Weight Decay Shapes Representation Geometry: Towards a More Nuanced Understanding of Sparse Autoencoders in Vision Transformers

Anonymous Authors[1]

Abstract

Mechanistic interpretability typically treats trained models as fixed objects, yet prior work shows that training fundamentally shapes representation geometry. We ask whether this geometry determines when sparse interpretability methods succeed versus fail. Training 64 ViT-Tiny models across varied hyperparameters on traffic sign datasets, we find that weight decay is the dominant factor shaping Sparse Autoencoder (SAE) behavior. Across the sweep, higher monosemanticity and fewer dead SAE features correlate with better cross-entropy recovery in deep layers. A matched weight-decay sweep reveals a sharp threshold near wd < 0.01. Below it, SAE feature usage collapses into repeated reuse of the same small set; above it, diverse features emerge. This suggest that representation geometry, controlled by training choices like weight decay, determines whether sparse methods recover meaningful structure. Training should therefore be treated as part of the interpretability pipeline.

1. Introduction

Mechanistic interpretability typically analyzes trained models post hoc as fixed objects, largely ignoring how initial training shapes internal structure. Yet prior work shows that optimization and regularization fundamentally alter representation geometry (Kobayashi et al., 2024; Li et al., 2025). This creates a blind spot: if training determines geometric structure, and geometry constrains which interpretability methods succeed, then analyzing models without their training context may misdiagnose when and why interpretability fails. We ask whether training-induced geometric changes determine when sparse autoencoders recover interpretable features versus collapse into degenerate solutions.

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

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

Sparse Autoencoders (SAEs) decompose activations into sparse dictionaries and are widely used to probe representations (Cunningham et al., 2023; Joseph et al., 2025). Though controversial (Kantamneni et al., 2025), they provide a direct test: can the learned representation support sparse decomposition? Prior work has studied how SAE architecture affects reconstruction within fixed base models (Gao et al., 2024), but no systematic study examines how base model training shapes whether SAEs recover diverse features or fail entirely. If representation geometry is unsuitable for sparse decomposition, even well-designed SAEs may produce unstable results.

We train 64 Vision Transformer (ViT)-Tiny models on traffic sign classification (GTSRB (Houben et al., 2013), ZOD (Alibeigi et al., 2023)) in a full factorial sweep over learning rate, batch size, patch size, dropout, weight decay, and label smoothing. Traffic signs are a controlled, safety-relevant domain with distinct classes. For each ViT, we train SAEs on residual-stream activations and evaluate cross-entropy recovery, monosemanticity, and dead features on a holdout split.

Our results show that weight decay dominates both representation geometry and SAE quality, doubling effective rank and halving condition number across layers while leaving accuracy unchanged (4.2). In deep layers, higher monosemanticity and fewer dead features strongly predict better CE recovery, whereas dropout and label smoothing have no comparable effect (4.1). A matched weight-decay sweep reveals a collapse threshold near $wd < 0.01$: below it, SAEs reuse a small, fixed feature set across images; above it, hundreds to thousands of distinct features activate (4.3). Task complexity further degrades SAE reconstruction even with fixed training configuration (4.4).

Contributions

1. Weight decay is the dominant hyperparameter driving both representation geometry and SAE behavior in ViTs.
2. Higher monosemanticity and fewer dead dictionary features of the SAE are associated with better representation reconstruction.
3. There is sharp collapse threshold in weight decay below which SAE feature usage degenerates into repeated reuse of the same small feature set.

2. Background and Related Work

SAEs and Monosemanticity – SAEs learn an overcomplete sparse dictionary over model activations and have been used to recover monosemantic features and support circuit-style analyses in language models (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024). In this context, we use *feature* to denote the units a network would ideally assign to individual neurons if neuron count were not a limiting factor (Bricken et al., 2023; Bereska & Gavves, 2024). A feature is *monosemantic* if it consistently aligns with a single interpretable concept, although in practice neurons often mix correlated attributes like texture and background (Elhage et al., 2022). In vision(-language) models, SAE reconstructions can improve loss when inserted back into the forward pass (Joseph et al., 2025). Recent work proposes CLIP-based and guided metrics to quantify monosemanticity (Pach et al., 2025; Härle et al., 2025). At the same time, several studies show that SAE behavior depends strongly on width, sparsity and training details and raise questions about when SAEs are a reliable interpretability tool (Kantamneni et al., 2025; Gao et al., 2024; Fereidouni et al., 2026). These works largely keep the base model fixed and optimize the autoencoder, rather than asking when the *underlying representation* geometry itself supports a diverse, monosemantic sparse decomposition.

