DIAGNOSING AND FIXING LATENT RECOVERY IN SPARSE AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) have recently seen rapidly increasing use as a tool for interpreting representations in large models. Despite their widespread adoption, training objectives for SAEs primarily focus on accurate reconstruction of observed data, often implicitly assuming that perfect reconstruction or dictionary recovery implies recovery of the underlying latent variables. Even in the ideal case of exact observation reconstruction and correct dictionary recovery, recovery of latent variables (concepts) is not guaranteed. We develop a theoretical analysis of latent recovery in SAEs, combining an upper-bound analysis of support-wise recovery error with a lower-bound certificate based on latent self-consistency. The upper-bound analysis highlights error induced by dictionary coherence and sparsity, while the lower bound reveals an intrinsic source of error arising from unstable encoder–decoder dynamics in latent space. Motivated by this lower bound, we introduce a simple latent self-consistency regularizer that can be applied off-the-shelf to existing SAE architectures without architectural changes. Experiments on synthetic and real datasets demonstrate that this regularizer consistently improves latent recovery and representation quality across a wide range of settings.

1 INTRODUCTION

Large neural networks have demonstrated remarkable ability to learn rich internal representations that generalize effectively across a wide range of downstream tasks Brown et al. (2020); Caron et al. (2021); Wiedemer et al. (2025); Bengio et al. (2013). Despite their strong empirical performance, a central challenge remains their black-box nature: interpreting these learned representations is difficult due to their highly entangled and polysemantic structure, where individual neurons or directions often encode multiple, overlapping concepts. Recently, it has been hypothesized that neural networks may implicitly encode monosemantic features as sparse linear combinations within these otherwise distributed representations Elhage et al. (2022). Motivated by this hypothesis, sparse autoencoders (SAEs) Ng et al. (2011) have re-emerged as a promising tool for disentangling and recovering monosemantic features from neural network activations. These methods have been increasingly applied to large language models Cunningham et al. (2023), vision models Fel et al. (2025a;b), and multimodal systems Pach et al. (2025), with the goal of improving interpretability by identifying concept-level features.

From a theoretical perspective, a central question is under what conditions SAEs can faithfully recover ground-truth concepts. As highlighted in Cui et al. (2025), even under idealized settings, where reconstruction error on the observations is minimized and the learned dictionary aligns with monosemantic features, there typically remains a nontrivial gap between the learned latent representations and the true underlying concepts.

In this work, we formally characterize this gap by deriving complementary upper and lower bounds on the latent recovery error. The upper bound captures error induced by properties of the learned dictionary, including its sparsity and coherence, as well as the choice of activation function used in the SAE. The lower bound, in contrast, reveals an intrinsic limitation arising from unstable encoder–decoder dynamics in the latent space Fumero et al. (2025). Motivated by this lower bound, we propose a simple latent self-consistency regularizer that encourages contractive behavior in the sparse autoencoder. This regularizer can be applied off-the-shelf to existing SAE architectures without

requiring architectural modifications, and directly targets the fundamental source of latent recovery error identified by our analysis. Our key contributions are as follows:

- **Theory:** We provide a theoretical analysis of latent recovery in SAEs, combining an upper-bound analysis of support-wise recovery error with a lower-bound self-consistency certificate. The upper bound quantifies coherence-induced cross-talk (with explicit sparsity/coherence dependence), while the lower bound provides an intrinsic self-consistency certificate; as corollaries, our results recover and generalize the phenomena reported in Cui et al. (2025).
- **Method:** Motivated by the theory, we propose a simple latent self-consistency regularizer that can be added off-the-shelf to standard SAE objectives without architectural changes nor additional parameters.
- **Empirics:** We validate this regularizer on synthetic and real datasets, showing consistent improvements in latent recovery in controlled settings and representation quality across most settings.

2 Problem Setting & SAE Background

2.1 Observation model (blind mixing)

We consider a latent-variable observation model where only mixed measurements are observed. Let $x \in \mathbb{R}^{d_m}$ denote an unobserved latent vector and let $W_p \in \mathbb{R}^{d_p \times d_m}$ be an unknown mixing matrix. The learner observes

$$x_p = W_p x, \tag{1}$$

and the training set consists of samples $\{x_p\}$ only.

Overcomplete regime. We focus on the overcomplete setting $d_m \gg d_p$, which is typical in sparse autoencoders. Accordingly, the mixing matrix satisfies $W_p \in \mathbb{R}^{d_p \times d_m}$ with $d_p < d_m$, so the observation model Equation (1) is non-invertible in general.

2.2 Sparse autoencoder with tied weights

A sparse autoencoder (SAE) maps an input $x_p \in \mathbb{R}^{d_p}$ to a code via an encoder $E : \mathbb{R}^{d_p} \mapsto \mathbb{R}^{d_m}$ and reconstructs it via a decoder $D : \mathbb{R}^{d_m} \mapsto \mathbb{R}^{d_p}$. We consider a tied-weight SAE where the code dimension matches the latent dimension d_m and the encoder and decoder are parametrized by the same matrix W_m:

$$E_{W_m}(x_p) = \sigma(W_m x_p), \qquad D_{W_m}(z) = W_m^\top z, \tag{2}$$

where $W_m \in \mathbb{R}^{d_m \times d_p}$ is a learnable dictionary and $\sigma(\cdot)$ is a (typically sparsity-inducing) nonlinearity (e.g., ReLU or soft-thresholding). The reconstructed output is

$$\hat{x}_p = D_{W_m}(E_{W_m}(x_p)) = W_m^\top \sigma(W_m x_p). \tag{3}$$

When training the SAE on observed inputs $x_p \in \mathbb{R}^{d_p}$, we denote the learned code and reconstruction by

$$\begin{aligned} x_m &\triangleq E_{W_m}(x_p) = \sigma(W_m x_p) \in \mathbb{R}^{d_m}, \\ \hat{x}_p &\triangleq D_{W_m}(x_m) = W_m^\top x_m \in \mathbb{R}^{d_p}. \end{aligned} \tag{4}$$

2.3 Training objective (reconstruction with sparsity)

A standard SAE is trained by minimizing reconstruction error on the observed data, optionally with a sparsity/structure-inducing penalty on the code:

$$\min_W \ \mathbb{E}\big[\|x_p - \hat{x}_p\|_2^2\big] \ + \ \lambda\, \mathbb{E}[\Omega(x_m)]\,, \tag{5}$$

where $\Omega(\cdot)$ is a sparsity regularizer and $\lambda > 0$ controls its strength.

Remark. The observation model Equation (1) distinguishes two criteria of success: (i) *observation reconstruction*, i.e., accurately reconstructing x_p via Equation (3), and (ii) *latent recovery*, i.e., whether the learned code x_m captures information about the underlying latent x.

