Stable and Steerable Sparse Autoencoders with Weight Regularization

Piotr Jedryszek [1 †] **Oliver M. Crook** [2 3]

Abstract

Sparse autoencoders (SAEs) are widely used to extract human-interpretable features from neural network activations, but their learned features can vary substantially across random seeds and training choices. To improve stability, we studied *weight regularization* by adding L1 or L2 penalties on encoder and decoder weights, and evaluate how regularization interacts with common SAE training defaults. On MNIST, we observe that L2 weight regularization produces a core of highly aligned features and, when combined with tied initialization and unit-norm decoder constraints, it dramatically increases cross-seed feature consistency. For TopK SAEs trained on language model activations (Pythia-70M-deduped), adding a small L2 weight penalty increased the fraction of features shared across three random seeds and roughly doubles steering success rates, while leaving the mean of automated interpretability scores essentially unchanged. Finally, in the regularized setting, activation steering success becomes better predicted by auto-interpretability scores, suggesting that regularization can align text-based feature explanations with functional controllability. Code including for training and analysis is available at https://github.com/oxPJ/sae-weight-regularization, trained SAE's and result data is available at https://huggingface.co/anonsaereg/SAE-REG-models and https://huggingface.co/datasets/anonsaereg/SAE-REG-results respectively.

[1]Department of Biology, University of Oxford, Oxford, UK [2]Kavli Institute for Nanoscience Discovery, University of Oxford, Oxford, UK [3]Department of Chemistry, University of Oxford, Oxford, UK. Correspondence to: Piotr Jedryszek <piotr.jedryszek@sjc.ox.ac.uk >.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

1. Introduction

Sparse autoencoders (SAEs) have become central to mechanistic interpretability, offering a potential solution to the superposition hypothesis, the idea that neural networks represent more features than dimensions by encoding them in overlapping patterns (Elhage et al., 2022). By learning an overcomplete basis with sparse coefficients, SAEs aim to recover the "true" features underlying model computations (Bricken et al., 2023; Cunningham et al., 2023). Despite their success, recent work raised concerns about SAE reliability. Chanin & Garriga-Alonso (2025) showed that SAE features are highly sensitive to the L0 sparsity hyperparameter, and feature splitting across dictionary sizes (Bricken et al., 2023) similarly demonstrates that which features are learned depends on model capacity. Paulo & Belrose (2025) demonstrated that SAEs trained with different random seeds on identical data learn substantially different features. This variability suggests **underconstrained optimization**: activation sparsity alone does not uniquely determine a solution.

This underdetermination is consistent with mixed downstream results: in a large-scale study of activation probing, Kantamneni et al. (2025) find that SAE-based probes do not provide a consistent advantage over strong baselines on raw activations across several challenging regimes, although they sometimes outperform on individual datasets. Recent work also argues that SAE objectives should prioritize *functional faithfulness* to model behavior rather than reconstruction alone; that is, by training SAEs end-to-end to preserve model outputs (Braun et al., 2024). Other related work has begun to address these challenges by adding additional structure to SAE training beyond activation sparsity. For example, Marks et al. (2024) propose *Mutual Feature Regularization*, which trains multiple SAEs in parallel and encourages them to converge to similar features, motivated by the idea that features replicated across runs are more likely to correspond to genuine input factors. Similarly, recently Martin-Linares & Ling (2025) introduced Distilled Matryoshka Sparse Autoencoders (DMSAEs), which run iterative train-and-select cycles: after each training run, features are scored by their gradient×activation contribution to next-token loss, and only the highest-attribution subset is carried forward as frozen encoder directions for the next

cycle. Where Mutual Feature Regularization encourages convergence across *parallel* SAEs, DMSAEs enforce consistency across *sequential* retraining cycles.

In classical machine learning, regularization is often used to bias underdetermined or overparameterized optimization problems toward simpler solutions by penalizing particular classes of parameters (Goodfellow et al., 2016). L1 regularization (Laplace prior) encourages weight sparsity, while L2 regularization (Gaussian prior) encourages small, smoothly distributed weights. Both improve generalization in many settings (Tibshirani, 1996; Krizhevsky et al., 2012). In an overcomplete setting, however, weight penalties can also affect the effective size of the learned dictionary by suppressing weakly-used features — a mechanism we engage with directly when interpreting our results.

In this work we revisit a simple idea: add an explicit weight penalty to SAE training, in addition to the usual activation sparsity term. We focus on three questions:

1. **Cross-seed consistency:** Does weight regularization increase the reproducibility of features across random seeds?
2. **Feature quality:** Does weight regularization affect downstream evaluations such as interpretability and steering?
3. **Interaction with SAE design choices:** How does weight regularization interact with common architectural and training decisions such as encoder/decoder initialization, decoder constraints, sparsity mechanism (TopK, BatchTopK, Matryoshka)?

We explore weight regularization first on a toy model of MNIST images to build intuitions, then on a small language model (Pythia-70M-deduped) to test real-world applicability.

2. Methods

2.1. Sparse autoencoders with weight regularization

Given an activation vector $x \in \mathbb{R}^d$, an SAE produces a sparse latent $z \in \mathbb{R}^m$ and a reconstruction $\hat{x}$:

$$z = f(W_{\text{enc}}x + b_{\text{enc}}), \tag{1}$$

$$\hat{x} = W_{\text{dec}}z + b_{\text{dec}}. \tag{2}$$

We consider standard sparsity mechanisms for z (e.g. an explicit ℓ_1 penalty or a hard TopK constraint) and add weight regularization:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{sparse}}\,\mathcal{L}_{\text{sparse}}(z) \\ + \lambda_{\text{w}}\Big(\|W_{\text{enc}}\|_p^p + \|W_{\text{dec}}\|_p^p\Big),\end{aligned} \tag{3}$$

where $p \in \{1, 2\}$ and $\mathcal{L}_{\text{recon}} = \|x - \hat{x}\|_2^2$. For TopK models, sparsity is enforced by the encoder and we set $\mathcal{L}_{\text{sparse}}(z) = 0$.

2.2. Decoder constraints and tied initialization

Two implementation choices are common in recent SAE work and are present in the default SAE implementations of SAEBench (Karvonen et al., 2025):

1. **Tied initialization:** initialize $W_{\text{dec}} \leftarrow W_{\text{enc}}^\top$ so that each latent starts as an approximately self-consistent "detector" and "writer" direction.
2. **Unit-norm decoder columns:** enforce $\|W_{\text{dec}}(:, i)\|_2 = 1$ for each decoder column via normalization during training.