Vision Transformers – ViTs process images as patch sequences and build representations through self-attention (Dosovitskiy et al., 2021). Compared with language models, they operate on continuous image patches and use a learnable CLS token to aggregate information across the image (Joseph et al., 2025). Mechanistic studies indicate that deeper layers organize increasingly semantic structure, from colors and textures to class-level concepts (Dorszewski et al., 2025). Existing ViT-based SAE work mainly probes which concepts appear in a single trained model or how SAE reconstructions affect loss (Joseph et al., 2025; Pach et al., 2025). In contrast, we treat ViT training hyperparameters as part of the interpretability problem and sweep them to study when standard SAEs recover many distinct features versus collapsing to a small reused set.

Representation Geometry – Prior work links representation geometry to network structure and shows that optimization and regularization shape learned representations (Golechha & Dao, 2024; Kobayashi et al., 2024; Li et al., 2025). Recent studies further argue that geometry mediates the relationship between training choices and downstream transformer behavior (Kulkarni et al., 2026; Adam et al., 2026). In parallel, SAE-focused studies examine how autoencoder width, sparsity and objectives affect reconstruction, dead units and feature recovery within a fixed base model (Gao et al., 2024; Kantamneni et al., 2025). We bring these lines together by asking how training-induced geometric changes in ViT residual-stream representations—in particular those driven by weight decay—alter the conditions under which SAEs can recover diverse, monosemantic features rather than collapsing into degenerate sparse decompositions.

Task Design and Interpretability – Most mechanistic interpretability studies use a small set of benchmark tasks and treat dataset and label design as background choices (Bereska & Gavves, 2024; Golechha & Dao, 2024). Recent discussions suggest that task complexity and label structure can influence how disentangled representations become, and thus how amenable they are to geometric probes and feature-level analysis (Bowcott, 2025). Yet there is little empirical work that varies task complexity while holding architecture and training settings fixed to measure its effect on SAE quality. Our GTSRB experiments address this gap by increasing the number of classes at constant per-class sample counts and showing that task complexity is an additional axis that degrades SAE reconstruction.

3. Experimental Setup

Dataset – We study traffic-sign classification using GTSRB (Houben et al., 2013) and ZOD crops (Alibeigi et al., 2023) because these datasets provide a controlled, safety-relevant vision setting with visually distinct classes without the confounds introduced by highly diverse visual distributions such as ImageNet (Deng et al., 2009).

ViT Training Sweep Design – We train a $2^6 = 64$ ViT-Tiny factorial sweep on ZOD, varying learning rate, batch size, patch size, dropout, weight decay, and label smoothing (see Appendix Appendix B). We then train a matched six-model weight-decay sweep using the best configuration, varying only weight decay over $\{0.0, 0.001, 0.005, 0.01, 0.05, 0.1\}$ to isolate its effect. On GTSRB, we run a controlled task-complexity experiment, training on balanced subsets with class counts $C \in \{2, \ldots, 8\}$ and per-class sample counts $S \in \{200, \ldots, 1200\}$.

SAE Training and Layer Selection – We use ViT-Prisma (Joseph et al., 2025; Joseph, 2023) to train SAEs on residual-stream activations with ReLU, expansion factor 20, and $L1 = 10^{-4}$ (Appendix C). We train SAEs on layers $l \in \{3, 5, 7, 8, 9, 10, 11\}$ and focus on layer 10, where semantic structure is strongest while still preserving spatial detail needed to study SAE geometry (Dorszewski et al., 2025; Joseph et al., 2025).

Evaluation Metrics – SAE quality is evaluated using cross-entropy (CE) recovery (Joseph, 2023), which measures how well SAE reconstructions preserve the original model computation. We additionally measure dead features and monosemanticity (MS). Following Pach et al. (2025), we compute MS using CLIP image embeddings as the semantic

reference space. A feature is considered dead if it never activates on the evaluation set (C.3).