3 THEORETICAL RESULTS

3.1 WHY BOUNDING LATENT RECOVERY ERROR?

Objective Equation (5) enforces reconstruction in the observation space, but does not directly guarantee recovery of the underlying latent x in Equation (1). In fact, even under an idealized "perfect fit" regime Cui et al. (2025), where the learned dictionary matches the ground truth, $W_m = W_p^\top$, and reconstruction is exact, $\hat{x}_p = x_p$, the learned latent vector satisfies

$$\begin{aligned} x_m \;&=\; \sigma(W_m x_p) \;=\; \sigma(W_p^\top W_p x) \;=\; \sigma(Gx), \\ G &\triangleq W_p^\top W_p, \end{aligned} \tag{6}$$

which need not coincide with x unless G is sufficiently close to identity on the active support and σ preserves the relevant coordinates. Thus, perfect observation reconstruction and dictionary recovery do not, by themselves, certify latent recovery.

Motivated by this gap, we study the latent recovery error:

$$\mathcal{L}_{GT}(x_m, x) = \|x_m - x\|_2^2 \tag{7}$$

and derive an upper-bound analysis and a lower-bound certificate in terms of structural properties (e.g., sparsity and coherence of W_p) and intrinsic quantities induced by the SAE (e.g., a latent self-consistency residual). These results clarify when reconstruction may fail to reflect recovery, and guide algorithmic choices that favor more recoverable representations.

3.2 A SPARSITY–COHERENCE UPPER BOUND FOR LATENT RECOVERY

We first analyze an idealized regime where (i) the learned dictionary matches the ground-truth mixing and (ii) the SAE achieves zero reconstruction error on the observations. Concretely, we assume

$$W_m = W_p^\top, \qquad x_p = W_m^\top \sigma(W_m x_p) \text{ (i.e., } \hat{x}_p = x_p). \tag{8}$$

Recall $x_p = W_p x$ and $x_m = \sigma(W_m x_p)$. Under Equation (8),

$$x_m = \sigma(W_p^\top W_p x) = \sigma(Gx), \;\; G \triangleq W_p^\top W_p \in \mathbb{R}^{d_m \times d_m}. \tag{9}$$

The key obstruction is that G may have non-negligible off-diagonal entries, so each coordinate of the pre-activation Gx can contain *cross-talk* from other latent coordinates.

Normalized Gram and relative coherence. Let

$$D \triangleq \operatorname{diag}(G_{11}, \ldots, G_{d_x d_x}), \qquad \tilde{G} \triangleq D^{-1/2} G D^{-1/2}, \tag{10}$$

so that $\tilde{G}_{ii} = 1$ for all i. We define the relative coherence

$$\mu \triangleq \max_{i \neq j} |\tilde{G}_{ij}| \;=\; \max_{i \neq j} \frac{|G_{ij}|}{\sqrt{G_{ii} G_{jj}}}. \tag{11}$$

k-sparse latents and support-wise distortion. Assume the latent x is k-sparse with support $S = \operatorname{supp}(x)$, $|S| = k$. Define the rescaled latent $\tilde{x} \triangleq D^{1/2} x$. Then

$$D^{-1/2} G x = \tilde{G} \tilde{x}. \tag{12}$$

Restricting to S gives $(\tilde{G}\tilde{x})_S - \tilde{x}_S = (\tilde{G}_{SS} - I)\tilde{x}_S$. Since $\tilde{G}_{SS}$ has unit diagonal and off-diagonal entries bounded by μ, then Gershgorin circle theorem implies that:

$$\|\tilde{G}_{SS} - I\|_2 \leq (k-1)\mu. \tag{13}$$

Lemma 1 (Universal sparsity–coherence bound)**.** *Assume Equation* (8) *and σ is coordinate-wise 1-Lipschitz and positively homogeneous, (e.g., ReLU). With $G, D, \tilde{G}$ as in Equation* (9)*–Equation* (10) *and $\tilde{x} = D^{1/2} x$, if x is k-sparse then*

$$\|(D^{-1/2} x_m - \tilde{x})_S\|_2 \;\leq\; (k-1)\mu \, \|\tilde{x}\|_2 \;+\; \|(\tilde{x} - \sigma(\tilde{x}))_S\|_2. \tag{14}$$

In particular, when $k = 1$ the cross-talk term vanishes, and for ReLU one has $\|(\tilde{x} - \sigma(\tilde{x}))_S\|_2 = \|(\tilde{x}_-)_S\|_2$.

Proof. Proof in Appendix C □

Remark: Lemma 1 shows that even under perfect dictionary recovery, on-support latent recovery depends on two factors: (i) cross-talk controlled by $(k-1)\mu$, and (ii) distortion induced by the nonlinearity $\|(\tilde{x} - \sigma(\tilde{x}))_S\|_2$. Thus sparsity mitigates interference, while high coherence amplifies it. The full proof is deferred to Appendix C.

3.3 A SELF-CONSISTENCY LOWER BOUND FOR LATENT RECOVERY

We now derive a lower bound on latent recovery via an intrinsic *self-consistency residual* induced by the tied-weight SAE. Let W_m denote the learned dictionary and define the latent self-map

$$F_{W_m}(z) \triangleq \sigma\big(W_m W_m^\top z\big), \qquad r(z) \triangleq \|F_{W_m}(z) - z\|_2, \tag{15}$$

where σ is applied coordinate-wise (e.g., ReLU or soft-thresholding). Intuitively, a model-consistent code should lie close to a fixed point of F_{W_m}.

Connecting F_{W_m} to the learned code. Recall the observation model $x_p = W_p x$ and the SAE encoding $x_m = \sigma(W_m x_p)$. In the idealized regime considered in Section 3.2, namely, perfect reconstruction on the observations and dictionary matching $W_m = W_p^\top$, we can rewrite

$$x_m = \sigma(W_m x_p) = \sigma(W_p^\top W_p x) = \sigma(W_m W_m^\top x) = F_{W_m}(x) \tag{16}$$

Moreover, the decode–encode cycle applied to x_m yields

$$x_t \triangleq F_{W_m}(x_m) = \sigma(W_m W_m^\top x_m), \tag{17}$$

so the observable *cycle gap* is precisely $r(x_m) = \|x_t - x_m\|_2$.

Theorem 2 (Self-consistency lower bound for latent recovery)**.** *Consider the blind-mixing model $x_p = W_p x$ and the SAE code $x_m = \sigma(W_m x_p)$. In the idealized regime where the learned dictionary matches the mixing, i.e., $W_m = W_p^\top$, define*

$$F(z) \triangleq \sigma(W_p^\top W_p z).$$

Then $x_m = F(x)$. Assume further that F is L-Lipschitz on a region $\mathcal{U} \subseteq \mathbb{R}^{d_x}$ containing $\{x, x_m\}$, i.e.,

$$\|F(u) - F(v)\|_2 \le L\|u - v\|_2, \qquad \forall u, v \in \mathcal{U}. \tag{18}$$

Then the latent recovery error satisfies

$$\|x_m - x\|_2 \;\ge\; \frac{\|x_t - x_m\|_2}{L}, \qquad x_t \triangleq F(x_m). \tag{19}$$

Proof. Proof in Appendix C □

Corollary 3 (ReLU specialization)**.** *Let $\sigma = \mathrm{ReLU}$ and consider the idealized dictionary matching regime $W_m = W_p^\top$ in Theorem 2. Then $F(z) = \mathrm{ReLU}(W_p^\top W_p z)$ is L-Lipschitz with*

$$L \le \|W_p^\top W_p\|_2 = \|W_p\|_2^2. \tag{20}$$

Consequently,

$$\|x_m - x\|_2 \;\ge\; \frac{\|x_t - x_m\|_2}{\|W_p\|_2^2}, \;\; x_t = \mathrm{ReLU}(W_p^\top W_p x_m). \tag{21}$$

Proof. Proof in Appendix C □

Remarks on Top-K activation. For Top-K encoders, sparsity is enforced structurally by retaining the K largest coordinates of the pre-activation. In the dictionary-matching regime $W_m = W_p^\top$ with exact reconstruction, the learned code satisfies

$$x_m = \Pi_K(W_p^\top W_p x),$$

where $\Pi_K(\cdot)$ denotes the Top-K operator.