A unit-norm constraint will neutralize L2 shrinkage on the decoder (since $\|W_{\text{dec}}\|_F^2$ becomes constant), so the same λ_{w} may behave differently depending on whether decoder normalization is applied. Motivated by this, we explore these choices in a toy MNIST setting and retain SAEBench defaults when moving to language-model activations. When unit-norm decoder columns are enforced, the weight penalty acts as encoder regularization.

2.3. Cross-seed feature consistency

To quantify reproducibility, we followed Paulo & Belrose (2025) and measured feature similarity across SAEs trained with different random seeds. For two trained SAEs A and B, let $D_A, D_B \in \mathbb{R}^{m\times d}$ be the decoder feature matrices. We compute the absolute cosine similarity matrix $S = |\cos(D_A, D_B)|$ and use Hungarian matching to obtain a one-to-one pairing. We report: **mean max cosine** (average over features of $\max_j S_{ij}$, symmetrized), **fraction paired** (fraction with $\max_j S_{ij} > 0.7$), and **shared features** (a stricter criterion requiring encoder and decoder Hungarian matchings to agree *and* both similarities exceed 0.7).

We call a feature *alive* if its encoder–decoder cosine similarity exceeds 0.1 using this threshold as a pragmatic filter for collapsed or weakly self-consistent latents. This excludes both features whose encoder or decoder norm has collapsed to zero during training and the small number of features with weakly aligned encoder–decoder pairs. On our trained SAEs the encoder–decoder cosine distribution is strongly bimodal (Figure 3, Appendix C), so results are insensitive to the exact threshold value.

2.4. Steering and evaluation

For language-model SAEs, we evaluated *feature steering* by injecting a decoder feature vector into the residual stream during text generation: $r_t \leftarrow r_t + \alpha\, d_i$, where d_i is de-

coder feature i and α is the steering strength. We used non-deterministic decoding (temperature $= 0.7$, top-$p = 0.9$) with generation-only steering and fixed scaling ($\alpha = 5$). Full hyperparameter details appear in Appendix A.

For each SAE we sampled 300 features uniformly at random from the alive pool (per §2.3), generated auto-interpretability explanations (LLM-generated descriptions of features, scored by how well a second LLM can use them to predict the feature's activations on example texts)for each, and retained features with valid explanations as steering candidates. Each candidate was steered against 3 prompts. Sampling was performed independently per SAE; because regularized and unregularized models are trained as separate trainers, feature indices are not comparable across SAEs and we report aggregate sample-level statistics rather than paired per-feature comparisons.

An LLM judge (GPT-5.1) scored whether the steered output was differently related to the feature's expected concept than the unsteered output on a 1–5 scale (with 3 being no difference). We counted scores ≥ 4 as successful steering.

3. Toy Experiments on MNIST

3.1. Weight regularization induces an "aligned core"

We trained SAEs on flattened MNIST images ($d = 784$) with $m = 1{,}568$ latents ($2\times$ overcomplete) and an L1 sparsity penalty on activations. With untied initialization, adding L2 regularization ($\lambda_w = 10^{-5}$) produced a bimodal distribution of encoder–decoder cosine similarities and yielded a small set of aligned features that qualitatively correspond to clean strokes and curves (Figure 1). A small number of high-cosine similarity latents accounted for most reconstruction capacity (Appendix B).

3.2. Regularization improves cross-seed consistency

To match language-model SAE practices, we repeated MNIST training with tied initialization and a unit-norm decoder constraint to match the design choices present in SAEBench (Karvonen et al., 2025). Table 1 summarizes cross-seed consistency across three random seeds.

Without decoder constraints and tied initialization, features are mostly not reproducible by the strict shared-feature criterion ($\approx 0\%$ shared). With tied initialization and unit-norm decoders, adding weight regularization substantially increases the fraction of shared features. L2 yields the highest shared-feature fraction for alive features (22.5% vs. 1.9% without regularization). We also observed that shared features appear qualitatively more interpretable than random features (Figure 2). Based on these results, we proceeded with SAEBench defaults for Pythia experiments.

4. Language Model Experiments

4.1. Setup

We evaluated weight regularization on SAEs trained on layer-3 residual stream activations from Pythia-70M-deduped using SAEBench (Karvonen et al., 2025). We tested TopK (Gao et al., 2024), BatchTopK (Bussmann et al., 2024), and Matryoshka (Bussmann et al., 2025) architectures with sweeps over sparsity ($k \in \{40, 80, 150, 320\}$) and regularization strengths ($\lambda \in \{0, 10^{-12}, 10^{-10}, 10^{-7}, 10^{-5}, 10^{-4}\}$), keeping tied initialization and unit-norm decoder columns throughout.

We evaluate on sparse probing (SP), spurious correlation removal (SCR@10), targeted probe perturbation (TPP@10), and CE loss recovery for all architectures. We then measure cross-seed feature sharedness for the TopK architecture, and proceeded with automated interpretability and steering experiments.

4.2. SAEBench performance and cosine-similarity structure

Across architectures, weight regularization frequently appears in Pareto-optimal configurations under SAEBench metrics (Appendix D). As in MNIST, TopK models with L2 regularization produce a high-cosine cluster of aligned features alongside many dead latents (Figure 3 and Appendix C). This effect is architecture-dependent: BatchTopK with L2 regularization instead shows a general shift toward lower cosine similarities without the bimodal structure. In TopK models, L2 regularization is particularly aggressive, killing off the majority of features during training (see Appendix C).

4.3. L2 regularization increases cross-seed feature sharedness

We computed cross-seed feature reproducibility across three random seeds for TopK SAEs ($k \in \{40, 80, 150, 320\}$), comparing $\lambda \in \{0, 10^{-4}\}$. Adding an L2 weight penalty produced a large and consistent improvement across all sparsity levels (Figure 4): the fraction of strictly shared features among alive features increases over tenfold (from $< 2\%$ to $\sim 35\%$), mean max cosine similarity among alive features roughly doubles (from ≤ 0.32 to ~ 0.7), and the fraction of features with cosine similarity > 0.7 in decoder increases almost fivefold (from $< 10\%$ to $\sim 50\%$).