4. Results

4.1. Geometry Predicts SAE Performance

Across the 64-model sweep, SAE dictionary quality is most strongly associated with representation quality in the later ViT layers (Appendix D). For monosemanticity, the correlation with CE recovery is significant in layers 9-11, with strong effects in layer 10 ($\tau = 0.2877, p = 0.000779$) and layer 11 ($\tau = 0.3333, p = 0.000099$). For dead features, the association with CE recovery is significant from layer 7 onward and strengthens in the deepest layers, reaching $\tau = -0.3964, p = 1.04 \times 10^{-5}$ in layer 10 and $\tau = -0.7445, p = 6.32 \times 10^{-17}$ in layer 11. This pattern is consistent with prior work showing that deeper ViT layers contain more semantic structure (Dorszewski et al., 2025). Since MS is computed using CLIP embedding alignment, these findings suggest that CLIP-based semantics become increasingly informative in later layers. At the same time, the strong negative association between dead-feature count and CE recovery indicates that feature utilization is closely tied to reconstruction quality in deep layers.

Given that dead features in late layers degrade CE recovery, we next examine which training hyperparameters influence dead-feature formation. Among all hyperparameters in the sweep, only weight decay produces a significant change in dead-feature count across all analyzed layers (Appendix E). In contrast, learning rate affects dead-feature count only in the earliest layers and has no detectable effect from layers 7-11, where semantic structure and SAE quality are most tightly coupled. This distinction suggests that weight decay is the primary controllable hyperparameter shaping dead-feature formation and, consequently, SAE reconstruction quality in the layers most relevant for interpretability.

Finally, late-layer representation geometry and training-time regularization jointly determine SAE quality. Dead features and monosemanticity in the deepest layers predict how well SAEs reconstruct the original computation, and weight decay is the only hyperparameter in our sweep that reliably shifts these quantities. In this sense, weight decay controls the geometry that SAEs see, linking a simple training choice to the success or failure of sparse interpretability in ViTs.

4.2. Weight Decay Reshapes Representation Geometry

Since weight decay is the clearest factor affecting SAE quality in the full sweep, we next examine the geometry of the underlying ViT activations to determine whether these SAE differences reflect changes in the base representation space. Figure 1 shows effective rank across all 12 ViT layers for the wd $= 0.0$ and wd $= 0.1$ groups. Effective rank

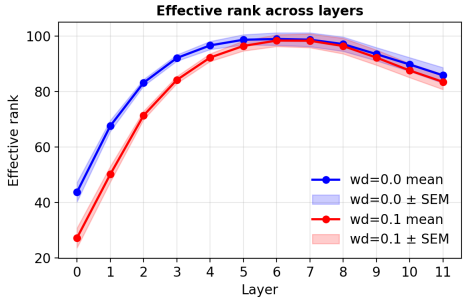

Figure 1. Effective rank across ViT layers for wd $= 0.0$ and wd $= 0.1$ (mean $\pm$ standard error of the mean (SEM); SEM $= \sigma/\sqrt{n}$).

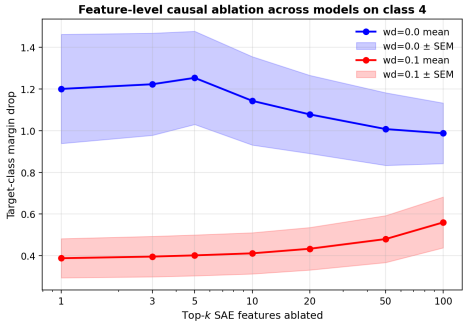

Figure 2. Top-k SAE feature ablation in layer 10 for an exemplary ZOD class.

(Appendix F) measures how many directions contribute substantially to the representation. Models trained with weight decay exhibit consistently lower effective rank across layers, with the largest differences appearing in early and middle layers, indicating that representations are concentrated in fewer dominant directions.

