S^\top) \le \lambda_+\,\mathrm{Tr}(B_S B_S^\top).$$

Since $\mathrm{Tr}(B_S B_S^\top) = \|B_S\|_F^2$, this gives equation 30. □

B.3 Coherence control of the Gram-induced amplitude bias

We record the coherence-based bounds used in Section 4.1. Assume the columns of W_p are normalized, and let

$$\mu := \max_{i\neq j} |\langle (W_p)_i, (W_p)_j\rangle|\,.$$

For a fixed support S with $|S| = k$, the active Gram matrix $G_S = W_{p,S}^\top W_{p,S}$ has unit diagonal. Therefore, $G_S - I$ has zero diagonal and off-diagonal entries bounded in magnitude by μ.

Operator-norm bound. For each row of $G_S - I$, the sum of the absolute values of the off-diagonal entries is at most $(k-1)\mu$. Since $G_S - I$ is symmetric, Gershgorin's theorem implies that every eigenvalue of $G_S - I$ lies in $[-(k-1)\mu, (k-1)\mu]$. Hence

$$\|G_S - I\|_{\mathrm{op}} \le (k-1)\mu.$$

Consequently, for any x_S,

$$\|(G_S - I)x_S\|_2 \le \|G_S - I\|_{\mathrm{op}}\|x_S\|_2 \le (k-1)\mu\|x_S\|_2.$$

Frobenius-norm bound. Since $G_S - I$ has $k(k-1)$ off-diagonal entries and each is bounded in magnitude by μ,

$$\|G_S - I\|_F^2 = \sum_{i\neq j,\, i,j\in S} |\langle (W_p)_i, (W_p)_j\rangle|^2 \le k(k-1)\mu^2.$$

Combining this with Proposition 6, or equivalently with equation 30, yields

$$\inf_{\bar{b}_S} \mathbb{E}\|(x_m)_S - x_S\|_2^2 \le \lambda_+ k(k-1)\mu^2$$

whenever $\Sigma_S \preceq \lambda_+ I$. Thus, sparsity and incoherence reduce the tied-projection amplitude bias: the single-sample bound scales with $(k-1)\mu$, and the population upper bound scales with $k(k-1)\mu^2$. These bounds vanish under the strong condition $\mu = 0$, which implies $G_S = I$ for every support S; more generally, for a fixed support, the actual bias disappears only when the active Gram matrix acts as the identity on the relevant latent amplitudes, e.g., $G_S = I$ when the covariance on S is full rank.

B.4 Derivation of the subspace-coefficient decomposition

Derivation of equation 15. By the definitions of $P_{\hat{S}}$ and $x_{\mathrm{LS}}^{\hat{S}}$, we have

$$P_{\hat{S}} x_p = W_{m,\hat{S}}^{\top} x_{\mathrm{LS}}^{\hat{S}}.$$

Therefore,

$$\begin{aligned} x_p - W_{m,\hat{S}}^{\top}(x_m)_{\hat{S}} &= (I - P_{\hat{S}})x_p + P_{\hat{S}}x_p - W_{m,\hat{S}}^{\top}(x_m)_{\hat{S}} \\ &= (I - P_{\hat{S}})x_p + W_{m,\hat{S}}^{\top} x_{\mathrm{LS}}^{\hat{S}} - W_{m,\hat{S}}^{\top}(x_m)_{\hat{S}} \\ &= (I - P_{\hat{S}})x_p + W_{m,\hat{S}}^{\top}\left(x_{\mathrm{LS}}^{\hat{S}} - (x_m)_{\hat{S}}\right). \end{aligned}$$

This proves the residual decomposition.

It remains to verify orthogonality. Since $P_{\hat{S}}$ is the orthogonal projector onto

$$\mathrm{span}(W_{m,\hat{S}}^{\top}),$$

the term

$$(I - P_{\hat{S}})x_p$$

lies in

$$\mathrm{span}(W_{m,\hat{S}}^{\top})^{\perp}.$$

On the other hand,

$$W_{m,\hat{S}}^{\top}\left(x_{\mathrm{LS}}^{\hat{S}} - (x_m)_{\hat{S}}\right) \in \mathrm{span}(W_{m,\hat{S}}^{\top}).$$

Hence the two terms are orthogonal. Applying the Pythagorean theorem gives

$$\left\| x_p - W_{m,\hat{S}}^{\top}(x_m)_{\hat{S}} \right\|_2^2 = \|(I - P_{\hat{S}})x_p\|_2^2 + \left\| W_{m,\hat{S}}^{\top}\left(x_{\mathrm{LS}}^{\hat{S}} - (x_m)_{\hat{S}}\right) \right\|_2^2.$$

This completes the proof. □

Near-correct subspace error. Suppose the support is correct, $\hat{S} = S$, and the learned decoder is close to the ground-truth dictionary on S:

$$W_{m,S}^{\top} = W_{p,S} + \Delta_S, \qquad \|\Delta_S\|_{\mathrm{op}} \le \delta.$$

Assume further that

$$x_p = W_{p,S}x_S + \xi.$$

Then

$$\begin{aligned} (I - P_S)x_p &= (I - P_S)(W_{p,S}x_S + \xi) \\ &= (I - P_S)\big((W_{m,S}^{\top} - \Delta_S)x_S + \xi\big) \\ &= -(I - P_S)\Delta_S x_S + (I - P_S)\xi, \end{aligned}$$

because $W_{m,S}^{\top}x_S \in \mathrm{span}(W_{m,S}^{\top})$, and hence

$$(I - P_S)W_{m,S}^{\top}x_S = 0.$$

Since P_S is an orthogonal projection, $\|I - P_S\|_{\mathrm{op}} \le 1$. Therefore,

$$\|(I - P_S)x_p\|_2 \le \|\Delta_S x_S\|_2 + \|\xi\|_2 \le \delta\|x_S\|_2 + \|\xi\|_2.$$

B.5 Proof of the self-consistency LS-gap identity

Proof of Proposition 3. Under the stable-support and value-preserving conditions, both the first encoder pass and the decode-encode pass are linear on the selected support $\hat{S}$. Hence

$$(x_m)_{\hat{S}} = W_{m,\hat{S}}x_p + (b_m)_{\hat{S}}, \qquad (x_t)_{\hat{S}} = W_{m,\hat{S}}\hat{x}_p + (b_m)_{\hat{S}}.$$

Since $\hat{S} = \mathrm{supp}(x_m)$, the reconstruction is

$$\hat{x}_p = W_m^{\top}x_m = W_{m,\hat{S}}^{\top}(x_m)_{\hat{S}}.$$

Therefore,

$$\begin{aligned}(x_t - x_m)_{\hat{S}} &= W_{m,\hat{S}}\hat{x}_p - W_{m,\hat{S}}x_p \\ &= W_{m,\hat{S}}W_{m,\hat{S}}^{\top}(x_m)_{\hat{S}} - W_{m,\hat{S}}x_p \\ &= G_{m,\hat{S}}(x_m)_{\hat{S}} - W_{m,\hat{S}}x_p.\end{aligned}$$

By definition,

$$x_{\mathrm{LS}}^{\hat{S}} = G_{m,\hat{S}}^{-1}W_{m,\hat{S}}x_p,$$

and hence

$$W_{m,\hat{S}}x_p = G_{m,\hat{S}}x_{\mathrm{LS}}^{\hat{S}}.$$

Substituting this identity gives

$$(x_t - x_m)_{\hat{S}} = G_{m,\hat{S}}\left((x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}}\right),$$

which proves equation 18.