4.4. Regularization improves steering and strengthens the auto-interp–steering link

For TopK SAEs at k=40, L2 weight regularization increased steering success under our sampling protocol. We independently sampled 300 alive features from each SAE, drawn

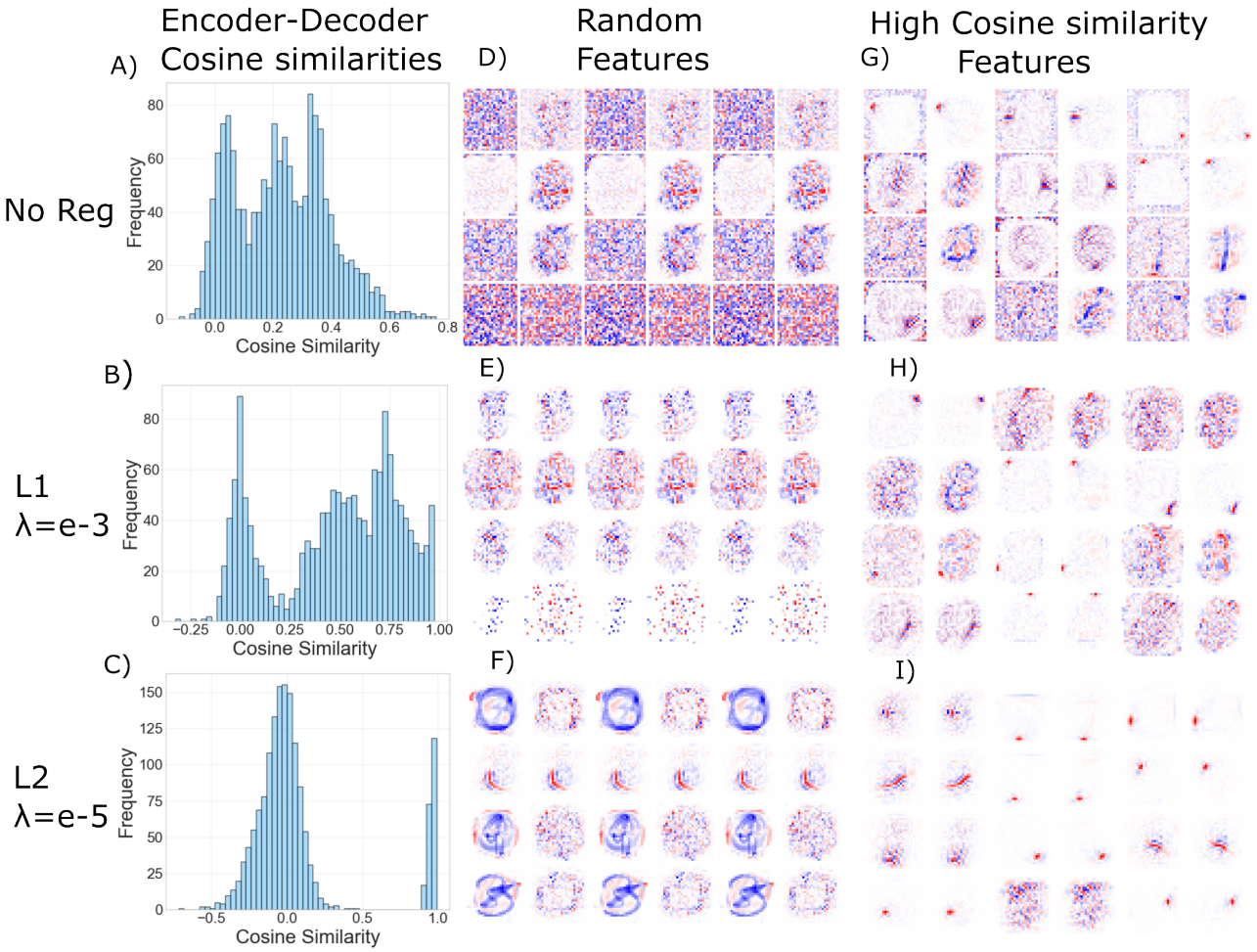

Figure 1. **MNIST cosine similarity and feature visualizations. Left column (A–C):** Histograms of encoder–decoder cosine similarity for base, L1, and L2 SAEs (1,568 latents). L2 creates a bimodal distribution with a small high-alignment core. **Right panels (D–I):** Feature pairs showing encoder and decoder as 28×28 heatmaps (blue = negative, red = positive). Left subcolumns show random samples; right subcolumns show high cosine-similarity features. L2's high-alignment features capture clean strokes and curves. The plotted features come from SAEs without decoder norm constraints.

from alive pools of 16,384 features for the unregularized model and 1,699 features for the L2-regularized model, and steered each feature against 3 prompts. All 300 unregularized features received valid auto-interpretability explanations (900 samples), compared with 185/300 L2 features (555 samples). Conditional on valid explanations, sample-level success rates (judge score ≥ 4) rose from 6.3% (57/900) to 13.0% (72/555) with L2 ($\chi^2 = 17.92$, $p < 10^{-4}$). Thus, the steering improvement should be read as conditional on the explained surviving features; the 115 invalid L2 explanations are themselves informative, indicating a weakly structured tier of alive but hard-to-explain features. Conditional on a valid explanation, the auto-interpretability score distribution is preserved across conditions (Figure 6B).

The relationship between auto-interpretability and steering success also strengthens with regularization. The Spearman correlation between auto-interpretability score and per-feature steering success is weak without regularization ($r = 0.060$, $p = 0.075$) but becomes significant with L2 regularization ($r = 0.144$, $p = 7 \times 10^{-4}$). The corresponding slope of success probability against auto-interpretability score increases from 0.11 (SE 0.06) without regularization to 0.40 (SE 0.12) with L2, a significant change ($Z = 2.21$, $p = 0.027$), indicating that auto-interpretability becomes a meaningfully stronger predictor of steering success under regularization. The distribution of auto-interpretability scores themselves remains similar across conditions (Figure 6), suggesting that regularization does not simply improve interpretability scores but instead better aligns what a feature *means* with what it *does*.