While Π_K is not globally smooth (due to possible changes of the active set when the K-th and $(K+1)$-th coordinates are close), it is 1-Lipschitz on any region where the Top-K support is fixed: in that case Π_K reduces to an orthogonal coordinate projection. Consequently, on such a region the induced latent self-map

$$F(z) = \Pi_K(W_m W_m^\top z)$$

is L-Lipschitz with

$$L \le \|W_m W_m^\top\|_2 = \|W_m\|_2^2.$$

Therefore, the self-consistency lower bound in Theorem 2 applies directly to Top-K encoders on stable-support regimes.

4 ALGORITHM

Motivated by the self-consistency lower bound in Theorem 2, we propose a simple latent self-consistency regularizer that explicitly penalizes unstable latent dynamics. Given an input activation $x_p \in \mathbb{R}^{d_p}$, the encoder produces a sparse latent representation

$$x_m = \sigma(W_m x_p), \tag{22}$$

where $W_m \in \mathbb{R}^{d_m \times d_p}$ is the learned dictionary and $\sigma(\cdot)$ denotes a sparsifying nonlinearity (e.g., ReLU or Top-K). The decoder reconstructs the input via tied weights,

$$\hat{x}_p = W_m^\top x_m. \tag{23}$$

To measure latent stability under the induced encode–decode dynamics, we apply one additional decode–encode step,

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

where F_{W_m} denotes the latent self-map induced by the SAE. Following Theorem 2, the deviation $\|x_t - x_m\|_2$ serves as an observable *certificate* of intrinsic latent recovery error. We therefore introduce the latent self-consistency regularization term

$$\mathcal{L}_{\text{cert}} \triangleq \mathbb{E}_{x_p}\big[\|x_t - x_m\|_2^2\big], \tag{25}$$

which encourages latent codes to become approximate fixed points of F_{W_m}.

Remark (Top-K certificate). As discussed in Section 3, for Top-K encoders we compute the certificate only on the *in-support* coordinates to avoid penalizing support switching. Concretely, letting $S = \mathrm{supp}(x_m)$ be the Top-K support selected in the first encoding pass, we use $\|(x_t - x_m)_S\|_2^2$ and set the weight on the out-of-support part to zero (implemented via a masked difference, optionally detaching the mask). The full training objective is:

$$\min_{W_m} \; \mathbb{E}_{x_p}\big[\|x_p - \hat{x}_p\|_2^2\big] \; + \; \beta\, \mathbb{E}_{x_p}[\Omega(x_m)] \; + \; \lambda\, \mathcal{L}_{\text{cert}}, \tag{26}$$

where $\Omega(\cdot)$ denotes a sparsity penalty (e.g., ℓ_1 for ReLU-based SAEs), $\beta \ge 0$ controls sparsity strength, and $\lambda \ge 0$ controls the contribution of the self-consistency regularizer. For Top-K encoders, sparsity is enforced structurally and the term $\Omega(\cdot)$ may be omitted. The proposed regularization requires only one additional encoder pass per training example and introduces no auxiliary parameters.

Remark. Our regularizer does not rely on any specific activation function or architectural constraint. They can be incorporated off-the-shelf into existing SAE implementations, including models with arbitrary sparsifying nonlinearities and untied weights.

5 EMPIRICAL EVIDENCE

We empirically validate that the proposed certificate $\|x_t - x_m\|_2^2$ (Eq. 25) provides a useful training signal and that enforcing latent self-consistency improves latent recovery and representation quality. We evaluate in two regimes: (i) a synthetic blind-mixing setting where ground-truth latents are available, and (ii) CE-bench Gulko et al. (2025) on SAE features trained from `gpt2` residual-stream activations. Across experiments, we report both task-level metrics (latent recovery error or CE-bench scores) and dictionary-level health diagnostics (dead fraction, usage entropy, coherence RMS) to ensure gains are not explained by trivial activation-density shortcuts. Detailed sweeps, plots, and implementation details are provided in Appendix B.

Synthetic latent recovery. In the synthetic study (Sec. B.2), we sweep the regularization weight λ and directly measure the true latent recovery error $\mathcal{L}_{GT} = \|x_m - x\|_2$ (Eq. 7) together with the reconstruction error $\|x_p - \hat{x}_p\|_2$. Fig. 1 shows that adding $\lambda \mathcal{L}_{\text{cert}}$ consistently reduces $\|x_m - x\|_2$ compared to reconstruction-only training over a broad range of λ, while reconstruction remains stable under the same sweep. We also compare against objective-reweighting baselines (WSAE), which are noticeably more sensitive to sparsity and do not yield comparably consistent reductions in $\|x_m - x\|_2$ across k (Fig. 1). Fig. 2 confirms that the proxy is optimized as intended: increasing λ monotonically decreases the certificate value. Finally, Fig. 3 shows that the certificate is more informative in sparser regimes (smaller k), motivating stronger sparsity control (e.g., Top-K with small K) for training and model selection.

Real-model representation quality. On CE-bench Gulko et al. (2025), we report Contrastive, Independence, and $\text{CE}_{\text{total}} = \text{Contrastive} + \text{Independence} - \alpha \cdot \text{Sparsity}$ with $\alpha = 0.25$, together with health diagnostics (dead fraction, usage entropy, coherence RMS).

For ReLU+ℓ_1 SAEs (Tab. 1), self-consistency regularization improves CE-bench scores and yields substantially healthier dictionaries: in particular, the dead feature fraction drops by **62%** (from 0.084 ± 0.008 to 0.032 ± 0.019), alongside higher usage entropy. For Top-K SAEs (Tab. 2), we observe even larger dictionary health gains under self-consistency regularization, with the dead feature fraction reduced by **82%** (from 0.160 ± 0.039 to 0.029 ± 0.006), together with consistent improvements in CE-bench scores.

These results indicate that enforcing latent self-consistency not only improves representation quality but also makes the learned dictionaries markedly more usable (substantially fewer dead features and more balanced utilization).