It remains to prove the two-sided estimate. Suppose

$$\alpha I \preceq G_{m,\hat{S}} \preceq \beta I.$$

Let

$$v = (x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}}.$$

Since $G_{m,\hat{S}}$ is symmetric positive definite, the spectral bound gives

$$\alpha\|v\|_2 \le \|G_{m,\hat{S}}v\|_2 \le \beta\|v\|_2.$$

Using $G_{m,\hat{S}}v = (x_t - x_m)_{\hat{S}}$, we obtain

$$\alpha\left\|(x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}}\right\|_2 \le \left\|(x_t - x_m)_{\hat{S}}\right\|_2 \le \beta\left\|(x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}}\right\|_2.$$

Rearranging yields

$$\frac{1}{\beta}\left\|(x_t - x_m)_{\hat{S}}\right\|_2 \le \left\|(x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}}\right\|_2 \le \frac{1}{\alpha}\left\|(x_t - x_m)_{\hat{S}}\right\|_2,$$

which proves equation 19. □

B.6 Proof of the local latent recovery bound

Proof of Proposition 4. By Proposition 3 with $\hat{S} = S$, we have

$$(x_t - x_m)_S = G_{m,S}\left((x_m)_S - x_{\mathrm{LS}}^{S}\right).$$

Hence

$$(x_m)_S - x_{\mathrm{LS}}^{S} = G_{m,S}^{-1}(x_t - x_m)_S.$$

Therefore,

$$(x_m)_S - x_S = G_{m,S}^{-1}(x_t - x_m)_S + \left(x_{\mathrm{LS}}^{S} - x_S\right).$$

It remains to express $x_{\mathrm{LS}}^{S} - x_S$. By definition,

$$x_{\mathrm{LS}}^{S} = G_{m,S}^{-1}W_{m,S}x_p.$$

Using

$$W_{m,S}^{\top} = W_{p,S} + \Delta_S,$$

we have

$$W_{p,S} = W_{m,S}^{\top} - \Delta_S.$$

Since

$$x_p = W_{p,S}x_S + \xi,$$

we can write

$$x_p = (W_{m,S}^{\top} - \Delta_S)x_S + \xi = W_{m,S}^{\top}x_S + \xi - \Delta_S x_S.$$

Therefore,

$$\begin{aligned} x_{\mathrm{LS}}^{S} - x_S &= G_{m,S}^{-1} W_{m,S} x_p - x_S \\ &= G_{m,S}^{-1} W_{m,S} \left(W_{m,S}^{\top} x_S + \xi - \Delta_S x_S \right) - x_S \\ &= G_{m,S}^{-1} G_{m,S} x_S + G_{m,S}^{-1} W_{m,S} (\xi - \Delta_S x_S) - x_S \\ &= G_{m,S}^{-1} W_{m,S} (\xi - \Delta_S x_S). \end{aligned}$$

Combining the two decompositions gives

$$(x_m)_S - x_S = G_{m,S}^{-1} (x_t - x_m)_S + G_{m,S}^{-1} W_{m,S} (\xi - \Delta_S x_S),$$

which proves equation 20.

Taking norms yields

$$\begin{aligned} \|(x_m)_S - x_S\|_2 \le \|G_{m,S}^{-1}\|_{\mathrm{op}} \|(x_t - x_m)_S\|_2 \\ + \|G_{m,S}^{-1}\|_{\mathrm{op}} \|W_{m,S}\|_{\mathrm{op}} \left(\|\xi\|_2 + \|\Delta_S x_S\|_2 \right). \end{aligned}$$

Since

$$\lambda_{\min}(G_{m,S}) \ge \alpha,$$

we have

$$\|G_{m,S}^{-1}\|_{\mathrm{op}} \le \frac{1}{\alpha}.$$

Since

$$\lambda_{\max}(G_{m,S}) \le \beta$$

and $G_{m,S} = W_{m,S} W_{m,S}^{\top}$, we have

$$\|W_{m,S}\|_{\mathrm{op}} \le \sqrt{\beta}.$$

Finally, $\|\Delta_S x_S\|_2 \le \delta \|x_S\|_2$. Therefore,

$$\|(x_m)_S - x_S\|_2 \le \frac{1}{\alpha} \|(x_t - x_m)_S\|_2 + \frac{\sqrt{\beta}}{\alpha} \big(\|\xi\|_2 + \delta \|x_S\|_2 \big),$$

which proves equation 21. □

C Extensions: untied weights and TopK support control

C.1 Why masking is needed for signed hard TopK self-consistency

We justify the score-level masked self-consistency loss used for the signed hard TopK encoder in equation 26. The key point is that the TopK operator makes the support a discrete ranking decision. As a result, the unmasked post-TopK self-consistency loss $\|x_t - x_m\|_2^2$ does not only measure coefficient inconsistency; it also penalizes changes in the selected support. In contrast, the score-level masked loss isolates the coefficient-level residual on the first-pass selected support.

Let

$$a = W_m x_p + b_m, \qquad x_m = \mathrm{TopK}_{|\cdot|,K}(a), \qquad \hat{S} = \mathrm{supp}(x_m),$$

where $\mathrm{TopK}_{|\cdot|,K}$ keeps the K largest-magnitude entries of a and preserves their signs. Hence, on the selected support, $(x_m)_{\hat{S}} = a_{\hat{S}}$, while $(x_m)_{\hat{S}^c} = 0$. The reconstruction is

$$\hat{x}_p = W_m^{\top} x_m = W_{m,\hat{S}}^{\top} (x_m)_{\hat{S}}.$$

After decoding and re-encoding, let

$$a_t = W_m \hat{x}_p + b_m, \qquad x_t = \mathrm{TopK}_{|\cdot|,K}(a_t), \qquad T = \mathrm{supp}(x_t).$$

Define

$$G_{m,\hat{S}} := W_{m,\hat{S}} W_{m,\hat{S}}^{\top},$$

and, when $G_{m,\hat{S}}$ is invertible, define the selected-support least-squares coefficient

$$x_{\mathrm{LS}}^{\hat{S}} := G_{m,\hat{S}}^{-1} W_{m,\hat{S}} x_p. \tag{31}$$

Lemma 7 (Direct versus masked signed TopK self-consistency)**.** *Assume that $G_{m,\hat{S}}$ is invertible. Then the score-level masked residual satisfies the exact identity*

$$(a_t - a)_{\hat{S}} = G_{m,\hat{S}} \left((x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}} \right). \tag{32}$$

Thus the pointwise masked score-level residual

$$\ell_{\mathrm{sc}}^{\hat{S}} := \|(a_t - a)_{\hat{S}}\|_2^2$$

can be written as

$$\ell_{\mathrm{sc}}^{\hat{S}} = \left\| G_{m,\hat{S}} \left((x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}} \right) \right\|_2^2. \tag{33}$$

This identity does not require $T = \hat{S}$.

If $T = \hat{S}$, the post-TopK direct residual $\|x_t - x_m\|_2^2$ equals $\|(a_t - a)_{\hat{S}}\|_2^2$. For general T, it decomposes into coefficient inconsistency on retained coordinates and penalties for dropped and newly selected coordinates. Specifically, if

$$C = \hat{S} \cap T, \qquad L = \hat{S} \setminus T, \qquad N = T \setminus \hat{S},$$

then

$$\|x_t - x_m\|_2^2 = \underbrace{\|(a_t - a)_C\|_2^2}_{\text{coefficient inconsistency on stable support}} + \underbrace{\|a_L\|_2^2}_{\text{dropped-support penalty}} + \underbrace{\|(a_t)_N\|_2^2}_{\text{new-support penalty}}. \tag{34}$$

Proof. We first prove the masked identity. Since x_m is supported on $\hat{S}$, the reconstruction is

$$\hat{x}_p = W_{m,\hat{S}}^\top (x_m)_{\hat{S}}.$$

Therefore,

$$(a_t)_{\hat{S}} = W_{m,\hat{S}} \hat{x}_p + (b_m)_{\hat{S}} = W_{m,\hat{S}} W_{m,\hat{S}}^\top (x_m)_{\hat{S}} + (b_m)_{\hat{S}}.$$

On the other hand,

$$a_{\hat{S}} = W_{m,\hat{S}} x_p + (b_m)_{\hat{S}}.$$

Subtracting gives

$$(a_t - a)_{\hat{S}} = W_{m,\hat{S}} W_{m,\hat{S}}^\top (x_m)_{\hat{S}} - W_{m,\hat{S}} x_p.$$

Using $G_{m,\hat{S}} = W_{m,\hat{S}} W_{m,\hat{S}}^\top$, we get

$$(a_t - a)_{\hat{S}} = G_{m,\hat{S}} (x_m)_{\hat{S}} - W_{m,\hat{S}} x_p.$$

By the definition of $x_{\mathrm{LS}}^{\hat{S}}$,

$$W_{m,\hat{S}} x_p = G_{m,\hat{S}} x_{\mathrm{LS}}^{\hat{S}}.$$

Therefore,

$$(a_t - a)_{\hat{S}} = G_{m,\hat{S}} \left((x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}} \right),$$

which proves equation 32. Equation equation 33 follows by taking squared norms.

We now prove the direct post-TopK decomposition. Since

$$x_m = \mathrm{TopK}_{|\cdot|,K}(a),$$

we have

$$(x_m)_{\hat{S}} = a_{\hat{S}}, \qquad (x_m)_{\hat{S}^c} = 0.$$

Similarly, since

$$x_t = \mathrm{TopK}_{|\cdot|,K}(a_t),$$

we have

$$(x_t)_T = (a_t)_T, \qquad (x_t)_{T^c} = 0.$$

The coordinate set decomposes into the disjoint union of

$$C = \hat{S} \cap T, \qquad L = \hat{S} \setminus T, \qquad N = T \setminus \hat{S},$$

and the remaining coordinates, which are inactive in both x_m and x_t. On C, both codes are active, so

$$(x_t - x_m)_C = (a_t - a)_C.$$

On L, the first code is active and the second is inactive, so

$$(x_t - x_m)_L = 0 - a_L = -a_L.$$

On N, the first code is inactive and the second is active, so

$$(x_t - x_m)_N = (a_t)_N - 0 = (a_t)_N.$$

All other coordinates contribute zero. Summing the squared norms over these disjoint sets gives equation 34. □

Consequence. If

$$\alpha I \preceq G_{m,\hat{S}} \preceq \beta I,$$

then equation 32 gives

$$\frac{1}{\beta}\|(a_t - a)_{\hat{S}}\|_2 \le \left\|(x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}}\right\|_2 \le \frac{1}{\alpha}\|(a_t - a)_{\hat{S}}\|_2. \tag{35}$$

Therefore, the pointwise score-level masked residual directly controls the selected-support least-squares coefficient gap. After averaging over samples and applying the normalization in equation 26, this gives the masked loss used in the algorithm. Unlike the unmasked post-TopK loss, this conclusion does not require the decode-encode TopK support T to equal the first-pass support $\hat{S}$.