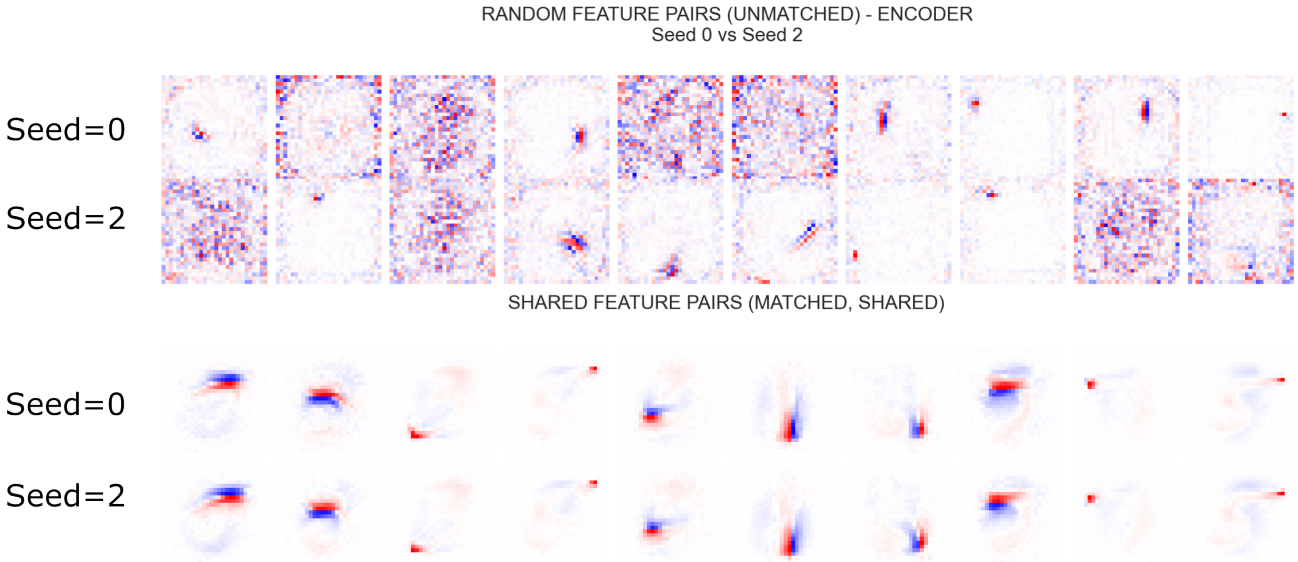

Figure 2. **MNIST shared vs. random feature visualization.** Encoder weights as 28×28 heatmaps (blue = negative, red = positive). Top two rows: random features; bottom two rows: features shared between SAEs with seed 0 and 2. Shared features capture clean strokes and curves; random features appear noisy. Features shown are from the SAE with tied weights and constrained decoder but no weight penalty.

Setting	Weight penalty	Mean max cos	Frac paired (> 0.7)	Frac shared (All)	Frac shared (Alive)
Untied init	none	0.2400	3.57%	0.00%	0.00%
Tied init (constr. Dec)	none	0.4783	48.28%	1.74%	1.74%
Untied init	L1 ($\lambda_w = 10^{-3}$)	0.3260	4.66%	3.21%	0.60%
Tied init (constr. Dec)	L1 ($\lambda_w = 10^{-3}$)	0.5486	51.62%	14.07%	14.10%
Untied init	L2 ($\lambda_w = 10^{-5}$)	0.5061	7.38%	4.91%	2.90%
Tied init (constr. Dec)	L2 ($\lambda_w = 10^{-5}$)	0.3858	42.88%	12.54%	22.50%

Table 1. MNIST cross-seed feature consistency (3 seeds) with unit-norm decoder columns. Weight regularization interacts strongly with decoder constraints and tied initialization; L2 regularization increases the fraction of strictly shared features by an order of magnitude.

Decoder orthogonality. To test whether L2 regularization encourages more orthogonal decoder columns, we computed the mean absolute pairwise cosine similarity of the decoder matrix (c_{dec}; (Chanin & Garriga-Alonso, 2025)) for each TopK configuration across 3 seeds (Figure 5). All values are robust across seeds (across-seed SD $\leq 2 \times 10^{-4}$, much smaller than the differences discussed below). For context, random unit vectors in $\mathbb{R}^{512}$ have expected mean absolute cosine $\sqrt{2/(\pi d)} \approx 0.035$. Over all features, L2 consistently lowers c_{dec}, with the largest reduction at k=40 (0.043 $\rightarrow$ 0.036). Restricting to alive features reveals a more nuanced picture: at k=40 the values are nearly identical (0.043 vs. 0.042), while at $k \geq 80$ the surviving L2 features become progressively more orthogonal than the full unregularized dictionary (e.g., 0.028 vs. 0.035 at k=320). Notably, the unregularized dictionary's alive-feature c_{dec} approaches the random-vector baseline as k increases, while L2 alive features cross *below* the baseline from $k \geq 80$ onward, indicating that regularization actively pushes the dictionary toward greater-than-random orthogonality.

5. Discussion

Adding L2 weight regularization to SAE training increases cross-seed feature reproducibility and improves steering success, but the majority of latents collapse to zero (Appendix C). Concretely, the TopK-L2 model retains over 1,500 alive features which is still 3× the Pythia-70M residual stream dimension of 512. The regularized model thus remains a standard SAE in both defining respects: sparse activations (fixed by k) and an overcomplete latent dictionary, just less aggressively so than the nominal 32× expansion. This raises a natural question: is L2 performing implicit dictionary-size selection, and would an unregularized SAE with 1,500 latents learn similar features? We suspect the answer is largely yes and that much of L2's benefit stems from finding an appropriate effective dictionary size, a choice that currently requires manual sweeps. However, L2 might shape which features survive in ways a smaller unregularized dictionary would not replicate. A direct comparison was outside the scope of this exploration and we leave it to future work. Framed this way, the collapse is not necessarily