To test whether these geometric differences affect feature importance, we perform feature-level ablation on an arbitrarily chosen class in layer 10 by iteratively removing the top-k SAE features ranked by attribution score (Figure 5). Models trained with wd $= 0.1$ exhibit substantially steeper degradation under ablation than wd $= 0.0$ models. Appendix G shows that this pattern is consistent across classes, i.e., predictive information is concentrated in fewer dominant features for wd $= 0.0$.

Weight decay therefore does not merely reduce effective rank, but restructures representations so that a compact set of SAE features carries most task-relevant signal in later layers. We hypothesize that geometric changes induced earlier in the network matter most once representations acquire stronger semantic structure in deeper layers. Together with C4.1, this supports a geometry-based view of SAE quality, where weight-decay-induced changes in representation geometry shape dead-feature formation and feature importance, and thereby determine how well SAEs reconstruct the original ViT computation.

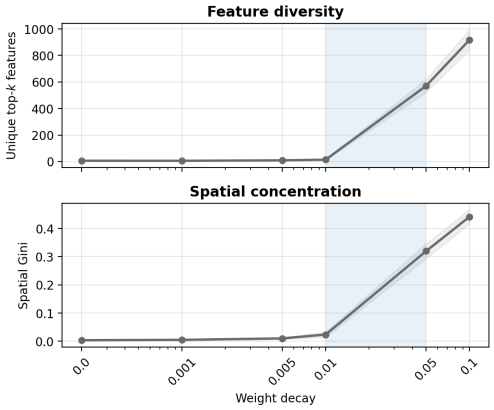

Figure 3. Feature diversity and spatial concentration across the matched weight-decay sweep across all layers with mean ± SEM.

4.3. Weight-Decay-Induced Feature Collapse

Figure 3 shows that below approximately wd ≈ 0.01, top-k SAE features are reused almost entirely across images, with the number of unique features remaining close to k. At wd $= 0.0$, feature diversity remains between 5 and 9 unique features across layers for $k = 5$ across all 1700 holdout images. Above this threshold, feature diversity increases sharply, reaching hundreds to more than one thousand unique features depending on the layer at wd $= 0.1$ (Appendix H). Spatial concentration follows the same transition, with Gini values remaining near zero below the threshold and increasing substantially above it (Appendix I).

These patterns indicate a collapse threshold in weight decay. Below wd ≈ 0.01, SAEs repeatedly reuse a small, nearly fixed feature set across images, whereas slightly above this value a large and diverse set of features activates. This supports a unified picture in which weight-decay-induced geometric changes can push the system across a regime boundary where sparse decomposition fails, because weight decay below a given threshold yields ViT representations that are geometrically unsuited for stable sparse interpretability, even with similar classification performance.

4.4. Task Complexity Affects SAE Quality

Using GTSRB, we fix the hyperparameters of the best-performing model from our 64-model-sweep and train models with increasing numbers of classes $C \in \{2, 3, \ldots, 8\}$, holding per-class sample count uniform across $S \in \{200, \ldots, 1200\}$. Here, only the training data changes, so task complexity varies while the model and optimization remain fixed. We therefore use average reconstruction CE, as CE recovery becomes noisy when both reconstruction error and baseline CE grow with class count.

A Kendall's τ test between C and average reconstruction CE is positive and significant at five of six sample-size thresholds ($\tau = 0.81$–0.90, $p < 0.03$), with only the 600-sample setting failing to reject H_0 (subsection I.1). More classes consistently produce worse SAE reconstruction regardless of how many samples per class are available.

Task complexity is a factor that degrades SAE quality (as training-geometry effects from weight decay). If the number of classes increases, the representation must support more decision boundaries, making it harder for SAEs to recover a sparse, well-decomposable structure even with fixed optimization settings. This reinforces that both training regime and task design shape whether a model's learned geometry supports stable sparse interpretability.

5. Discussion and Conclusion

Strenghts – Training-induced representation geometry, largely shaped by weight decay, determines whether SAEs in ViTs recover diverse features or collapse to repeated reuse. Across the ViT-Tiny sweep, monosemanticity and dead-feature counts in deep layers are strongly associated with CE recovery, linking geometry-driven feature utilization to reconstruction quality. Weight decay reorganizes residual-stream geometry, lowering effective rank and changing which SAE features carry predictive signal under ablation. This directly influences sparse decomposition even when model performance across the two groups of weight decay remains similar. A matched sweep across six weight-decay values further reveals a sharp threshold effect, where below approximately wd ≈ 0.01 SAEs repeatedly reuse a tiny feature set with little spatial structure. Slightly higher decay yields many distinct and spatially localized features. Together, these findings suggest that small regularization changes via weight decay can move models between representation regimes that differ substantially in how well they support stable sparse interpretability.