The decomposition equation 34 explains why masking is useful for signed hard TopK encoders. The unmasked post-TopK loss contains support-switching terms: $\|a_L\|_2^2$ penalizes coordinates dropped by the second TopK pass, and $\|(a_t)_N\|_2^2$ penalizes newly selected coordinates. These terms encourage the second TopK support to match the first, and therefore mix coefficient refinement with support-switching penalties. The score-level masked loss removes the new-support penalty and replaces the dropped-support hard-zero penalty with a score-level discrepancy on the original support, thereby preserving the LS-consistency interpretation on the first-pass selected support.

C.2 Extension to untied-weight SAEs

Our main analysis focuses on tied-weight SAEs, where the encoder and decoder share weights. This setting makes the projection-induced latent bias particularly transparent: the encoder returns projection coefficients rather than inverse-corrected coefficients. We now show that the same coefficient-level viewpoint extends to untied-weight SAEs.

Consider an untied SAE

$$x_m = \sigma(Ex_p + b_m), \qquad \hat{x}_p = Dx_m, \qquad x_t = \sigma(E\hat{x}_p + b_m), \tag{36}$$

where $E \in \mathbb{R}^{d_m \times d_p}$ is the encoder and $D \in \mathbb{R}^{d_p \times d_m}$ is the decoder. For an index set $U \subseteq [d_m]$, let $E_U \in \mathbb{R}^{|U| \times d_p}$ denote the rows of E indexed by U, and let $D_U \in \mathbb{R}^{d_p \times |U|}$ denote the columns of D indexed by U. Define the active encoder-decoder cross-Gram matrix

$$H_U := E_U D_U. \tag{37}$$

In the tied-weight case, $E = W_m$ and $D = W_m^\top$, so

$$H_U = W_{m,U} W_{m,U}^\top = G_{m,U}.$$

Thus, the active Gram matrix in the tied analysis is replaced by the encoder-decoder cross-Gram H_U in the untied setting.

C.2.1 Latent recovery requires active inverse consistency

We first show that, in untied SAEs, dictionary and support recovery reduce latent-code recovery to an active inverse-consistency condition. The relevant condition is that the encoder acts as a local inverse of the decoder on the active support.

Lemma 8 (Untied latent recovery requires active inverse consistency)**.** *Consider the untied SAE in equation 36. Suppose*

$$x_p = W_{p,S}x_S, \qquad S = \mathrm{supp}(x),$$

where x_S denotes the active coefficients, and assume that the decoder dictionary is exactly recovered on S:

$$D_S = W_{p,S}.$$

Assume also that the inferred support is correct,

$$\mathrm{supp}(x_m) = S,$$

and that the activation is value-preserving on S. Then

$$(x_m)_S = E_S W_{p,S} x_S + (b_m)_S. \tag{38}$$

Consequently,

$$(x_m)_S - x_S = (E_S W_{p,S} - I)x_S + (b_m)_S. \tag{39}$$

In particular, exact latent-code recovery for all x_S in a full-dimensional set requires

$$E_S W_{p,S} = I \qquad \textit{and} \qquad (b_m)_S = 0.$$

Conversely, if $E_S W_{p,S} = I$ and $(b_m)_S = 0$, then $(x_m)_S = x_S$.

Proof. Since the activation is value-preserving on S and the inferred support is S, we have

$$(x_m)_S = E_S x_p + (b_m)_S.$$

Using $x_p = W_{p,S}x_S$, this gives

$$(x_m)_S = E_S W_{p,S} x_S + (b_m)_S,$$

which proves equation 38 and equation 39. If $(x_m)_S = x_S$ for all x_S in a full-dimensional set, then

$$(E_S W_{p,S} - I)x_S + (b_m)_S = 0$$

for all such x_S, which implies $E_S W_{p,S} = I$ and $(b_m)_S = 0$. The converse is immediate. □

Remark 9 (Tied orthogonality versus untied inverse consistency). In the tied case, $E_S = W_{p,S}^\top$, so the untied inverse-consistency condition

$$E_S W_{p,S} = I$$

reduces to

$$W_{p,S}^\top W_{p,S} = I.$$

Thus, tied latent recovery requires the active dictionary to be self-inverting under the tied projection map, while untied latent recovery only requires active encoder-decoder inverse consistency. Untied encoders can therefore remove the tied projection bias by learning a local inverse of the decoder.

The same obstruction has a population form. Let

$$H_S^\star := E_S W_{p,S}.$$

Then equation 39 can be written as

$$(x_m)_S - x_S = (H_S^\star - I)x_S + (b_m)_S.$$

If x_S has mean m_S and covariance Σ_S, then optimizing over a fixed on-support bias gives

$$\inf_{\bar{b}_S} \mathbb{E}\|(x_m)_S - x_S\|_2^2 = \mathrm{Tr}\big((H_S^\star - I)\Sigma_S(H_S^\star - I)^\top\big). \tag{40}$$

The optimal bias is

$$\bar{b}_S^\star = -(H_S^\star - I)m_S.$$

Thus, a fixed bias can cancel the mean encoder-dictionary distortion, but cannot generally remove sample-dependent cross-Gram distortion. If $\Sigma_S \succ 0$, then the right-hand side of equation 40 is zero if and only if $H_S^\star = I$.

C.2.2 SELF-CONSISTENCY CONTROLS AN ENCODER-INDUCED INVERSE-CODE GAP

We next show that, in the untied setting, self-consistency controls the gap between the encoder output and an encoder-induced inverse code. This is the untied analogue of Proposition 3. The key difference is that the ordinary decoder least-squares code is replaced by a code induced jointly by the encoder and decoder.

Lemma 10 (Untied self-consistency identity)**.** *Let $\hat{S} = \mathrm{supp}(x_m)$. Assume that*

$$\mathrm{supp}(x_t) = \hat{S}$$

and that the activation is value-preserving on $\hat{S}$ for both encoder passes. Define

$$H_{\hat{S}} := E_{\hat{S}} D_{\hat{S}}.$$

Assume $H_{\hat{S}}$ is invertible, and define the encoder-induced inverse code

$$x_{\mathrm{inv}}^{\hat{S}} := H_{\hat{S}}^{-1} E_{\hat{S}} x_p. \tag{41}$$

Then

$$(x_t - x_m)_{\hat{S}} = H_{\hat{S}} \left((x_m)_{\hat{S}} - x_{\mathrm{inv}}^{\hat{S}} \right). \tag{42}$$

Consequently, if $\sigma_{\min}(H_{\hat{S}}) \geq \alpha > 0$, then

$$\|(x_m)_{\hat{S}} - x_{\mathrm{inv}}^{\hat{S}}\|_2 \leq \frac{1}{\alpha} \|(x_t - x_m)_{\hat{S}}\|_2. \tag{43}$$

Proof. By value preservation on the first encoder pass,

$$(x_m)_{\hat{S}} = E_{\hat{S}} x_p + (b_m)_{\hat{S}}.$$

By value preservation on the decode-encode pass,

$$(x_t)_{\hat{S}} = E_{\hat{S}} D_{\hat{S}} (x_m)_{\hat{S}} + (b_m)_{\hat{S}}.$$

Subtracting the two identities gives

$$(x_t - x_m)_{\hat{S}} = E_{\hat{S}} D_{\hat{S}} (x_m)_{\hat{S}} - E_{\hat{S}} x_p = H_{\hat{S}} (x_m)_{\hat{S}} - E_{\hat{S}} x_p.$$

Since $E_{\hat{S}} x_p = H_{\hat{S}} x_{\mathrm{inv}}^{\hat{S}}$, we obtain

$$(x_t - x_m)_{\hat{S}} = H_{\hat{S}} \left((x_m)_{\hat{S}} - x_{\mathrm{inv}}^{\hat{S}} \right).$$

The norm bound follows from $\|H_{\hat{S}}^{-1}\|_{\mathrm{op}} \leq \alpha^{-1}$. □