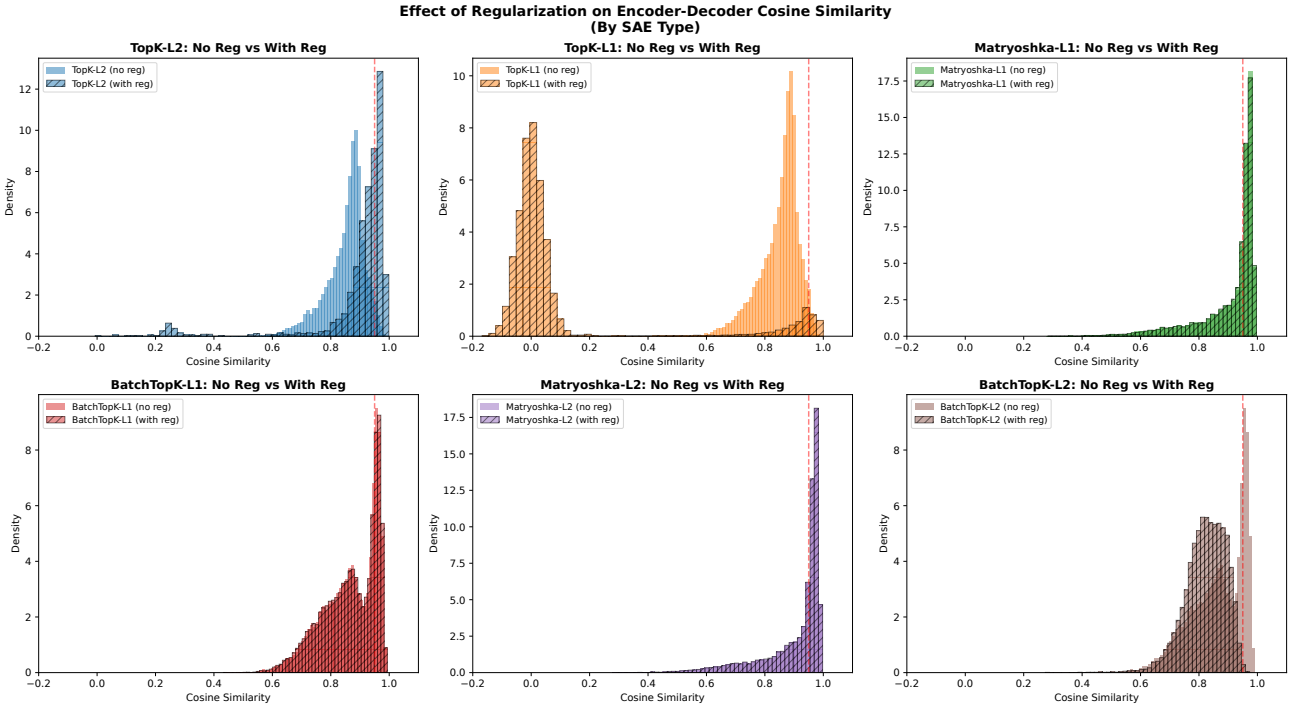

Figure 3. **Encoder–decoder cosine similarity distributions** for regularized ("reg") vs. unregularized ("no reg") SAEs across architectures.Features with zero encoder or decoder norm are excluded; for TopK-L2, the majority of features have collapsed to zero norm during training (see Appendix C). Distributions are of the Pareto-best SAE for each architecture and the unregularized SAE with corresponding k.

only a failure mode: even if L2's effect were largely capacity selection, an automatic pressure toward a smaller effective dictionary could still be useful, provided it does not remove features needed for downstream applications. Even without that comparison, two observations support the view that the surviving features are meaningful rather than arbitrary. First, the cross-seed consistency results (Figure 4) indicate that independent optimization runs converge on overlapping subsets of surviving features. Second, Martin-Linares & Ling (2025) arrive at a structurally similar outcome through an entirely different mechanism: their attribution-guided distillation converges on a core of 197 features (out of 65k) that persist across retraining cycles. That both L2 weight decay and iterative attribution-based pruning dramatically reduce the effective dictionary to a compact subset suggests that standard SAE dictionaries contain substantial redundancy, and that multiple forms of complexity control converge on similar effective sizes.

Interaction with SAE design choices. Regularization does not act in isolation. In MNIST, the combination of tied initialization, unit-norm decoder columns, and L2 yielded the highest fraction of alive features shared, an order of magnitude larger than regularization or decoder constraints alone (Table 1). The response is also architecture-dependent: TopK models with L2 produce a bimodal cosine-similarity distribution, while BatchTopK shows a general shift without bimodal structure (Figure 3). These interactions show that the effect of weight regularization is shaped by the full training recipe.

Steering and the interp–steering gap. The roughly doubled steering success rate is practically relevant. We measured steering only at k=40, where the regularised model had the highest fraction of shared features across seeds, making it the most natural test case; the discussion below is therefore directly grounded only at this sparsity level. We initially hypothesised that L2 produces more orthogonal decoder columns, and Figure 5 shows this holds only partially. At k=40, regularised and unregularised SAEs have nearly identical alive-feature orthogonality, so the steering improvement we observe is better attributed to dictionary pruning: by collapsing ∼90% of latents, L2 removes weakly-used or redundant directions that would otherwise introduce off-target interference when steering. At higher k, the surviving L2 features are more orthogonal than the full unregularised dictionary (Figure 5, right), suggesting that regularisation can also produce a more disentangled basis beyond pruning alone. Whether this orthogonality gain translates into further steering improvements at higher k – i.e. whether the mechanism behind improved steering genuinely shifts with sparsity – is an open question, since our steering experiments are limited to k=40 and the orthogonality gains are small in absolute terms. Independently, the strengthened cor-

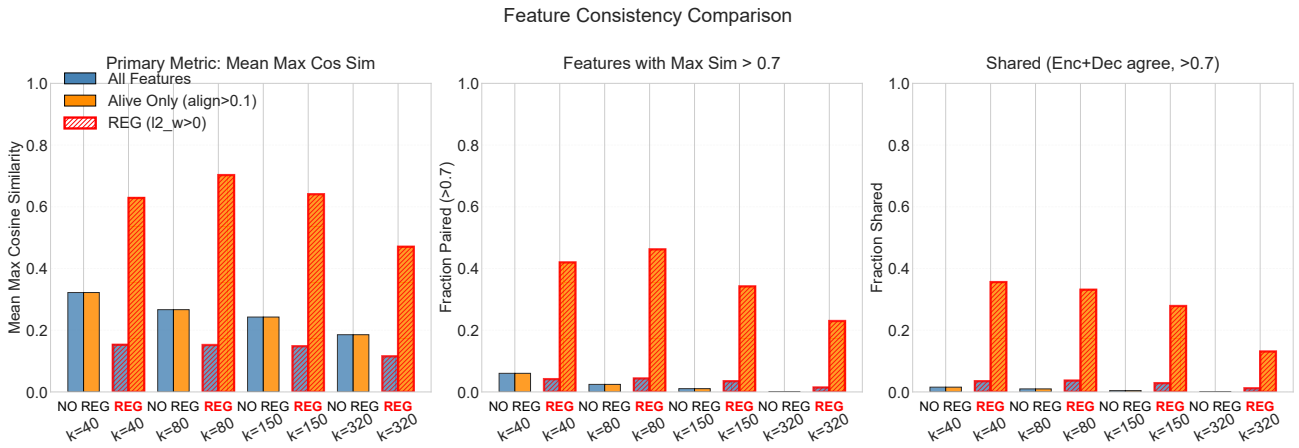

Figure 4. Pythia-70M TopK: cross-seed feature consistency metrics across sparsity levels (k) for unregularized (blue) and L2-regularized (orange) SAEs. L2 weight regularization substantially increases sharedness, particularly among alive features.