6 Conclusion

In this work, we showed that accurate reconstruction and even correct dictionary recovery are insufficient to guarantee recovery of the underlying latent variables. We formalized this gap through a theoretical analysis of latent recovery in SAEs, combining an upper-bound analysis of on-support recovery error driven by sparsity and dictionary coherence with a lower-bound self-consistency certificate that exposes an intrinsic error source arising from unstable encoder-decoder dynamics. We introduced a simple latent self-consistency regularizer that directly targets this failure mode. The regularizer is lightweight, requires no architectural changes, and can be applied off-the-shelf to standard SAE objectives. In controlled synthetic settings, optimizing this proxy reliably reduces true latent recovery error compared to reconstruction-only training and objective reweighting baselines. On real-model benchmarks, it consistently improves representation quality and dictionary health, yielding more balanced and less degenerate feature dictionaries without relying on stronger sparsity alone.

Limitations and future work. Our theoretical analysis focuses on tied-weight SAEs, which enables a clean characterization of the latent self-map $F(z) = \sigma(WW^\top z)$. Extending recovery guarantees to untied architectures remains open, as untied autoencoders introduce additional non-identifiability through families of equivalent solutions Bao et al. (2020), while regularization and weight tying can qualitatively affect representation structure and optimization dynamics Kunin et al. (2019); Refinetti & Goldt (2022). Importantly, instability of the induced latent dynamics does not depend on weight tying: in the untied case, the self-map becomes $F(z) = \sigma(W_{\text{enc}} W_{\text{dec}} z)$. From this perspective, our self-consistency regularizer enforces latent dynamical stability, conceptually related to Fumero et al. (2025). Analyzing stability properties of $W_{\text{enc}} W_{\text{dec}}$ to derive recovery bounds for untied architectures is a natural future direction.

More broadly, we hope this work encourages the development of interpretability methods that are grounded not only in reconstruction performance, but in explicit guarantees about latent-space behavior.

REFERENCES

Xuchan Bao, James Lucas, Sushant Sachdeva, and Roger B Grosse. Regularized linear autoencoders recover the principal components, eventually. *Advances in Neural Information Processing Systems*, 33:6971–6981, 2020.

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, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

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

Emmanuel J Candès et al. Compressive sampling. In *Proceedings of the international congress of mathematicians*, volume 3, pp. 1433–1452. Madrid, Spain, 2006.

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.

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.

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.

Daniel Kunin, Jonathan Bloom, Aleksandrina Goeva, and Cotton Seed. Loss landscapes of regularized linear autoencoders. In *International conference on machine learning*, pp. 3560–3569. PMLR, 2019.

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.

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.

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.

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.

Maria Refinetti and Sebastian Goldt. The dynamics of representation learning in shallow, non-linear autoencoders. In *International Conference on Machine Learning*, pp. 18499–18519. PMLR, 2022.

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.

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

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

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).

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). This has also been observed empirically, demonstrating the instability of SAEs solutions under retraining Paulo & Belrose (2025); Fel et al. (2025a). These results emphasize a distinction between *dictionary recovery* and *latent recovery*, and show that reconstruction-based objectives alone are insufficient to certify that an SAE has learned the intended concepts. Our work builds directly on this line of inquiry by characterizing both upper and lower bounds on latent recovery error and by proposing an explicit regularizer that targets latent self-consistency.

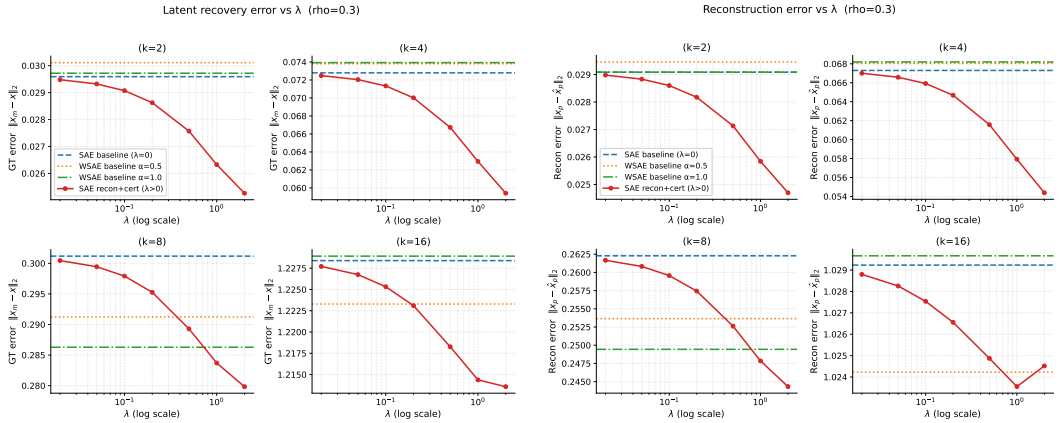
Sparse coding and dictionary learning. SAEs are closely related to classical sparse coding and dictionary learning methods Rubinstein et al. (2010); Tibshirani (1996); Olshausen & Field (2004); Lee et al. (2006); Tošić & Frossard (2011); Kreutz-Delgado et al. (2003). These approaches aim to represent 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 support recovery and stability under assumptions such as sparsity Spielman et al. (2012); Gribonval & Schnass (2010), incoherence, or restricted isometry Candes (2008), particularly in compressed sensing Donoho (2006); Candès et al. (2006); Duarte & Eldar (2011); Lopes (2013). However, these guarantees typically concern recovery of sparse codes given a fixed dictionary, rather than recovery of semantically meaningful latent variables in nonlinear encoder–decoder systems. Our analysis can be viewed as extending this tradition to tied-weight, nonlinear autoencoders, highlighting both the role of coherence-induced cross-talk and intrinsic instability introduced by the encoder–decoder dynamics.

B Experiments

B.1 Metrics and evaluation protocol

Across experiments, we evaluate both *task-level* representation scores and *dictionary-level* health diagnostics.

Synthetic experiments In the synthetic study of section B.2, where the ground-truth (GT) latent code is available, we directly report the latent recovery error $\mathcal{L}_{GT}$ in Equation (7) as the primary objective, together with the reconstruction error $\|x_p - \hat{x}_p\|_2$. In addition, to assess whether the certificate provides a meaningful learning signal, we quantify its informativeness by reporting the Spearman rank correlation between per-sample certificate values and per-sample latent errors. This



(a) **Latent recovery vs. λ.** We sweep the certificate weight λ (log-scale) and report the ground-truth latent recovery error $\|x_m - x\|_2$ for different sparsity levels k.

(b) **Reconstruction vs. λ.** Under the same sweep, we report the reconstruction error $\|x_p - \hat{x}_p\|_2$.

Figure 1: **Synthetic study.** Sweeping λ shows how certificate regularization affects latent recovery and reconstruction across sparsity levels k.

controlled evaluation allows us to test whether the proxy aligns with the true latent objective and whether certificate regularization improves the intended latent recovery, rather than merely reshaping reconstruction.