Limitations – We focus on ViT-Tiny models and traffic-sign datasets. While trends are consistent across GTSRB and ZOD, their generalization to larger architectures and broader datasets remains unclear. Monosemanticity in vision is inherently difficult to define and evaluate, meaning that our conclusions depend on the chosen CLIP-based metric. Our analysis is restricted to SAE behavior at a single hookpoint and relies heavily on correlational geometric measures rather than direct causal analysis of representation geometry. Hence, other approaches like model diffing might be as effective as SAEs integrated to training pipeline.

Implication – For safety-critical settings, our experiments show that geometric collapse can occur despite high accuracy. This makes geometry-aware training essential within the interpretability pipeline. Future work could train with geometric diagnostics to identify favorable regimes and co-design objectives that promote interpretable representations.

Impact Statement

This work contributes to understanding how training choices affect the structure of neural representations and the behavior of interpretability methods. There are no immediate societal risks associated with this work.

References

Adam, M., Furman, Z., and Hoogland, J. The loss kernel: A geometric probe for deep learning interpretability. OpenReview preprint, 2026. URL https://openreview.net/forum?id=c8p1zuTA33.

Alibeigi, M., Ljungbergh, W., Tonderski, A., Hess, G., Lilja, A., Lindström, C., Motorniuk, D., Fu, J., Widahl, J., and Petersson, C. Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20178–20188, 2023.

Bereska, L. and Gavves, E. Mechanistic interpretability for ai safety – a review, 2024. URL https://arxiv.org/abs/2404.14082.

Bowcott, R. Explainability, mechanistic interpretability and feature geometry. Manuscript, 2025. URL https://redmondbowcott.github.io/research/emifg.pdf.

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

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Dorszewski, T., Tětková, L., Jenssen, R., Hansen, L. K., and Wickstrøm, K. K. From colors to classes: Emergence of concepts in vision transformers, 2025. URL https://arxiv.org/abs/2503.24071.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

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, 2022. URL https://arxiv.org/abs/2209.10652.

Fereidouni, M., Haider, M. U., Ju, P., and Siddique, A. Evaluating sparse autoencoders for monosemantic representation. In Demberg, V., Inui, K., and Marquez, L. (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 5969–5984, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.313. URL https://aclanthology.org/2026.findings-eacl.313/.

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, 2024. URL https://arxiv.org/abs/2406.04093.

Golechha, S. and Dao, J. Challenges in mechanistically interpreting model representations, 2024. URL https://arxiv.org/abs/2402.03855.

Härle, R., Friedrich, F., Brack, M., and Bähr, B. Measuring and guiding monosemanticity. NeurIPS 2025 Poster, 2025. URL https://neurips.cc/virtual/2025/poster/118039.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

Joseph, S. Vit prisma: A mechanistic interpretability library for vision transformers. https://github.com/soniajoseph/vit-prisma, 2023.

Joseph, S., Suresh, P., Hufe, L., Stevinson, E., Graham, R., Vadi, Y., Bzdok, D., Lapuschkin, S., Sharkey, L., and Richards, B. A. Prisma: An open source toolkit for mechanistic interpretability in vision and video, 2025. URL https://arxiv.org/abs/2504.19475.

Kantamneni, S., Engels, J., Rajamanoharan, S., Tegmark, M., and Nanda, N. Are sparse autoencoders useful? a case study in sparse probing, 2025. URL https://arxiv.org/abs/2502.16681.

Kobayashi, S., Akram, Y., and Oswald, J. V. Weight decay induces low-rank attention layers, 2024. URL https://arxiv.org/abs/2410.23819.

Kulkarni, A., Springer, J. M., Subramonian, A., and Swayamdipta, S. Disentangling geometry, performance, and training in language models, 2026. URL https://arxiv.org/abs/2602.20433.