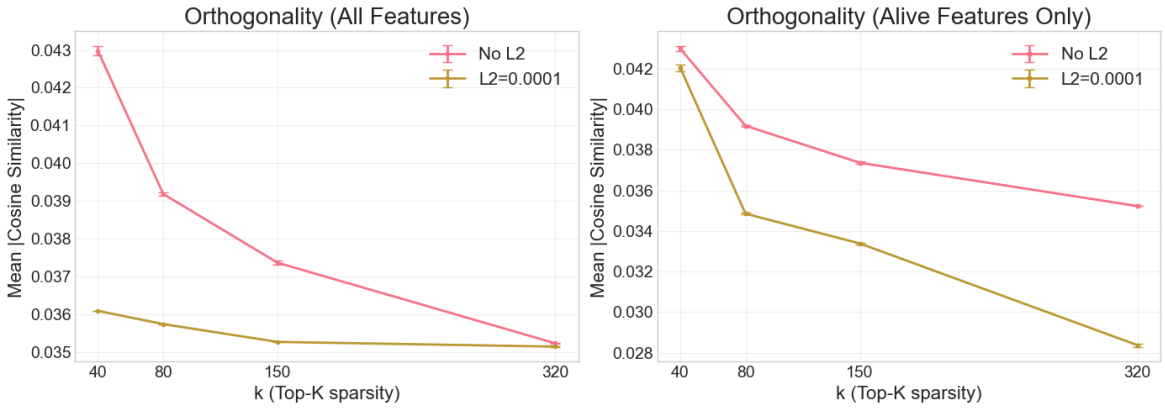

Figure 5. Decoder pairwise cosine similarity (c_{dec}) across TopK sparsity levels (mean across 3 seeds; across-seed SDs are $\leq 2 \times 10^{-4}$ and barely visible at this scale). **Left:** computed over all features; L2 reduces c_{dec}, partly driven by dead features. **Right:** computed over alive features only (encoder–decoder alignment > 0.1). At k=40, alive-feature orthogonality is nearly identical; at higher k, surviving L2 features are *more* orthogonal than the full unregularized dictionary.

relation between auto-interpretability and steering success (Spearman r: $0.060 \rightarrow 0.144$) suggests that regularisation may partially narrow the gap between what a feature *means* and what it *does*.

A complementary approach is to modify the SAE objective so that learned features are explicitly *functionally important*. Braun et al. (2024) propose end-to-end sparse dictionary learning, training an SAE to minimize divergence between original and reconstructed model outputs. Our findings show that even a simple L2 weight penalty can partially align textual explanations with controllability; combining weight regularization with end-to-end objectives might be a promising direction.

Connections to prior work. The dead-feature phenomenon can also be reinterpreted through the lens of the Minimum Description Length principle (Ayonrinde et al., 2024):L2 may eliminate features whose marginal reconstruction contribution does not justify their coding cost. Under this view, L2 regularization and attribution-guided selection (Martin-Linares & Ling, 2025) act as complementary forms of complexity control — one continuous during optimization, the other discrete between training cycles — and their convergence on compact feature sets is what an MDL perspective would predict.

Limitations and future work. All language-model experiments use Pythia-70M-deduped; scaling behavior is unknown. Cross-seed and steering analyses focus on TopK, though Figures 3 and Appendix D suggest architecture-dependent responses. Steering evaluation relies on a single LLM judge, and the high dead-feature rate limits applications requiring comprehensive coverage. Key next steps include directly testing the implicit-capacity-selection hypothesis by comparing regularized SAEs against smaller

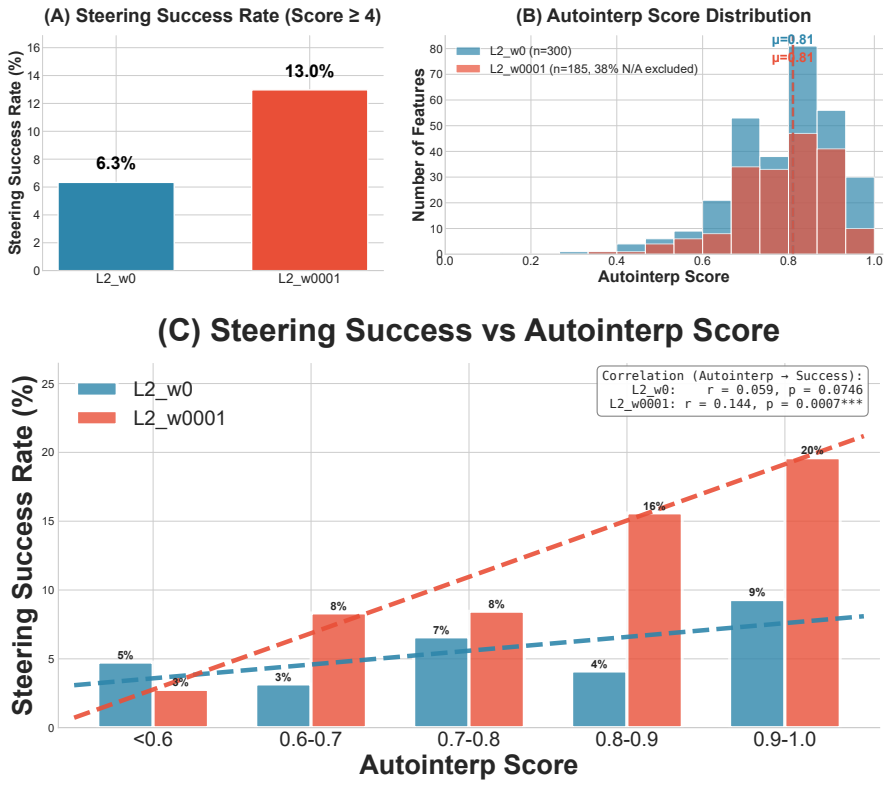

Figure 6. Pythia-70M TopK k=40: **(A)** Steering success rate (LLM judge score ≥ 4); L2 regularization roughly doubles the rate. **(B)** Auto-interpretability score distributions remain similar across conditions. **(C)** Spearman correlation between auto-interpretability and steering success; regularization strengthens the link.

unregularized dictionaries matched on alive-feature count — a comparison that would disentangle how much of L2's benefit is explained by dictionary size alone versus additional shaping of which features are learned — along with developing diagnostics that distinguish useful from pathological feature death, exploring regularization annealing schedules, and combining weight regularization with post-hoc feature selection methods such as attribution-guided distillation (Martin-Linares & Ling, 2025).

6. Conclusion

Weight regularization is a simple modification that meaningfully affects SAE behavior. In MNIST, it yields a small aligned core of clean features and under standard training constraints substantially increases cross-seed feature sharedness. In Pythia-70M TopK SAEs, L2 weight regularization increases the fraction of shared features across seeds and improves steering success among features with valid explanations, but also collapses many latents and increases the invalid-explanation rate. We therefore interpret L2 primarily as a simple form of complexity control or implicit effective-dictionary-size selection. Regularization also modestly strengthens the relationship between auto-interpretability scores and steering success, suggesting a possible pathway toward more reliable and functionally meaningful SAE features.

Impact Statement

This paper presents work whose goal is to advance the field of mechanistic interpretability. By improving the stability and reliability of sparse autoencoders, this work may contribute to better tools for understanding and monitoring neural network behavior, with potential benefits for AI safety and AI for Scientific discovery. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

We thank Peter Minary, Craig MacLean, and Adam Winnifrith for helpful feedback and discussion. PJ is supported by the Biotechnology and Biological Sciences Research Council (UKRI-BBSRC) grant BB/T008784/1. OMC acknowledges funding from a New College Todd-Bird Junior Research Fellowship and MRC Fellowship MR/Y010078/1.