Real-model experiments For real-model evaluations we follow the CE-bench benchmark Gulko et al. (2025) and report *Contrastive* and *Independence* scores, together with the *sparsity* level (lower is better). The overall CE-bench score is computed as $\mathrm{CE}_{\mathrm{total}} = \mathrm{Contrastive} + \mathrm{Independence} - \alpha \cdot \mathrm{Sparsity}$, and we use the standard choice $\alpha = 0.25$ in all experiments unless stated otherwise. Contrastive measures the strength of view-specific signal captured by the representation (higher is better), while Independence measures how well the shared component is disentangled from the contrasting component (higher is better).

However, CE-bench alone does not fully characterize the quality of the learned SAE dictionary: improvements can be confounded by shortcuts such as increasing activation density or reallocating usage in a way that increases CE scores without yielding a healthier, more interpretable representation. To diagnose these effects, we additionally report dictionary health metrics that are largely orthogonal to CE-bench: *dead fraction* (the proportion of rarely activated latent units; lower is better), *usage entropy* (how evenly units are utilized; higher is better), and *coherence RMS* (correlation among dictionary atoms; lower is better). More details can be found in Appendix D.

Together, these metrics serve as a sanity check for our method: they help verify that our regularizer improves representation quality by producing healthier and less redundant dictionaries, rather than improving CE-bench through changes in activation density or feature usage alone.

B.2 Synthetic Study

We first validate our self-consistency regularizer in a controlled synthetic setting where the *ground-truth* latent code is known. This synthetic study serves two purposes: (i) to test our central claim that the certificate term provides a usable proxy signal for latent recovery error, and (ii) to examine whether adding the certificate as a regularizer can *actually improve training* with respect to the true latent objective, rather than merely reshaping reconstruction. In addition, we include WSAE Cui et al. (2025) as a representative baseline that modifies the training objective via weighting, and ask and compare whether such reweighting alone can reliably reduce the true latent error.

Setting. We consider a blind-mixing model. For each sample, we draw a k-sparse latent code $x \in \mathbb{R}^n$ and form an observed activation $x_p = W_p x \in \mathbb{R}^d$, where $W_p \in \mathbb{R}^{d \times n}$ is a random dictionary with controllable column correlation (parameterized by ρ). We train an SAE with encoder

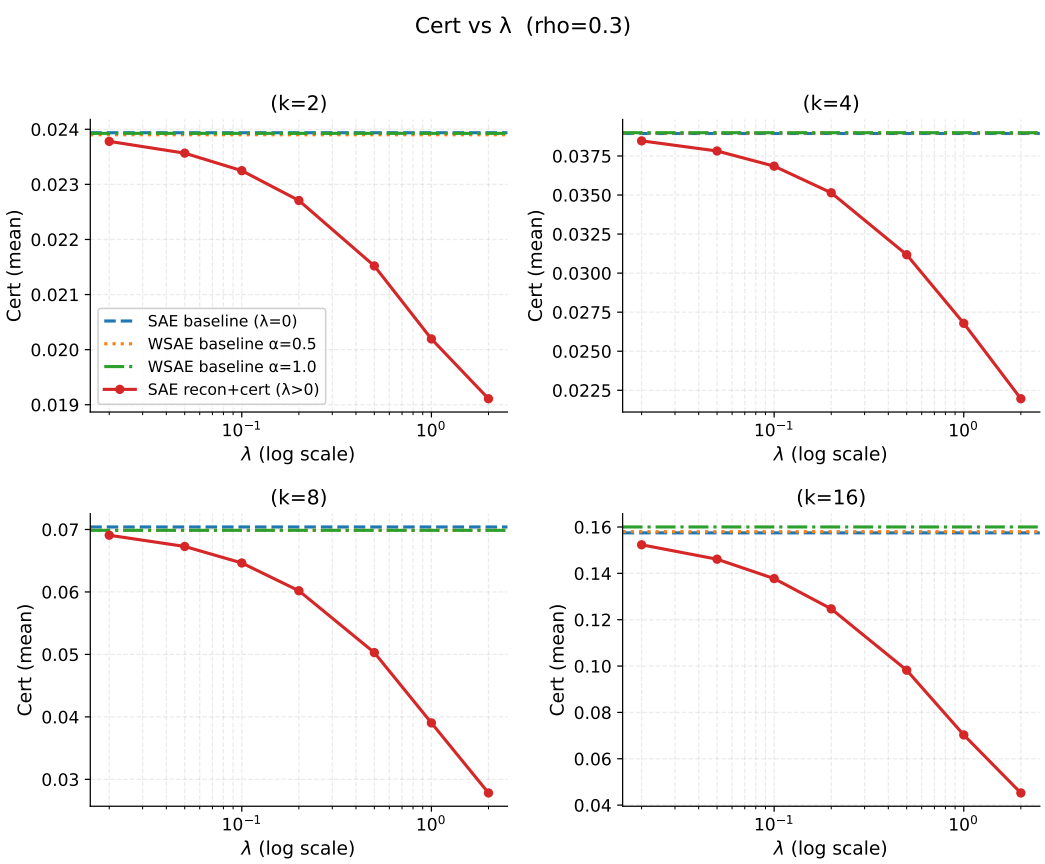


Figure 2: **Certificate vs. λ.** The certificate decreases with increasing λ, confirming that the regularizer optimizes its intended proxy objective.

$x_m = \sigma(x_p W^\top)$ and decoder $\hat{x}_p = x_m W$, where $\sigma(\cdot)$ is ReLU. To align with the idealized "perfect-fit" regime, we adopt an *oracle-decoder* setup: we fix the decoder to the ground-truth dictionary $W_p^\top$ (i.e., $\hat{x}_p = x_m W_p^\top$) and optimize only the encoder parameters. We evaluate (i) the latent recovery error $\mathcal{L}_{GT} = \|x_m - x\|_2$ (the true objective of interest), (ii) reconstruction error $\|x_p - \hat{x}_p\|_2$, and (iii) the certificate cert computed from the SAE self-map induced by `decode→encode`. We sweep the certificate weight λ in the training objective $\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{cert}}$ and vary sparsity k; results are averaged over 30 random seeds. As additional baselines, we also report WSAE with two standard choices of the weighting exponent ($\alpha \in \{0.5, 1\}$, as set in Cui et al. (2025)), evaluated under the same settings.