Remark 11 (Relation to the tied case). When $E = D^\top$, we have

$$H_{\hat{S}} = D_{\hat{S}}^\top D_{\hat{S}}.$$

Moreover,

$$x_{\mathrm{inv}}^{\hat{S}} = (D_{\hat{S}}^\top D_{\hat{S}})^{-1} D_{\hat{S}}^\top x_p = x_{\mathrm{LS}}^{\hat{S}}.$$

Thus, equation 42 reduces exactly to the tied-weight identity

$$(x_t - x_m)_{\hat{S}} = G_{m,\hat{S}} \left((x_m)_{\hat{S}} - x_{\mathrm{LS}}^{\hat{S}} \right).$$

Remark 12 (When self-consistency implies $H_U \approx I$). Fix the model parameters and a stable support U, and let $Z = (x_m)_U$ have mean m_U and covariance Σ_U. With $s_U = (H_U - I)Z + b_U$, we have

$$\mathbb{E}\|s_U\|_2^2 = \mathrm{Tr}\big((H_U - I)\Sigma_U (H_U - I)^\top\big) + \|(H_U - I)m_U + b_U\|_2^2.$$

If $\Sigma_U \succeq \nu I$ for $\nu > 0$, it follows that

$$\|H_U - I\|_F \leq \nu^{-1/2} \big(\mathbb{E}\|s_U\|_2^2\big)^{1/2}.$$

For a fixed distribution of Z, minimizing over a free intercept removes the squared-mean term. The covariance lower bound ensures variation along every active direction, allowing the population self-consistency residual to control the full active cross-map.

C.2.3 LATENT RECOVERY BOUNDS FROM RECONSTRUCTION, SELF-CONSISTENCY, AND INVERSE CONSISTENCY

We now compare three ways of controlling latent-code recovery in the near-correct untied regime. The comparison is local: we assume the inferred and decode-encode supports are correct and stable, and we quantify the remaining latent amplitude error.

Let

$$x_p = W_{p,S} x_S + \xi, \qquad \operatorname{supp}(x_m) = \operatorname{supp}(x_t) = S,$$

and assume that the activation is value-preserving on S for both encoder passes. Suppose the untied decoder satisfies

$$D_S = W_{p,S} + \Delta_S, \qquad \|\Delta_S\|_{\mathrm{op}} \le \delta.$$

Define the decoder mismatch

$$q := x_p - D_S x_S = \xi - \Delta_S x_S, \tag{44}$$

and the latent error

$$e := (x_m)_S - x_S.$$

Let

$$H_S := E_S D_S.$$

Proposition 13 (Latent recovery bounds in the near-correct untied regime)**.** *Under the conditions above, define the reconstruction residual*

$$r := x_p - D_S (x_m)_S$$

and the self-consistency residual

$$s := (x_t - x_m)_S.$$

Then the following exact identities hold:

$$r = q - D_S e, \tag{45}$$

and

$$s = H_S e - E_S q. \tag{46}$$

Consequently, if $\sigma_{\min}(D_S) \ge \sigma_D > 0$, then

$$\|e\|_2 \le \frac{1}{\sigma_D} \left(\|r\|_2 + \|q\|_2 \right). \tag{47}$$

If $\sigma_{\min}(H_S) \ge \sigma_H > 0$, then

$$\|e\|_2 \le \frac{1}{\sigma_H} \left(\|s\|_2 + \|E_S\|_{\mathrm{op}} \|q\|_2 \right). \tag{48}$$

Moreover, define the active inverse-consistency regularizer

$$\mathcal{R}_{\mathrm{inv}}(S) := \|H_S - I\|_F^2.$$

Then

$$\|e\|_2 \le \sqrt{\mathcal{R}_{\mathrm{inv}}(S)}\, \|x_S\|_2 + \|E_S\|_{\mathrm{op}} \|q\|_2 + \|(b_m)_S\|_2. \tag{49}$$

Proof. Since $x_p = D_S x_S + q$, we have

$$r = x_p - D_S (x_m)_S = q - D_S\big((x_m)_S - x_S\big) = q - D_S e,$$

which proves equation 45. Thus $D_S e = q - r$, and if $\sigma_{\min}(D_S) \ge \sigma_D$,

$$\|e\|_2 \le \sigma_D^{-1} (\|q\|_2 + \|r\|_2),$$

which proves equation 47.

For the self-consistency residual, value preservation on S gives

$$(x_m)_S = E_S x_p + (b_m)_S, \qquad (x_t)_S = E_S D_S (x_m)_S + (b_m)_S.$$

Hence

$$s = (x_t - x_m)_S = E_S D_S (x_m)_S - E_S x_p.$$

Using $x_p = D_S x_S + q$, we obtain

$$s = H_S (x_m)_S - H_S x_S - E_S q = H_S e - E_S q,$$

which proves equation 46. Therefore $H_S e = s + E_S q$, and if $\sigma_{\min}(H_S) \geq \sigma_H$,

$$\|e\|_2 \leq \sigma_H^{-1} (\|s\|_2 + \|E_S\|_{\mathrm{op}} \|q\|_2),$$

which proves equation 48.

Finally,

$$(x_m)_S = E_S x_p + (b_m)_S = E_S (D_S x_S + q) + (b_m)_S = H_S x_S + E_S q + (b_m)_S.$$

Thus

$$e = (H_S - I) x_S + E_S q + (b_m)_S.$$

Since

$$\|H_S - I\|_{\mathrm{op}} \leq \|H_S - I\|_F = \sqrt{\mathcal{R}_{\mathrm{inv}}(S)},$$

we obtain equation 49. □

Interpretation. Proposition 13 shows that reconstruction loss, self-consistency, and inverse-consistency regularization control the same latent error through different operators. Reconstruction loss controls $D_S e$, and therefore its latent recovery guarantee depends on the conditioning of the active decoder D_S. Self-consistency controls $H_S e$, where $H_S = E_S D_S$, and therefore its guarantee depends on the conditioning of the latent self-map. Inverse-consistency regularization directly controls $H_S - I$.

Conditioning and noise amplification. The reconstruction and SC bounds in Proposition 13 read

$$\|e\|_2 \leq \frac{\|r\|_2 + \|q\|_2}{\sigma_{\min}(D_S)}, \qquad \|e\|_2 \leq \frac{\|s\|_2 + \|E_S\|_{\mathrm{op}} \|q\|_2}{\sigma_{\min}(H_S)}.$$

The second bound depends on encoder amplification as well as cross-map conditioning. In particular, $H_S \approx I$ need not imply robustness to observation noise or decoder mismatch. For a right singular vector v of D_S with its smallest singular value,

$$\sigma_{\min}(H_S) \leq \|H_S v\|_2 \leq \|E_S\|_{\mathrm{op}} \sigma_{\min}(D_S), \quad \text{so} \quad \frac{\|E_S\|_{\mathrm{op}}}{\sigma_{\min}(H_S)} \geq \frac{1}{\sigma_{\min}(D_S)}.$$

The perturbation term is governed jointly by encoder amplification and active cross-map conditioning, as quantified by $\|E_S\|_{\mathrm{op}} / \sigma_{\min}(H_S)$.

Inverse consistency as a distinct matrix condition. If $\|H_S - I\|_{\mathrm{op}} \leq \varepsilon < 1$, then

$$1 - \varepsilon \leq \sigma_{\min}(H_S) \leq \sigma_{\max}(H_S) \leq 1 + \varepsilon, \qquad \kappa(H_S) \leq \frac{1+\varepsilon}{1-\varepsilon}.$$

The matrix penalty $\|H_S - I\|_F^2$ explicitly targets this condition. Sample-wise SC provides such control only under the covariance condition above.

The role of the auxiliary objective. On a stable value-preserving support, $s = -E_S r = H_S e - E_S q$. Self-consistency therefore weights the reconstruction residual through the current encoder and supplies a coefficient-level training signal under the local conditions above. The controlled untied and fixed-decoder experiments in Appendix F.1 evaluate the resulting coefficient refinement.

The role of the auxiliary objective. On a stable value-preserving support, $s = -E_S r = H_S e - E_S q$. SC therefore weights the observation residual through the current encoder. Together with reconstruction it changes the training objective while preserving the inference architecture. Its connection to coefficient recovery is strongest when support, dictionary mismatch, cross-map conditioning, and encoder amplification are controlled. These are conditions for interpreting the residual; the pointwise bounds do not establish monotonic error reduction during joint optimization. The controlled untied and fixed-decoder experiments in Appendix F.1 provide empirical tests of this refinement mechanism.