References

Ayonrinde, K., Pearce, M. T., and Sharkey, L. Interpretability as Compression: Reconsidering SAE Explanations of Neural Activations with MDL-SAEs. 10 2024. URL http://arxiv.org/abs/2410.11179.

Braun, D., Taylor, J., Goldowsky-Dill, N., and Sharkey, L. Identifying Functionally Important Features with End-to-End Sparse Dictionary Learning. 5 2024. URL http://arxiv.org/abs/2405.12241.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N. L., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bussmann, B., Leask, P., and Nanda, N. BatchTopK Sparse Autoencoders. In *Science of Deep Learning Workshop at the 38th Conference on Neural Information Processing Systems (NeurIPS 2024).*, 12 2024. URL http://arxiv.org/abs/2412.06410.

Bussmann, B., Nabeshima, N., Karvonen, A., and Nanda, N. Learning Multi-Level Features with Matryoshka Sparse Autoencoders. 3 2025. URL http://arxiv.org/abs/2503.17547.

Chanin, D. and Garriga-Alonso, A. Sparse but Wrong: Incorrect L0 Leads to Incorrect Features in Sparse Autoencoders. In *Mechanistic Interpretability Workshop at NeurIPS*, 2025. URL https://github.com/chanind/sparse-but-wrong-paper.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse Autoencoders Find Highly Interpretable Features in Language Models. 10 2023. URL http://arxiv.org/abs/2309.08600.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy Models of Superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. 6 2024. URL http://arxiv.org/abs/2406.04093.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Kantamneni, S., Engels, J., Rajamanoharan, S., Tegmark, M., and Nanda, N. Are Sparse Autoencoders Useful? A Case Study in Sparse Probing. 2 2025. URL http://arxiv.org/abs/2502.16681.

Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J., Chanin, D., Lau, Y.-T., Farrell, E., McDougall, C., Ayonrinde, K., Till, D., Wearden, M., Conmy, A., Marks, S., and Nanda, N. SAEBench: A Comprehensive Benchmark for Sparse Autoencoders in Language Model Interpretability. In *Proceedings of the 42nd International Conference on Machine Learning*, 6 2025. URL http://arxiv.org/abs/2503.09532.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012. URL http://code.google.com/p/cuda-convnet/.

Marks, L., Paren, A., Krueger, D., and Barez, F. Enhancing Neural Network Interpretability with Feature-Aligned Sparse Autoencoders. 11 2024. URL http://arxiv.org/abs/2411.01220.

Martin-Linares, C. P. and Ling, J. P. Attribution-Guided Distillation of Matryoshka Sparse Autoencoders. 12 2025. URL http://arxiv.org/abs/2512.24975.

Paulo, G. and Belrose, N. Sparse Autoencoders Trained on the Same Data Learn Different Features. In *39th Conference on Neural Information Processing Systems (NeurIPS 2025) Mechanistic Interpretability Workshop*, 2025. URL https://arxiv.org/abs/2501.16615.

Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *J. R. Statist. Soc. B*, 58(1):267–288, 1996. URL https://academic.oup.com/jrsssb/article/58/1/267/7027929.

A. Steering Hyperparameters

We ran a grid search over steering hyperparameters including: (i) deterministic vs. sampling decoding, (ii) applying steering during generation only vs. all forward passes, and (iii) different strength parameterizations (fixed, residual-RMS scaled, target pre-activation delta). Across both regularized and unregularized TopK models, residual-RMS scaling gave the best average LLM-judge score, while fixed-strength steering often gave higher "high-change" rates (fraction of samples with judge score ≥ 4). The configuration used for the main analyses was non-deterministic decoding (temperature $= 0.7$, top-$p = 0.9$) with generation-only steering and fixed scaling ($\alpha = 5$), as it gave the best results for both regularized and unregularized SAEs.

B. MNIST Reconstruction MSE Ablation

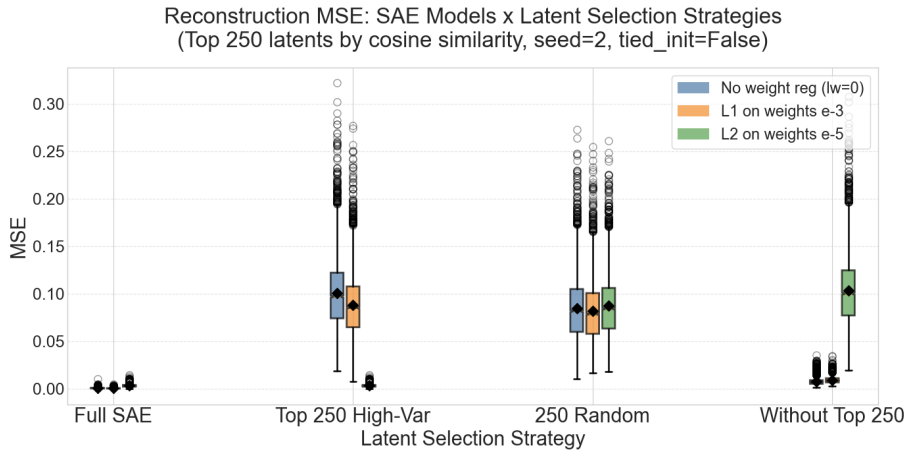

Figure 7. **MNIST reconstruction MSE.** Boxplots measured on the first 250 MNIST examples for three SAE variants (base, L1, L2) under four latent selection strategies: full SAE (all latents active), top-250 high-cosine-similarity latents, 250 random latents, and all latents except the top-250. A small number of high-cosine similarity latents accounts for most reconstruction capacity in L2-regularized SAEs.