Analysis of results. Figures 1a and 1b report the main effect of regularization by sweeping λ for multiple sparsity levels k, together with the SAE and WSAE baselines. Across a wide range of λ, adding the certificate term consistently improves the true latent recovery error $\|x_m - x\|_2$ relative to the reconstruction-only baseline ($\lambda = 0$), showing that the regularizer can directly optimize the intended latent objective in this controlled regime. Importantly, the improvement is not merely a consequence of sacrificing reconstruction: the reconstruction error does not degrade catastrophically and often improves in tandem with $\mathcal{L}_{GT}$ over the same sweep. In contrast, WSAE does not yield a comparable or consistent reduction in the true latent error across sparsity levels: its performance is noticeably more sensitive to k and can be worse than the plain SAE baseline in regimes where accurate latent recovery is most challenging. This comparison highlights that explicitly regularizing the *latent self-consistency* of the SAE self-map is more effective than objective reweighting for reducing the true latent recovery error. Figure 2 further confirms that our method optimizes the intended proxy objective. Across all sparsity levels $k \in \{2, 4, 8, 16\}$, increasing the certificate weight λ monotonically reduces the average certificate value under SAE+RECON+CERT, while the SAE

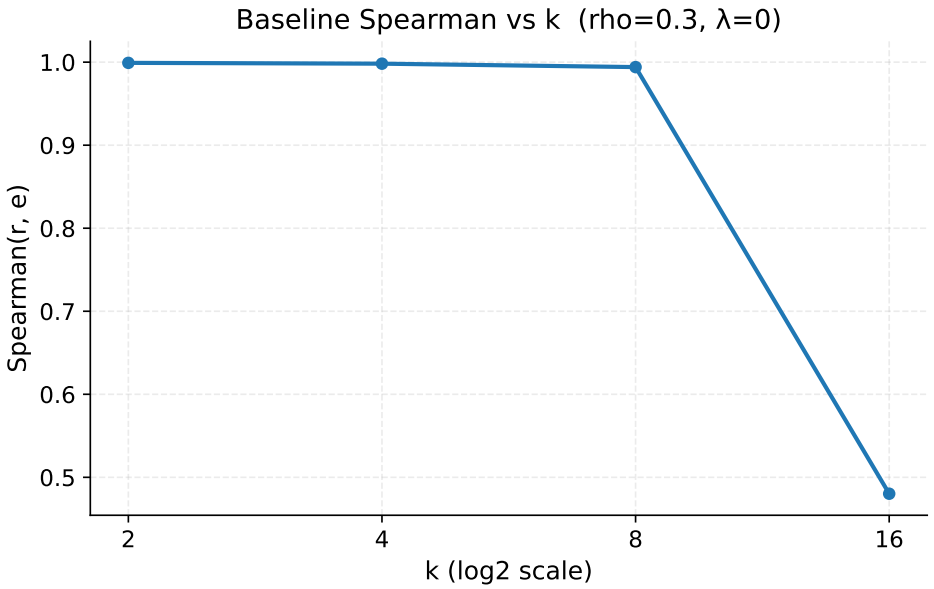


Figure 3: **Certificate: GT alignment improves with sparsity.** Baseline ($\lambda = 0$) Spearman correlation between per-sample certificate values and true latent errors versus k. Smaller k yields stronger alignment, motivating later TopK/sparser (small K) choices.

Metric	No reg	+Reg (λ=10)	Change
Dead frac ↓	0.0844 ± 0.0079	**0.0320 ± 0.0186**	**-62.1% (better)**
Usage entropy ↑	0.9049 ± 0.0013	**0.9403 ± 0.0110**	**+3.92% (better)**
Coherence RMS ↓	0.0346 ± 0.0001	**0.0339 ± 0.0004**	**-2.14% (better)**
Contrastive ↑	0.8374 ± 0.0029	**0.8484 ± 0.0052**	**+1.32% (better)**
Independence ↑	0.8655 ± 0.0032	**0.8782 ± 0.0044**	**+1.47% (better)**
Sparsity ↓	**0.3137 ± 0.0010**	0.3432 ± 0.0134	**+9.39% (worse)**
CE total ↑	1.6245 ± 0.0049	**1.6408 ± 0.0065**	**+1.01% (better)**

Table 1: **CE-bench (ReLU+L1).** Mean ± std over 9 runs. CE-bench settings: N_{pairs}=3000, d_{latent}=2048, α=0.25. Reg uses the proposed self-consistency objective with fixed coefficient λ=10. Arrows indicate whether lower/higher is better for each metric; last column reports whether the change is *better* or *worse* accordingly.

and WSAE baselines remain essentially flat. This observation rules out the possibility that the improvements in ground-truth latent recovery are incidental: the regularizer produces a consistent, controllable reduction in the proxy it is designed to minimize, which aligns with the reductions in $\mathcal{L}_{GT}$ in Figure 1a.

Takeaway. Taken together, the synthetic study supports one main conclusion aligned with our theory: optimizing a latent discrepancy proxy can improve true latent recovery. In particular, certificate regularization yields consistent reductions in the latent recovery error $\mathcal{L}_{GT}$ compared to both reconstruction-only training and WSAE-style reweighting baselines, while maintaining stable reconstruction behavior over the same sweep. Separately, we also observe that the certificate becomes more aligned with the ground-truth latent error in sparser regimes (smaller k), which motivates our later emphasis on stronger sparsity control (e.g., TopK with small K) so that training and model selection operate in regimes where the proxy is more informative.

Metric	No reg	+Reg (λ=0.5)	Change
Dead frac ↓	0.1599 ± 0.0386	**0.0286 ± 0.0058**	**-82.14% (better)**
Usage entropy ↑	0.8124 ± 0.0040	**0.8477 ± 0.0123**	**+4.34% (better)**
Coherence RMS ↓	0.0387 ± 0.0003	**0.0373 ± 0.0001**	**-3.57% (better)**
Contrastive ↑	0.8832 ± 0.0054	**0.9145 ± 0.0030**	**+3.54% (better)**
Independence ↑	0.8823 ± 0.0053	**0.9051 ± 0.0024**	**+2.59% (better)**
Sparsity (fixed) ↓	0.0156 ± 0.0000	**0.0156 ± 0.0000**	+0.00% (same)
CE total ↑	1.7616 ± 0.0103	**1.8157 ± 0.0052**	**+3.07% (better)**

Table 2: **CE-bench (TopK, K=32).** Mean ± std over 5 runs. CE-bench settings: N_{pairs}=3000, d_{latent}=2048, α=0.25. Reg uses the proposed self-consistency objective with fixed coefficient λ=0.5. Arrows indicate whether lower/higher is better for each metric; last column reports whether the change is *better*, *worse*, accordingly.

B.3 Experiments on CE Benchmark

Setting. We evaluate our proposed self-consistency regularizer on the CE-bench benchmark using SAE features trained on `gpt2` Radford et al. (2019) residual-stream activations. Unless otherwise stated, we extract hidden states from the second-to-last layer, tokenize to a maximum length of 128, and sample 16 sequences per step (yielding 16,384 tokens/step). We train for 3,000 optimization steps with learning rate 10^{-3} and latent dimension d_{latent}=2048. CE-bench evaluation uses N_{pairs}=3000 and aggregation weight α=0.25 following the CE-bench protocol, where $\text{CE}_{\text{total}} = \text{Contrastive} + \text{Independence} - \alpha \cdot \text{Sparsity}$. Motivated by our theory that reconstruction can be misleading for latent quality, we report the standard CE-bench scores and, in addition, independent dictionary health diagnostics (dead fraction, usage entropy, and coherence) that quantify feature utilization and redundancy directly.