D Local Assumptions and the Meaning of Coefficient Recovery

D.1 Local conditions for support recovery

Under exact dictionary recovery for a tied SAE in the noiseless model, write

$$a(x) = W_p^\top W_p x + b_m, \qquad S = \operatorname{supp}(x), \qquad 0 < k = |S| < d_m.$$

For SignedTopK retaining the k largest-magnitude pre-activations, a sufficient condition for recovering S is

$$\min_{i \in S} |a_i(x)| \geq \max_{j \notin S} |a_j(x)| + \gamma, \qquad \gamma > 0. \tag{50}$$

For ReLU, a sufficient condition is

$$\min_{i \in S} a_i(x) \geq \gamma, \qquad \max_{j \notin S} a_j(x) \leq -\gamma. \tag{51}$$

Both activations preserve the pre-activation values on the recovered support. These values need not equal the true coefficients.

For unit-norm dictionary atoms with coherence μ, let $x_{\min} = \min_{i \in S} |x_i|$. Applying the triangle inequality to active and inactive pre-activations gives the following sufficient condition for equation 50:

$$(1 + \mu) x_{\min} - 2\mu \|x_S\|_1 \geq \|(b_m)_S\|_\infty + \|(b_m)_{S^c}\|_\infty + \gamma. \tag{52}$$

For equal-magnitude active coefficients of magnitude c, the left-hand side is $c(1 - (2k-1)\mu)$. Thus, if $\mu < 1/(2k-1)$ and the coefficient scale is unrestricted, sufficiently large c satisfies this condition for any fixed bias. Appendix D.2 gives a separate overcomplete ReLU construction with strict support margins.

A pre-activation perturbation of infinity norm below $\gamma/2$ preserves the SignedTopK selection, since the magnitude gap between active and inactive coordinates remains positive. For ReLU, a perturbation of infinity norm below γ preserves all pre-activation signs. These margins ensure local stability of the first-pass support. The unmasked self-consistency results additionally assume that the second encoder pass preserves the first-pass support. The masked formulation in Appendix C.1 allows the second-pass support to change.

D.2 An overcomplete example of activation-ranking reversal

After dictionary correspondence and column normalization are fixed, coefficient errors can still depend on the other active features. The following example shows that this interference can reverse activation rankings across samples in a way that fixed positive feature-wise rescaling cannot correct.

Let $d_p = 2$, $d_m = 3$, and define

$$d_1 = (1, 0)^\top, \qquad d_2 = (1/5, \sqrt{24}/5)^\top, \qquad d_3 = -\frac{d_1 + d_2}{\sqrt{12/5}}.$$

All three atoms have unit norm. Set $W_p = (d_1, d_2, d_3)$, take the exact tied decoder $W_m^\top = W_p$, and use zero encoder bias:

$$x_p = W_p x, \qquad x_m = \mathrm{ReLU}(W_p^\top x_p).$$

For $x = (x_1, x_2, 0)^\top$ with $x_1, x_2 > 0$, the active support is $S = \{1, 2\}$ and

$$G_S = \begin{pmatrix} 1 & 1/5 \\ 1/5 & 1 \end{pmatrix}, \qquad G_S x_S > 0, \qquad d_3^\top x_p = -\sqrt{3/5}\,(x_1 + x_2) < 0.$$

The vector inequality is componentwise, so the ReLU encoder recovers exactly S. The active Gram eigenvalues are $4/5$ and $6/5$, hence the selected decoder is full rank.

Consider the true codes

$$x^{(a)} = (1, 0.1, 0)^\top, \qquad x^{(b)} = (0.9, 1, 0)^\top.$$

Their encoder outputs are

$$x_m^{(a)} = (1.02, 0.30, 0)^\top, \qquad x_m^{(b)} = (1.10, 1.18, 0)^\top.$$

Feature 1 has a larger true amplitude in sample a, since $1 > 0.9$, but its encoder activation is larger in sample b, since $1.02 < 1.10$. For every $c > 0$, multiplying both activations of feature 1 by c preserves this reversed ordering.

The decode-encode pass also preserves S. Indeed, x_m has positive entries exactly on S, so applying the same calculation to $W_p x_m$ again gives positive active pre-activations and a negative third pre-activation. For both samples and both encoder passes, the ReLU margin in equation 51 holds with $\gamma = 0.30$. Thus the example has an exact normalized overcomplete dictionary, correct first-pass support, and stable value-preserving cycle support, while its coefficient distortion changes the activation ranking.

D.3 How self-consistency and reconstruction weight coefficient errors

Under the tied-weight conditions of Proposition 3, let $U = \mathrm{supp}(x_m) = \mathrm{supp}(x_t)$ and write

$$D_U = W_{m,U}^\top, \qquad G = D_U^\top D_U, \qquad e = (x_m)_U - x_{\mathrm{LS}}^U.$$

The orthogonal LS decomposition and self-consistency identity give the pointwise losses

$$\mathcal{L}_{\mathrm{rec}} = \|(I - P_U)x_p\|_2^2 + e^\top G e, \tag{53}$$

$$\mathcal{L}_{\mathrm{sc}} = e^\top G^2 e. \tag{54}$$

Let $\alpha > 0$ and β be the smallest and largest eigenvalues of G, and define $E_{\mathrm{coef}} = e^\top G e$. Then

$$\alpha E_{\mathrm{coef}} \le \mathcal{L}_{\mathrm{sc}} \le \beta E_{\mathrm{coef}}.$$

For $\lambda \ge 0$,

$$\mathcal{L}_{\mathrm{rec}} + \lambda \mathcal{L}_{\mathrm{sc}} = \|(I - P_U)x_p\|_2^2 + e^\top (G + \lambda G^2) e.$$

In a Gram eigendirection with eigenvalue g, adding self-consistency multiplies the coefficient weight by $1 + \lambda g$ relative to reconstruction alone. Directions with larger Gram eigenvalues therefore receive relatively more weight.

For a fixed decoder and support, both coefficient penalties, viewed as unconstrained quadratic forms, have the same unique LS minimizer $e = 0$. The two penalties coincide when $G = I$. For $\lambda \ge 0$, the spectral condition number of their combined weighting matrix satisfies

$$\kappa(G + \lambda G^2) = \frac{\beta(1 + \lambda\beta)}{\alpha(1 + \lambda\alpha)} \ge \frac{\beta}{\alpha} = \kappa(G).$$

This comparison concerns the quadratic form in the coefficient error e, with the decoder and support fixed. During SAE training, a shared encoder couples the codes across samples, while decoder updates can change the Gram matrix and selected supports. The experiments evaluate how this additional weighting affects coefficient recovery during training.

For hard TopK with a nonempty first-pass support U and a full-column-rank selected decoder, define G and e on that support. Using the first- and second-pass pre-activations a and a_t from Appendix C.1, the normalized masked penalty is

$$\mathcal{L}_{\mathrm{sc}}^{\mathrm{mask}} = \frac{1}{|U|}\|(a_t - a)_U\|_2^2 = \frac{1}{|U|} e^\top G^2 e.$$

This identity holds even when the second TopK support changes, so the same weighting comparison applies with λ replaced by $\lambda/|U|$. Appendix C.1 gives the proof; the untied identities and perturbation bounds are developed in Appendix C.2.

E Experimental Settings and Evaluation

This appendix collects the experimental settings, evaluation metrics, and reporting protocols used in Section 6. Appendix F contains the corresponding results.

E.1 Synthetic data and model settings

Data generation. We generate observations from the sparse-mixing model $x_p = Az^\star$, where $A \in \mathbb{R}^{d\times n}$ has unit-norm columns. Here $A = W_p$, and $z^\star$ denotes the ground-truth latent code x used in Section 3. The dictionary is constructed by mixing an i.i.d. Gaussian matrix with a shared rank-one component and normalizing its columns; the parameter ρ controls the strength of the shared component. For each sample, we choose k coordinates uniformly without replacement and draw their nonzero coefficients independently from $[0.5, 1.0]$. The main tied-encoder experiment uses $k \in \{8, 16, 32\}$ and $\rho = 0.05$. The correlation sweep fixes $\lambda_{\text{sc}} = 0.5$ and varies ρ for $k \in \{8, 16\}$.

Tied encoders and baselines. The tied TopK-ReLU SAE has code $\hat{z} = \mathrm{TopK}_k(\mathrm{ReLU}(D^\top x_p))$ and reconstruction $\hat{x}_p = D\hat{z}$. We initialize D near the generating dictionary and learn it during training. The regularized objective adds the unmasked decode-encode penalty

$$\mathcal{L}_{\text{sc}} = \mathbb{E}_{x_p}\left[\|\,\mathrm{Enc}(D\hat{z}) - \hat{z}\|_2^2\right]$$

to the reconstruction loss. We compare the weight sweep with reconstruction-only training and our WSAE reimplementation (Cui et al., 2025), using the same synthetic data generation, optimization budget, and evaluation protocol. The WSAE implementation follows the published algorithmic description; it is a best-effort reimplementation rather than a reproduction from released code.