Li, M. Z., Agrawal, K. K., Ghosh, A., Teru, K. K., Santoro, A., Lajoie, G., and Richards, B. A. Tracing the representation geometry of language models from pretraining to post-training, 2025. URL https://arxiv.org/abs/2509.23024.

Pach, M., Karthik, S., Bouniot, Q., Belongie, S., and Akata, Z. Sparse autoencoders learn monosemantic features in vision-language models, 2025. URL https://arxiv.org/abs/2504.02821.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Tamkin, A., Durmus, E., Hume, T., Mosconi, F., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. https://transformer-circuits.pub/2024/scaling-monosemanticity/, 2024. Transformer Circuits Thread.

A. Visual Example Monosemantic vs. Polysemantic Feature

To illustrate the difference between monosemantic and polysemantic features, we inspect the images from the ViT validation set that elicit the strongest activations for each neuron in the SAE dictionary layer in Figure 4. For each selected feature, we rank all validation images by activation strength and visualize the top activating examples. This lets us qualitatively compare features whose top examples are visually coherent and correspond to a single interpretable pattern, versus features whose top examples span multiple unrelated concepts. As mentioned in section 2, it remains unclear for which part(s) or patch(es) of the image the neuron fires strongly.

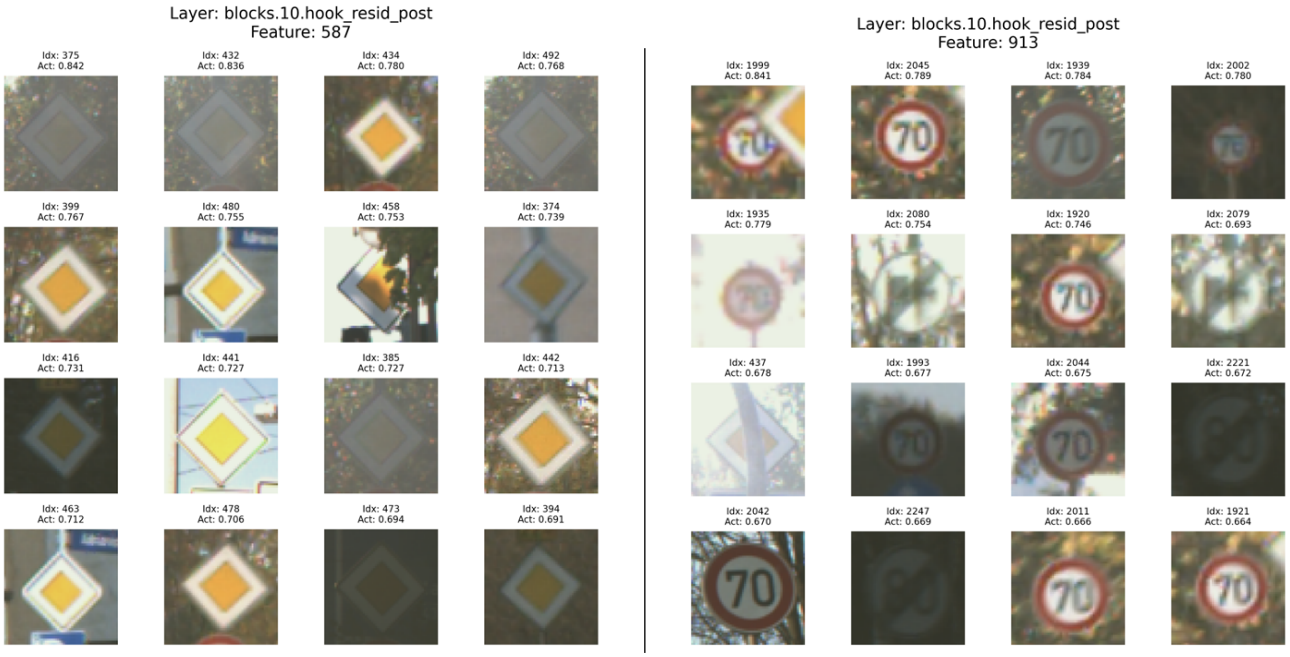

Figure 4. Top activating validation images for a visually monosemantic (left) and a polysemantic (right) SAE feature. For each feature, we rank ViT validation images by activation strength and visualize the highest-activating examples.