B.3.1 ReLU activation + L1 sparsity constraint

In this section, the SAE uses a ReLU encoder $x_m = \text{ReLU}(W_{\text{enc}}x + b)$ and a linear decoder $\hat{x}_p = W_{\text{dec}}x_m$, trained with a reconstruction loss plus an ℓ_1 penalty on x_m; our method additionally includes the proposed self-consistency objective Equation (25). Unless otherwise stated, we use *untied parameters* for W_{enc} and W_{dec}.

Analysis of results. Table 1 reports mean±std over 9 runs (with one-to-one matched seeds between our regularization method and the baseline). Adding the self-consistency regularizer improves CE-bench representation quality: both `contrastive` ($0.837 \pm 0.003 \rightarrow 0.848 \pm 0.005$) and `independence` ($0.865 \pm 0.003 \rightarrow 0.878 \pm 0.004$) increase, leading to a higher overall `CE total` ($1.624 \pm 0.005 \rightarrow 1.641 \pm 0.007$). Notably, these gains occur despite a modest *worsening* of the CE-bench sparsity term (active fraction increases from 0.314 ± 0.001 to 0.343 ± 0.013), which is penalized in CE_{total}; this suggests the improvement is driven by stronger contrastive and independence components rather than sparsity changes alone. Beyond the CE-bench scores, we observe substantial improvements in dictionary health diagnostics, indicating that the regularizer changes the internal structure of the learned feature dictionary rather than only improving the benchmark objective. In particular, the dead feature fraction drops by 62% ($0.084 \pm 0.008 \rightarrow 0.032 \pm 0.019$), meaning that a much larger portion of the features becomes actively used instead of remaining effectively unused. At the same time, the usage entropy increases ($0.905 \pm 0.001 \rightarrow 0.940 \pm 0.011$), suggesting that activation mass is distributed more evenly across features and the representation is less dominated by a small subset of units. We also observe a slight reduction in coherence RMS ($0.0346 \pm 0.0001 \rightarrow 0.0339 \pm 0.0004$), consistent with reduced redundancy and weaker cross-talk between dictionary atoms. Notably, these improvements in health metrics occur even though the CE-bench sparsity term becomes worse (active fraction increases), implying that the regularizer does not simply enforce "more sparsity"; instead, it promotes broader and more balanced utilization of the dictionary while mildly reducing inter-feature redundancy. Overall, the dictionary health results

provide complementary evidence that certificate regularization yields more usable and less degenerate feature dictionaries, which aligns with the observed gains in CE-bench representation quality.

Takeaway. For ReLU-based SAEs, enforcing latent self-consistency improves CE-bench performance and substantially strengthens dictionary health, indicating that the gains arise from more stable and better-utilized latent features rather than from reconstruction or sparsity effects alone.

B.3.2 TopK activation

In this section, we replace the ReLU+ℓ_1 sparsity mechanism with a Top-K encoder that enforces exact sparsity by construction. Concretely, we compute pre-activations $u = W_{\text{enc}}x_p + b$ and form the latent code $x_m = \Pi_K(u)$ where $\Pi_K(\cdot)$ retains the K largest coordinates of its input (and zeros out the rest). We then decode linearly as $\hat{x}_p = W_{\text{dec}}x_m$. The model is trained with the same reconstruction objective, and our method additionally includes the proposed self-consistency objective Equation (25). Unless otherwise stated, we use untied parameters, i.e., W_{enc} and W_{dec} are learned separately. With TopK, the active fraction is fixed (here K=32 and d_{latent}=2048, so the sparsity term in CE-bench is constant across methods), allowing a cleaner attribution of CE-bench improvements to representation quality rather than changes in activation density.

Analysis of results. Table 2 reports mean±std over 5 matched-seed runs. Under fixed sparsity, adding the self-consistency regularizer yields a clear and consistent improvement on CE-bench: `contrastive` increases from 0.8832 ± 0.0054 to 0.9145 ± 0.0030 and `independence` increases from 0.8823 ± 0.0053 to 0.9051 ± 0.0024, resulting in a higher overall `CE total` ($1.7616\pm0.0103 \rightarrow 1.8157\pm0.0052$, $+3.07\%$). Since the CE-bench sparsity term is identical across the baseline and our method (`sparsity` $= 0.0156$), these gains cannot be explained by sparsity changes and directly reflect stronger contrastive and disentangling behavior of the learned representations. Beyond CE-bench, the dictionary health metrics improve even more dramatically: the dead feature fraction drops from 0.1599 ± 0.0386 to 0.0286 ± 0.0058 (-82.14%), indicating that TopK features become substantially less degenerate under regularization despite the same enforced activation budget. Usage entropy also increases ($0.8124\pm0.0040 \rightarrow 0.8477\pm0.0123$), suggesting a more balanced allocation of activations across units, while coherence RMS decreases slightly ($0.0387\pm0.0003 \rightarrow 0.0373\pm0.0001$), consistent with reduced redundancy among dictionary atoms. Overall, the TopK results strengthen the main takeaway from CE-bench: even when sparsity is held fixed, enforcing latent self-consistency improves both representation quality (CE-bench) and the health of the learned feature dictionary.

Takeaway. Under fixed sparsity with Top-K encoders, self-consistency regularization yields clear improvements in CE-bench scores and dictionary health, demonstrating that stabilizing latent dynamics improves representation quality even when activation density is held constant.

C Missing proofs in Section 3

Proof of Lemma. 1. Under $W_m = W_p^\top$ and $x_p = W_p x$, the SAE code satisfies

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

Moreover, with $\tilde{x} = D^{1/2}x$ and $\tilde{G} = D^{-1/2}GD^{-1/2}$ we have

$$D^{-1/2}Gx = (D^{-1/2}GD^{-1/2})(D^{1/2}x) = \tilde{G}\tilde{x}.$$

Since σ is coordinate-wise and $D^{-1/2}$ is a positive diagonal scaling, they commute for positively homogeneous nonlinearities such as ReLU; equivalently, for ReLU one has $D^{-1/2}\sigma(u) = \sigma(D^{-1/2}u)$ for all u. Hence

$$D^{-1/2}x_m = D^{-1/2}\sigma(Gx) = \sigma(D^{-1/2}Gx) = \sigma(\tilde{G}\tilde{x}). \tag{27}$$

Let $S = \operatorname{supp}(x) = \operatorname{supp}(\tilde{x})$. Restricting to the support S and applying the triangle inequality gives

$$\|(D^{-1/2}x_m - \tilde{x})_S\|_2 \le \|(D^{-1/2}x_m - \sigma(\tilde{x}))_S\|_2 + \|(\sigma(\tilde{x}) - \tilde{x})_S\|_2. \tag{28}$$

For the first term, using equation 27, restriction to S, and the 1-Lipschitz property of σ yields

$$\|(D^{-1/2}x_m - \sigma(\tilde{x}))_S\|_2 = \|(\sigma(\tilde{G}\tilde{x}) - \sigma(\tilde{x}))_S\|_2 \le \|((\tilde{G} - I)\tilde{x})_S\|_2.$$