Untied and fixed-decoder controls. The controlled untied experiments learn the encoder and decoder independently and use $k \in \{8, 16, 32\}$. A separate fixed-decoder control sets $D = A$ throughout training and optimizes only the encoder. We also evaluate an explicitly overcomplete untied setting with observation dimension $d_p = 128$ and latent dimension $d_m = 256$. These controls use ten paired training seeds, with each regularized run compared with its reconstruction-only counterpart.

E.2 Synthetic alignment and recovery metrics

Dictionary matching and scale correction. We use a one-to-one Hungarian assignment that maximizes cosine similarity between learned decoder columns and ground-truth dictionary columns. If $\pi(i)$ is the learned column assigned to ground-truth column a_i, dictionary recovery is measured by the mean matched cosine

$$\mathrm{MCC} = \frac{1}{n}\sum_{i=1}^{n}\frac{\langle d_{\pi(i)}, a_i\rangle}{\|d_{\pi(i)}\|_2\|a_i\|_2}.$$

After matching, positive atom-wise scale correction expresses inferred coefficients in the normalized ground-truth coordinates. In particular, if $d_{\pi(i)} = s_i a_i$ with $s_i > 0$, the corresponding aligned coefficient is $\tilde{z}_i = s_i\hat{z}_{\pi(i)}$. Both matching and scale correction are determined from the decoder columns.

True-code and support recovery. We report the mean latent L_2 error $\|\tilde{z} - z^\star\|_2$ after alignment. Let $S^\star = \mathrm{supp}(z^\star)$ and $\tilde{S} = \mathrm{supp}(\tilde{z})$. Support recovery is measured by

$$\mathrm{F1}(\tilde{S}, S^\star) = \frac{2|\tilde{S}\cap S^\star|}{|\tilde{S}| + |S^\star|},$$

which equals $|\tilde{S}\cap S^\star|/k$ when both supports have size k. We additionally report amplitude MSE on samples with exactly recovered support, $\tilde{S} = S^\star$. This conditional metric measures the error in active amplitudes and is interpreted together with support F1, since the exactly recovered subset can differ between methods.

Selected-support least-squares discrepancy. The coefficient diagnostic uses the learned decoder in its own coordinates. Writing $\hat{S} = \mathrm{supp}(\hat{z})$, define

$$z_{\text{LS}}^{\hat{S}} = (D_{\hat{S}}^\top D_{\hat{S}})^{-1}D_{\hat{S}}^\top x_p, \qquad E_{\text{coef}} = \left\|D_{\hat{S}}(z_{\text{LS}}^{\hat{S}} - \hat{z}_{\hat{S}})\right\|_2^2, \tag{55}$$

for full-column-rank $D_{\hat{S}}$. Thus, E_{coef} is a decoder-weighted discrepancy from the least-squares fit on the selected support. It differs from the aligned true-code error and from conditional amplitude

MSE. The additional diagnostic plots also show the unweighted decoder LS gap $\|\hat{z}_{\hat{S}} - z_{\mathrm{LS}}^{\hat{S}}\|_2$, the decode-encode residual, support stability, and reconstruction error. For the first- and second-pass supports $\hat{S}$ and T, support stability is $|\hat{S} \cap T|/k$.

Untied cross-map diagnostic. For encoder matrix E, decoder D, and active support S in the learned coordinates, we form $H_S = E_S D_S$. The normalized cross-map error is

$$R_H = \frac{\|H_S - I\|_F^2}{|S|}.$$

We report this diagnostic alongside dictionary, support, and coefficient measurements in the untied controls.

E.3 Real-model evaluation protocols

Signed TopK architecture. The real-model experiments use untied signed TopK SAEs with latent width 2048. For scores $a = W_{\mathrm{enc}} x_p + b$, the code $\hat{z} = \Pi_K(a)$ retains the K largest-magnitude entries and preserves their signs. The decoder is linear, $\hat{x}_p = W_{\mathrm{dec}} \hat{z}$. The regularized model adds the score-level masked self-consistency loss on the first-pass support $\hat{S}$:

$$\mathcal{L}_{\mathrm{sc}}^{\hat{S}} = \mathbb{E}_{x_p}\left[|\hat{S}|^{-1} \|(a_t - a)_{\hat{S}}\|_2^2\right], \qquad a_t = W_{\mathrm{enc}} \hat{x}_p + b.$$

TopK fixes the activation budget within each comparison, and no additional ℓ_1 penalty is used.

CE-bench. We train on GPT-2 residual-stream activations from the second-to-last layer, with additional evaluations at layers 2 and 6. The CE-bench experiments use $K = 32$, learning rate 10^{-3}, and 3000 optimization steps. Each step processes 16 sequences of up to 128 tokens, giving at most 2048 tokens per batch before any accumulation. Evaluation uses up to 3000 paired samples. Following CE-bench (Gulko et al., 2025), we report Contrastive, Independence, and the active fraction, with

$$\mathrm{CE}_{\mathrm{total}} = \mathrm{Contrastive} + \mathrm{Independence} - 0.25\,\mathrm{Sparsity}.$$

Contrastive measures view-specific signal, while Independence measures separation of shared and contrasting components. The active fraction is fixed at $32/2048 = 0.015625$, displayed as 0.0156 in the result table.

Dictionary health. We estimate each feature's usage rate from sampled token activations, centered by a running mean. For $m_{t,i} = \mathbf{1}[\hat{z}_{t,i} \neq 0]$, let $u_i = \mathbb{E}_t[m_{t,i}]$ and $p_i = u_i / \sum_j u_j$. The dead-feature fraction and normalized usage entropy are

$$\mathrm{DeadFrac} = \frac{1}{d_{\mathrm{lat}}} \sum_i \mathbf{1}[u_i < 10^{-4}], \qquad \mathrm{UsageEntropy} = -\frac{\sum_i p_i \log(p_i + 10^{-12})}{\log d_{\mathrm{lat}}}.$$

To measure dictionary coherence, normalize decoder columns as $\tilde{w}_i = w_i/(\|w_i\|_2 + 10^{-8})$ and report

$$\mathrm{CoherenceRMS} = \left[\frac{1}{d_{\mathrm{lat}}(d_{\mathrm{lat}} - 1)} \sum_{i \neq j} \langle \tilde{w}_i, \tilde{w}_j \rangle^2\right]^{1/2}.$$

Lower dead fraction and coherence RMS, together with higher usage entropy, indicate broader feature use and less redundant decoder directions.

SAEBench checkpoints and probing. The added SAEBench evaluations (Karvonen et al., 2025) use $K = 8$, context length 128, and GPT-2 activations from `blocks.10.hook_resid_post`. Legacy Sparse Probing evaluates all eight default datasets, with 4000 probe-training and 1000 probe-test examples per dataset. Top-1, Top-2, and Top-5 use one, two, and five SAE features, respectively; Dense uses the full representation. We first macro-average the eight datasets within each seed and then average the changes across paired seeds. SAE-Probes evaluates accuracy, AUC, and F1 over 113 tasks at probe sparsities $k \in \{1, 2, 5\}$. Here, probe sparsity k is distinct from the SAE activation budget $K = 8$.

Automated interpretability and CORE. AutoInterp evaluates 1000 latents per checkpoint with a fixed DeepSeek-V4-Flash judge. We identify these results as a custom-judge AutoInterp evaluation, since the judge is part of the scoring protocol. CORE reports the CE-loss score, KL-divergence score, and explained variance. All three are higher-is-better scores; the first two are not raw losses or divergences.

E.4 Training schedules, weight selection, and reporting

CE-bench schedule. The regularized CE-bench model uses target weight $\lambda = 0.5$. The weight is zero for the first 600 steps, increases linearly to its target from steps 600 to 1200, and remains fixed thereafter. Other hyperparameters match the reconstruction-only baseline. The main CE-bench table reports mean and standard deviation over five matched seeds.

SAEBench weight selection. The SignedTopK weight grid is

$$\lambda \in \{0,\ 3 \times 10^{-6},\ 10^{-5},\ 3 \times 10^{-5},\ 10^{-4},\ 3 \times 10^{-4},\ 10^{-3},\ 3 \times 10^{-3}\}.$$

We select the common weight $\lambda = 10^{-5}$ using Legacy Sparse Probing and keep it fixed for SAE-Probes, AutoInterp, and CORE. Legacy Sparse Probing therefore serves as the weight-selection evaluation. The full Legacy and AutoInterp grids are reported in Appendix F.3. Legacy Sparse Probing, SAE-Probes, and CORE use four paired training seeds, 50–53; AutoInterp uses three paired training seeds. Each regularized checkpoint is compared with the unregularized checkpoint trained with the same seed.