C. TopK-L2 Cosine Similarity Including Dead Features

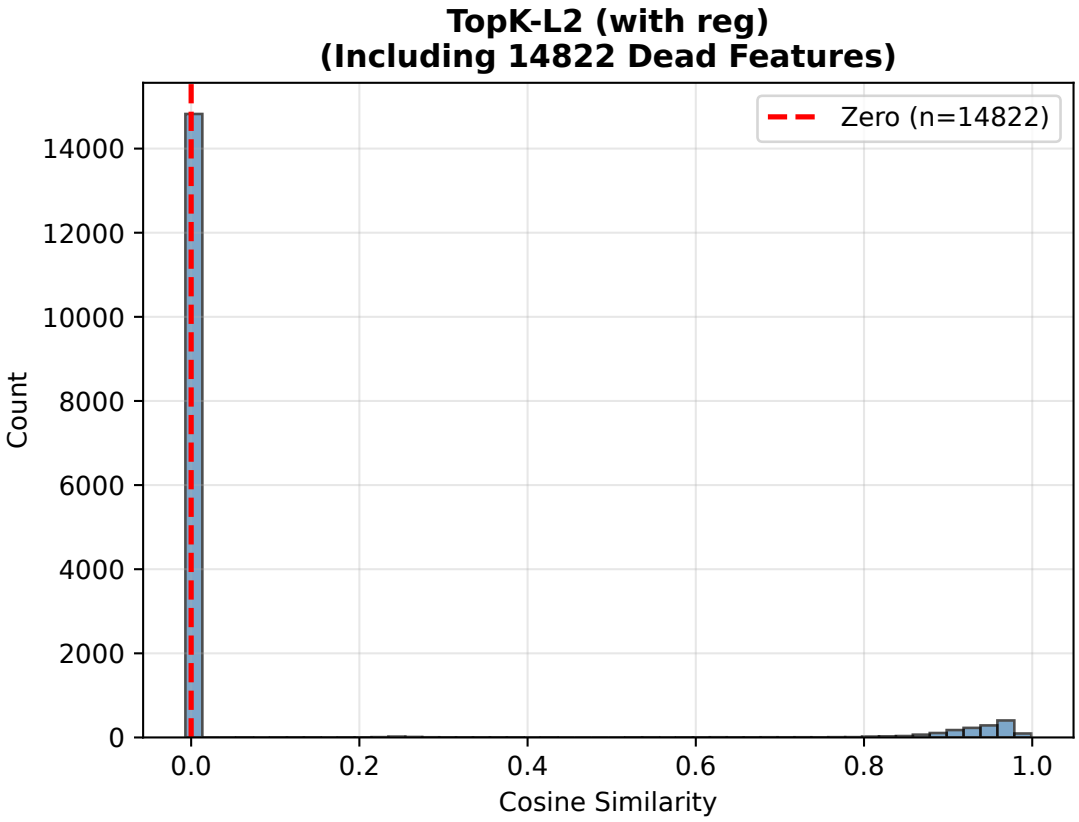

Figure 8. **TopK cosine similarity including dead features.** Encoder–decoder cosine similarity distributions for regularized vs. unregularized TopK SAEs, retaining features where the encoder or decoder norm is zero. The majority of L2-regularized features collapse to zero norm during training.

D. SAEBench Pareto Analysis

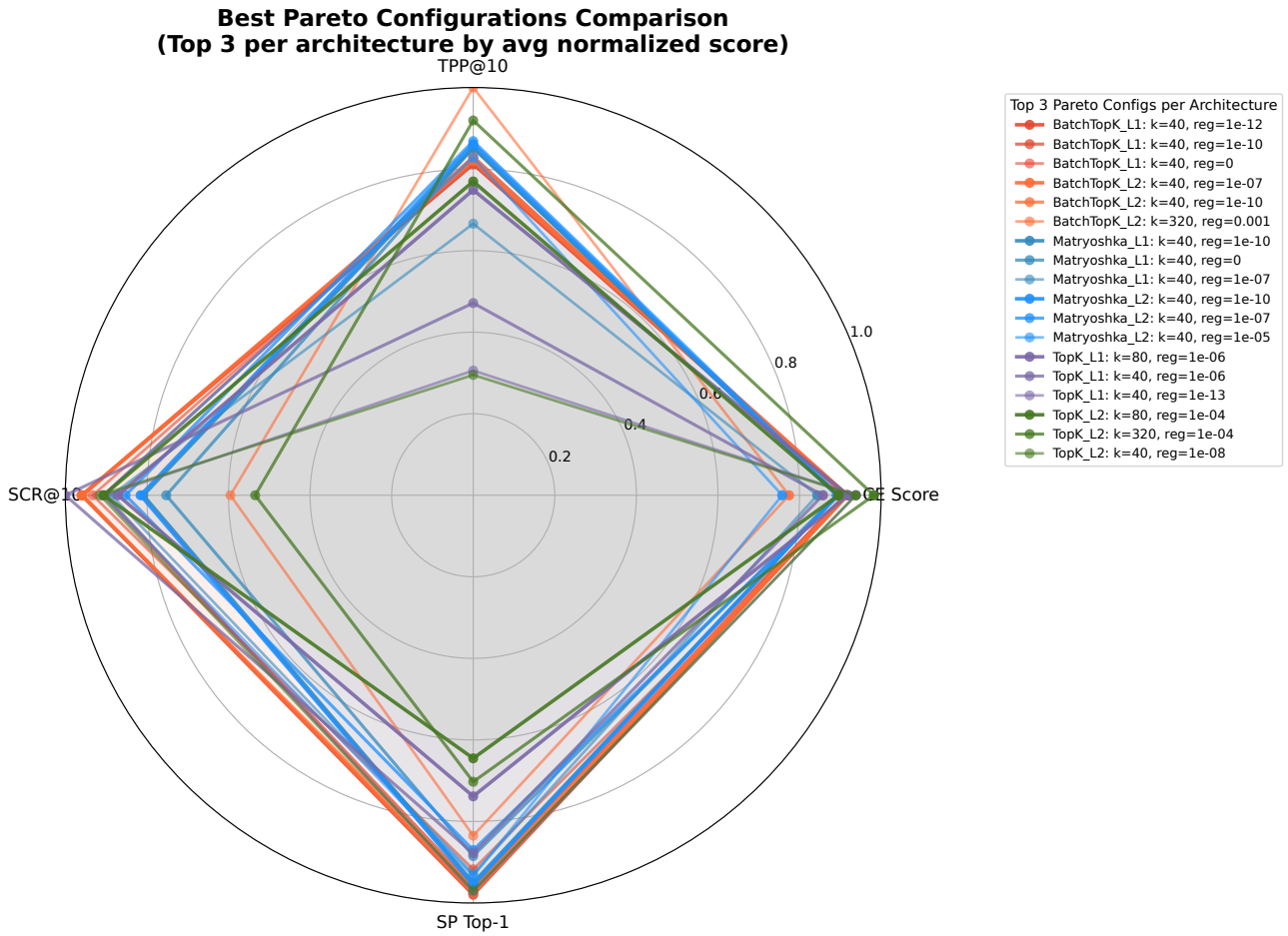

Figure 9. **SAEBench Pareto analysis.** Aggregated top-3 Pareto configurations per architecture. Regularized models frequently appear on the Pareto frontier. Radar plots showing normalized SAEBench metrics. Red = Pareto-optimal; gray = dominated. Axes: TPP@10, CE score, SCR@10, SP Top-1 (higher = better).