Since $\tilde{x}$ is supported on S, we have

$$((\tilde{G} - I)\tilde{x})_S = (\tilde{G}_{SS} - I)\tilde{x}_S.$$

Moreover, $\tilde{G}$ has unit diagonal and off-diagonal entries bounded by μ, so each row of $\tilde{G}_{SS} - I$ has ℓ_1-norm at most $(k-1)\mu$, implying (e.g., by Gershgorin)

$$\|\tilde{G}_{SS} - I\|_2 \le (k-1)\mu.$$

Therefore,

$$\|((\tilde{G} - I)\tilde{x})_S\|_2 = \|(\tilde{G}_{SS} - I)\tilde{x}_S\|_2 \le \|\tilde{G}_{SS} - I\|_2 \, \|\tilde{x}_S\|_2 \le (k-1)\mu \, \|\tilde{x}\|_2.$$

Plugging this bound into equation 28 proves equation 14. For ReLU, $\sigma(t) = \max\{t, 0\}$ implies

$$\|(\tilde{x} - \sigma(\tilde{x}))_S\|_2 = \|(\tilde{x}_-)_S\|_2,$$

completing the proof. □

Proof of Lemma 2. Under the blind-mixing model $x_p = W_p x$ and the dictionary matching regime $W_m = W_p^\top$, we have

$$x_m = \sigma(W_m x_p) = \sigma(W_p^\top W_p x) = \sigma(W_m W_m^\top x) = F(x),$$

which is equation 16. Define $x_t \triangleq F(x_m)$, so that $\|F(x_m) - x_m\|_2 = \|x_t - x_m\|_2$.

Since $x, x_m \in \mathcal{U}$ and F is L-Lipschitz on $\mathcal{U}$, we have

$$\begin{aligned}\|F(x_m) - x_m\|_2 &= \|F(x_m) - F(x)\|_2 \\ &\le L\|x_m - x\|_2,\end{aligned}$$

where we used $x_m = F(x)$ in the first line and the Lipschitz property equation 18 in the second line. Rearranging yields

$$\|x_m - x\|_2 \ge \frac{\|F(x_m) - x_m\|_2}{L} = \frac{\|x_t - x_m\|_2}{L},$$

which proves equation 19. □

Proof of Lemma 3. Assume $\sigma = \mathrm{ReLU}$ and $W_m = W_p^\top$. Then

$$F(z) = \mathrm{ReLU}(W_m W_m^\top z) = \mathrm{ReLU}(W_p^\top W_p z).$$

ReLU is 1-Lipschitz, hence for any u, v,

$$\|F(u) - F(v)\|_2 = \|\mathrm{ReLU}(W_p^\top W_p u) - \mathrm{ReLU}(W_p^\top W_p v)\|_2 \le \|W_p^\top W_p(u - v)\|_2 \le \|W_p^\top W_p\|_2 \, \|u - v\|_2.$$

Therefore, F is L-Lipschitz with $L \le \|W_p^\top W_p\|_2 = \|W_p\|_2^2$, proving equation 20. Substituting this bound on L into Theorem 2 yields equation 21. □

D Experiments Details

Additional implementation details. We use two SAE settings throughout: (i) a ReLU+ℓ_1 SAE with ℓ_1 coefficient `L1_COEF`$= 10$, and (ii) a TopK SAE that enforces sparsity by construction (no ℓ_1 penalty is used in this setting). For our regularization method, we add the proposed self-consistency objective to the training loss with coefficient λ, while keeping all other hyperparameters identical to the corresponding baseline (the specific λ values used in each setting are stated in the main text and table captions). For both ReLU+ℓ_1 and TopK, we apply a linear warm-up schedule for the regularizer: we set $\lambda = 0$ for the first 600 steps, linearly increase λ from step 600 to 1200 until it reaches the target value, and keep λ fixed for the remaining steps. When a smooth proxy for activation statistics is needed, we use a temperature parameter τ=0.05; this is only used for measuring/monitoring sparsity-related quantities in the ReLU+ℓ_1 setting and does not change the CE-bench metric definitions (TopK does not require this).

Data budget and batching. Per training step, we process 16 sequences of length up to 128 tokens (BATCH_TEXT= 16, MAX_LENGTH= 128), yielding TOKENS_PER_STEP= 16,384. For CE-bench scoring we cap the number of paired samples to MAX_PAIRS= 3000. We report mean±std over multiple random seeds; seeds are matched across the baseline and our regularization method to enable paired comparisons (the number of seeds for each setting is stated in the corresponding result tables).

Dictionary health metrics. To complement CE-bench, we report three dictionary health metrics computed from the trained SAE. We sample token activations from the target LLM layer, center them by a running mean, and estimate per-unit usage rates.

Dead fraction (lower is better). Let $a_{t,i} = \mathbf{1}[x_{m;t,i} > 0]$. We define the empirical usage rate of unit i as $u_i = \mathbb{E}_t[x_{m;t,i}]$ (estimated by averaging over sampled tokens/steps), and report $\texttt{dead\_frac} = \frac{1}{d_{\text{lat}}} \sum_i \mathbf{1}[u_i < \tau_{\text{dead}}]$ with $\tau_{\text{dead}} = 10^{-4}$.

Usage entropy (higher is better). We normalize usage rates into $p_i = u_i / \sum_j u_j$ and compute the normalized Shannon entropy

$$\texttt{usage\_entropy} = \frac{-\sum_{i=1}^{d_{\text{lat}}} p_i \log(p_i + \epsilon)}{\log d_{\text{lat}}} \in [0, 1],$$

where ϵ is a small constant, we use 10^{-12} here.

Coherence RMS (lower is better). Let w_i be the i-th column of W_{dec}. After ℓ_2-normalizing columns $\tilde{w}_i = w_i / (\|w_i\|_2 + \epsilon)$, we form $G_{ij} = \langle \tilde{w}_i, \tilde{w}_j \rangle$ and report the RMS off-diagonal correlation:

$$\texttt{coh\_rms} = \sqrt{\frac{1}{d_{\text{lat}}^2} \sum_{i \neq j} G_{ij}^2}.$$

where ϵ is a small constant, we use 10^{-8} here.

Synthetic data generation. In the synthetic study, we generate data from a blind-mixing model $x_p = W_p x$ with controllable dictionary correlation ρ. We construct $W_p \in \mathbb{R}^{d \times n}$ by mixing an i.i.d. Gaussian matrix with a shared rank-one component (weighted by ρ) and then normalizing each column to unit norm, so larger ρ yields more correlated columns. Each latent code $x \in \mathbb{R}^n$ is sampled to be exactly k-sparse by choosing k coordinates uniformly without replacement and assigning nonzero amplitudes uniformly in $[0.5, 1.0]$.

WSAE baseline Cui et al. (2025) (reimplementation). We include WSAE as a representative weighting-based baseline. Since the original paper does not release code and leaves some implementation details unspecified, we implement WSAE based on the algorithmic description in the paper and match our synthetic data generation, optimization budget, and evaluation protocol. We treat this as a best-effort reimplementation: the purpose is to test whether objective reweighting alone can reliably reduce the ground-truth latent error in our controlled setting, rather than to claim a fully faithful reproduction of all experimental choices in the original work.