Reporting conventions. Synthetic error reductions and CE-bench changes are reported as relative percentages. Probing, AutoInterp, and CORE changes are reported in percentage points (pp). For scores on a zero-to-one scale, a change of 0.01 corresponds to 1 pp. Reported benchmark changes are averages of paired comparisons. AutoInterp error bars are sample standard deviations of checkpoint scores across seeds, rather than confidence intervals for paired changes. Ranges in the controlled untied results span tested sparsities and are not uncertainty intervals.

F Additional Experimental Results

F.1 Synthetic recovery and coefficient refinement

Dictionary, support, and coefficient recovery. Table 3 separates these measurements in the controlled tied setting. At $k = 8$, both methods already have support F1 equal to 1.000, while latent error, conditional amplitude MSE, and E_{coef} decrease with self-consistency. At $k = 16$ and $k = 32$, support recovery also improves, and the coefficient reductions are larger. Decoder MCC decreases slightly and reconstruction MSE increases by 1.43–4.66% across these comparisons, so the results support coefficient refinement with a modest reconstruction trade-off.

k	MCC (B → SC)	F1 (B → SC)	Latent L_2 reduction (%)	Amplitude MSE reduction (%)	E_{coef} reduction (%)
8	0.9984 → 0.9976	1.000 → 1.000	1.97	3.73	10.24
16	0.9961 → 0.9931	0.997 → 0.999	7.11	9.88	29.99
32	0.9907 → 0.9828	0.990 → 0.997	16.36	18.69	63.83

Table 3: **Controlled tied recovery over ten paired seeds.** B denotes reconstruction-only training and SC denotes self-consistency. The three rightmost columns give relative error reductions. Amplitude MSE is conditioned on exact support recovery.

Untied recovery. In the controlled untied experiments, decoder MCC and support F1 remain approximately 1.000 for both methods. Table 4 reports the ranges across $k \in \{8, 16, 32\}$. The reductions in true-code and amplitude errors accompany a 96.1–99.6% reduction in normalized cross-map error, supporting the active inverse-consistency mechanism when dictionary and support recovery are already accurate.

Metric	Relative reduction
True latent L_2 error	55.2–77.6%
Exact-support amplitude MSE	86.6–97.4%
Selected-support E_{coef}	59.3–88.8%
Normalized cross-map error $\|H_S - I\|_F^2/k$	96.1–99.6%

Table 4: **Controlled untied recovery over ten paired seeds.** Ranges span $k \in \{8, 16, 32\}$ and are not uncertainty intervals.

Fixed ground-truth decoder. When the decoder is fixed to the ground-truth dictionary, both methods have MCC and support F1 equal to one. Across ten paired seeds, self-consistency reduces true latent L_2 error by 69.62% and E_{coef} by 90.70%. This control isolates coefficient improvement with the dictionary fixed and support already correct.

Explicitly overcomplete setting. For the untied setting with $d_p = 128$ and $d_m = 256$, normalized cross-map error decreases from 0.1360 to 0.1225, a reported relative reduction of 9.93%. True latent L_2 error decreases from 0.4405 to 0.4354, a reduction of 1.15%, and E_{coef} decreases by 1.83%. The improvements occur in all ten paired seeds, with dictionary and support recovery unchanged.

Additional diagnostics and correlation sweep. Figure 2 reports the self-consistency residual, unweighted decoder LS gap, decode-encode support stability, and reconstruction error for the tied experiment. Figure 3 fixes $\lambda_{\text{sc}} = 0.5$ and varies dictionary correlation. Across the tested correlation levels, self-consistency reduces latent and coefficient errors while improving or maintaining support recovery.

Alignment check. Across 240 seed-configuration checks with known permutations and positive scalings, the alignment procedure recovered every permutation, with maximum aligned-code error 2.98×10^{-7}.

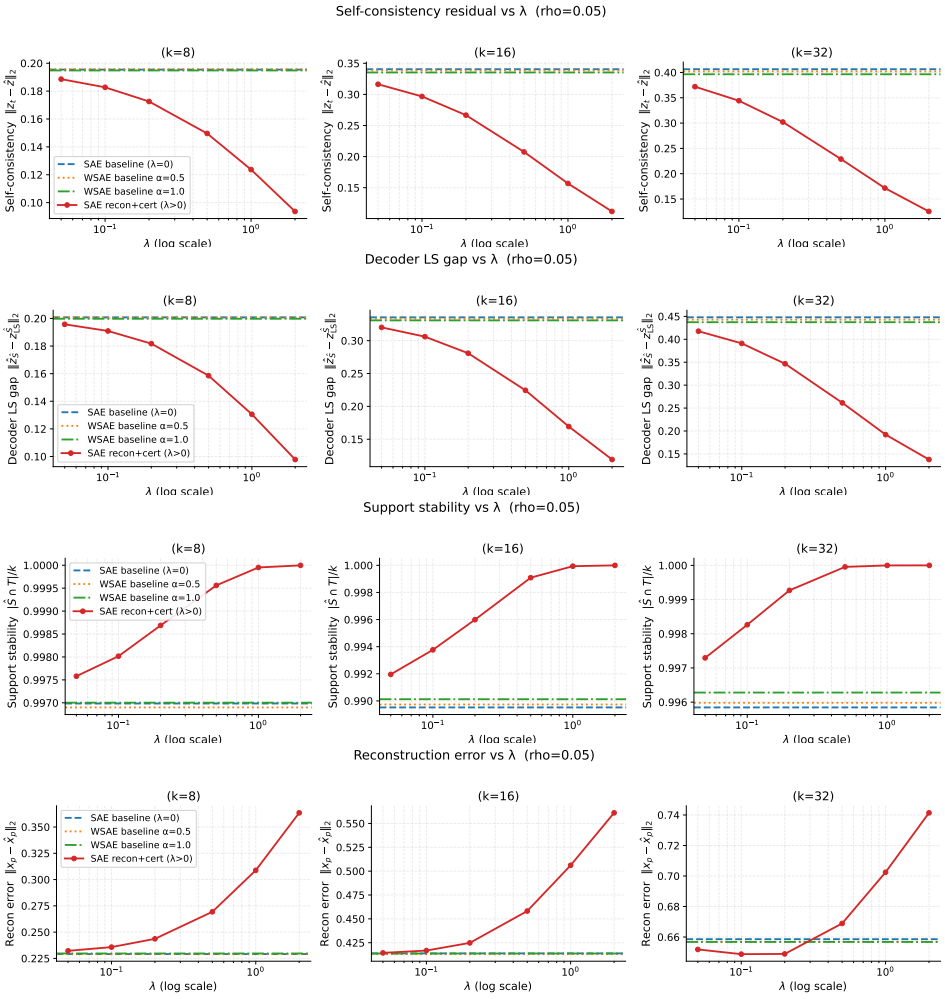

Figure 2: **Additional tied-encoder diagnostics.** Rows show the decode-encode residual, unweighted selected-support decoder LS gap, support stability, and reconstruction error for $k \in \{8, 16, 32\}$ at $\rho = 0.05$. Horizontal lines denote the reconstruction-only and WSAE baselines. The weight sweep complements the latent, coefficient, and support recovery measurements in the main text.

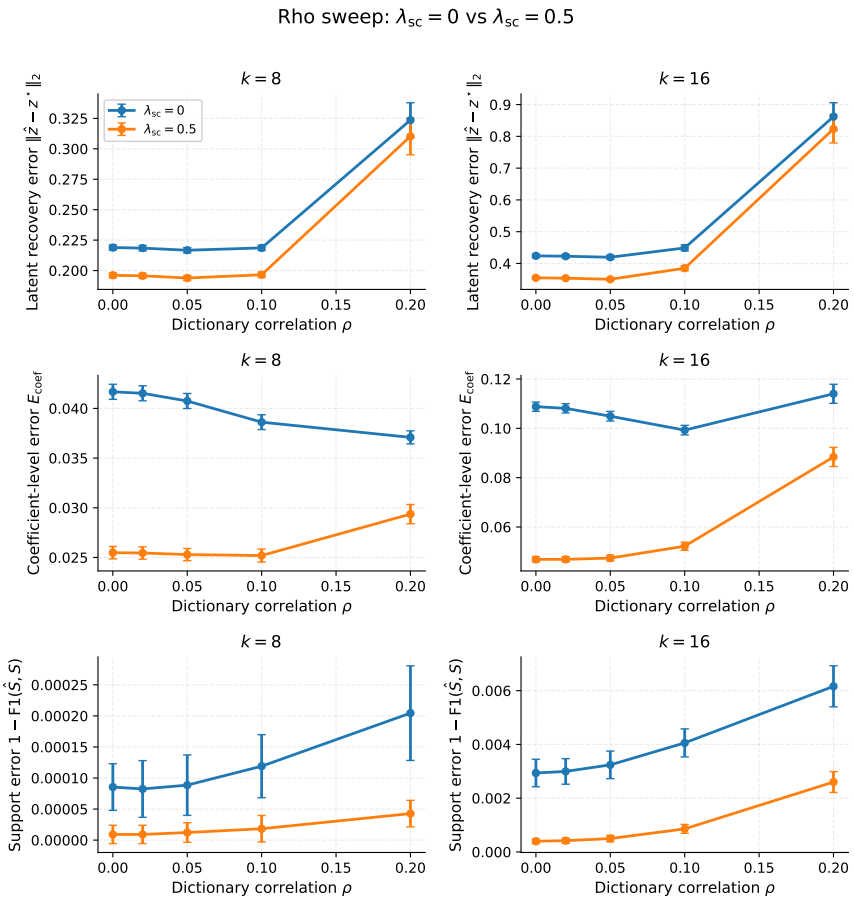

Figure 3: **Robustness to dictionary correlation.** Reconstruction-only training ($\lambda_{\text{sc}} = 0$) is compared with self-consistency ($\lambda_{\text{sc}} = 0.5$) for $k \in \{8, 16\}$. Rows show latent L_2 error, E_{coef}, and support error $1 - \text{F1}$. Larger ρ makes recovery more challenging, especially at higher sparsity.

F.2 CE-BENCH AND DICTIONARY HEALTH

Fixed-sparsity comparison. Table 5 gives the full late-layer CE-bench comparison summarized in the main text. At the same activation budget, Contrastive and Independence increase by 3.54% and 2.59%, respectively, yielding a 3.07% increase in CE total. The dead-feature fraction decreases by 82.14%, usage entropy increases by 4.34%, and coherence RMS decreases by 3.57%. These measurements show more balanced feature usage and less correlated decoder directions alongside the CE-bench improvement.

Metric	No reg	SC ($\lambda = 0.5$)	Relative change
Dead fraction ↓	0.1599 ± 0.0386	0.0286 ± 0.0058	-82.14%
Usage entropy ↑	0.8124 ± 0.0040	0.8477 ± 0.0123	$+4.34\%$
Coherence RMS ↓	0.0387 ± 0.0003	0.0373 ± 0.0001	-3.57%
Contrastive ↑	0.8832 ± 0.0054	0.9145 ± 0.0030	$+3.54\%$
Independence ↑	0.8823 ± 0.0053	0.9051 ± 0.0024	$+2.59\%$
Active fraction (fixed)	0.0156 ± 0.0000	0.0156 ± 0.0000	0.00%
CE total ↑	1.7616 ± 0.0103	1.8157 ± 0.0052	$+3.07\%$

Table 5: **CE-bench with signed TopK SAEs, $K = 32$.** Results are mean $\pm$ standard deviation over five matched seeds, with latent width 2048, $N_{\text{pairs}} = 3000$, and aggregation weight $\alpha = 0.25$.

Results across GPT-2 layers. Table 6 evaluates the same TopK setting at layers 2 and 6. All three CE-bench quantities improve at both layers, with CE-total gains of 1.55% and 1.07%, respectively.

Layer	Metric	No reg	SC	Relative change
2	Contrastive	0.8966	0.9124	$+1.76\%$
	Independence	0.8886	0.9004	$+1.34\%$
	CE total	1.7812	1.8089	$+1.55\%$
6	Contrastive	0.9220	0.9373	$+1.65\%$
	Independence	0.9102	0.9145	$+0.48\%$
	CE total	1.8283	1.8479	$+1.07\%$

Table 6: **CE-bench across GPT-2 layers.** The architecture and fixed-sparsity protocol follow Appendix E.3. All listed scores are higher-is-better.

F.3 SPARSE PROBING AND AUTOMATED INTERPRETABILITY

Legacy Sparse Probing. Table 7 reports the complete weight sweep used for selecting the common operating point. All tested positive weights give positive aggregate mean changes in Top-1, Top-2, Top-5, and Dense probing. The common weight $\lambda = 10^{-5}$ is then held fixed for SAE-Probes, AutoInterp, and CORE.

λ	Top-1	Top-2	Top-5	Dense
3×10^{-6}	+0.125	+0.060	+0.056	+0.011
10^{-5}	+0.504	+0.142	+0.108	+0.047
3×10^{-5}	+0.125	+0.028	+0.003	+0.029
10^{-4}	+0.327	+0.148	+0.046	+0.052
3×10^{-4}	+0.267	+0.177	+0.169	+0.078
10^{-3}	+0.156	+0.037	+0.100	+0.024
3×10^{-3}	+0.207	+0.006	+0.054	+0.081

Table 7: **Legacy Sparse Probing weight sweep.** Entries are mean percentage-point changes against paired unregularized SignedTopK checkpoints, macro-averaged over eight datasets and then over four training seeds.

SAE-Probes across 113 tasks. At $\lambda = 10^{-5}$, all nine aggregate accuracy, AUC, and F1 comparisons have positive mean changes (Table 8). These measurements evaluate task-relevant information accessible through one, two, or five features while the SAE activation budget remains fixed at $K = 8$.

Metric	Probe k	Baseline (%)	SC (%)	Mean change (pp)
Accuracy	1	64.854	64.973	+0.118
	2	66.121	66.222	+0.101
	5	68.381	68.405	+0.024
AUC	1	66.598	66.719	+0.121
	2	69.516	69.616	+0.101
	5	72.776	72.876	+0.101
F1	1	62.530	62.705	+0.175
	2	64.433	64.522	+0.089
	5	67.581	67.820	+0.239

Table 8: **SAE-Probes at the common weight $\lambda = 10^{-5}$.** Results cover 113 tasks and four paired training seeds. Displayed scores and changes are rounded separately.

AutoInterp with a fixed custom judge. Table 9 reports the full custom-judge AutoInterp sweep. All seven nonzero weights have positive mean changes relative to the paired baseline. At the common weight $\lambda = 10^{-5}$, the mean score increases from 0.658333 to 0.663384, or $+0.505$ pp. The largest mean score occurs at $\lambda = 10^{-3}$, while 10^{-5} remains the operating point used for the main cross-evaluation comparison.

λ	Mean $\pm$ sample SD	Mean change (pp)
0	0.658333 ± 0.009906	—
3×10^{-6}	0.663810 ± 0.013408	+0.548
10^{-5}	0.663384 ± 0.020187	+0.505
3×10^{-5}	0.665952 ± 0.019670	+0.762
10^{-4}	0.661703 ± 0.014614	+0.337
3×10^{-4}	0.659851 ± 0.021693	+0.152
10^{-3}	0.672381 ± 0.030155	+1.405
3×10^{-3}	0.666072 ± 0.019553	+0.774

Table 9: **Custom-judge AutoInterp across the weight grid.** Each checkpoint is evaluated on 1000 latents with a fixed DeepSeek-V4-Flash judge; results average three paired training seeds. Standard deviations describe checkpoint-score variability across seeds.

CORE reconstruction and model-behavior scores. At $\lambda = 10^{-5}$, the mean CE-loss score changes from 0.859250 to 0.859553, the KL-divergence score from 0.834427 to 0.834624, and explained variance from 0.825353 to 0.825385, over four paired seeds. These correspond to changes of $+0.0303$, $+0.0197$, and $+0.0032$ pp, respectively. The scores remain near the unregularized baseline while probing and AutoInterp means improve at the same operating point.

G Limitations and Future Work

For untied SAEs, our analysis establishes a local connection between self-consistency and coefficient recovery through the active encoder-decoder cross-map. The bound depends on cross-map conditioning and encoder amplification of dictionary mismatch and observation noise. Controlling the full cross-map through population self-consistency additionally requires a sufficiently rich active-code covariance in our analysis. Understanding when joint training maintains these conditions remains a direction for future work.

Our analysis focuses on coefficient refinement when the learned dictionary is close to the ground truth and the active support is correctly identified. The controlled synthetic experiments demonstrate coefficient improvements, including a fixed-decoder setting with exact support recovery for both the

baseline and regularized models. A natural next step is to study how self-consistency interacts with dictionary learning and support selection during joint training.

H BROADER IMPACTS

This work advances the theoretical and practical foundations of sparse autoencoders (SAEs) as tools for interpretability by clarifying when reconstruction accuracy and dictionary recovery fail to guarantee recovery of the underlying latent concepts. We identify intrinsic sources of latent recovery error and introduce a lightweight self-consistency regularizer that improves latent stability without increasing model capacity or architectural complexity. By improving the reliability of learned representations, this work may support the development of more faithful mechanistic interpretability methods, reducing the risk of misleading conclusions drawn from reconstruction-based analyses.

On the negative side, more reliable interpretability tools could create a false sense of confidence if their guarantees are overstated or applied outside their assumptions; we therefore emphasize that our recovery guarantee is local and tied to the near-correct regime, and we encourage downstream users to interpret SAE features accordingly. Beyond this, the contributions are theoretical and methodological in nature, do not introduce new pretrained models or datasets, and we do not foresee additional direct negative societal impacts.