B. ViT Training

B.1. Dataset

Experiments were conducted on a subset of the Zenseact Open Dataset (ZOD) (Alibeigi et al., 2023). We therefore cropped traffic signs out of the images. The resulting crops were resized to 56×56 pixels and classified into 17 traffic sign categories. Separate dataset splits were used for ViT training and validation, as well as SAE training and validation. In addition, we performed out evaluations on a heldout datatset neither used in ViT or SAE training.

For ViT training, data augmentation included random resized cropping, horizontal flipping, color jitter, RandAugment, Mixup, CutMix, and random erasing. SAE training used the same normalization and augmentation pipeline on a dedicated SAE train/test split.

B.2. ViT Training Setup

All models are ViT-Tiny classifiers implemented using *HookedViT* as implemented in Vision-Prisma (Joseph, 2023). Each model contains 12 transformer layers, indexed by $l \in \{0, \ldots, 11\}$. The hidden dimension is $d_{\text{model}} = 192$, with 3 attention heads of dimension $d_{\text{head}} = 64$. The MLP hidden dimension is $d_{\text{mlp}} = 768$. Models use a CLS token for classification, and the output head predicts one of 17 traffic sign classes.

We trained a full hyperparameter grid over batch size, learning rate, label smoothing, dropout, weight decay, and patch size. The grid was:

Table 1. Hyperparameter search space used for ViT training.

Hyperparameter	Values
Batch size	$\{256, 512\}$
Learning rate	$\{10^{-4}, 10^{-3}\}$
Label smoothing	$\{0.0, 0.2\}$
Dropout	$\{0.0, 0.2\}$
Weight decay	$\{0.0, 0.1\}$
Patch size	$\{4, 8\}$

This produced 64 ViT training configurations. All models were trained for 300 epochs using AdamW. The learning rate schedule used 10 warmup epochs followed by cosine decay. The model checkpoint with the best validation accuracy was saved and used for downstream SAE training.

The optimization objective used soft-target cross entropy during training, together with Mixup and CutMix augmentation. Mixup used $\alpha = 0.8$, CutMix used $\alpha = 1.0$, with probability 0.5 and switch probability 0.5. Validation used standard cross entropy.

B.3. Image Preprocessing and Augmentation

Training images were transformed using random resized crops to 56×56 pixels with a scale range of $(0.8, 1.0)$, random horizontal flips, color jitter $(0.4, 0.4, 0.4, 0.1)$, RandAugment with two operations and magnitude 9, normalization, and random erasing with probability 0.25. Validation images were resized to 56×56, center-cropped, converted to tensors, and normalized.

All ZOD crops were resized to a fixed resolution of 56×56 pixels to match the preprocessing convention used for GTSRB (Houben et al., 2013), ensuring a consistent input resolution across datasets and experiments.

Normalization used channel means of the GTSRB (0.4112, 0.3572, 0.3556) and standard deviations (0.3062, 0.2936, 0.2970).

C. Sparse Autoencoder Training

C.1. SAE Training Setup

For each trained ViT, we trained SAEs on residual-stream activations from the `hook_resid_post` hookpoint. Unless otherwise stated, SAEs were trained at a fixed transformer layer l, with the layer index specified in each experiment. The hookpoint corresponds to the residual representation after each transformer block.

SAEs were trained with input and output dimensions $d_{\text{in}} = d_{\text{out}} = 192$, matching the ViT residual-stream dimensionality. The SAE dictionary used an expansion factor of 20, resulting in $192 \times 20 = 3840$ dictionary features. SAE training was performed exclusively on CLS-token activations.

Each SAE was trained for 40 epochs with batch size 256 and learning rate 10^{-5}. The L_1 coefficient was set to 10^{-4}. Checkpoints were saved during training, with five checkpoints per SAE run. SAE training used the same image preprocessing pipeline as the ViT training setup.

C.2. SAE Evaluation

After training, each SAE was evaluated on the held-out SAE validation split. Evaluation metrics included reconstruction quality, cross-entropy recovery, and dead neuron count. Per-model SAE metrics were saved as JSON files and consolidated into CSV and Excel files for downstream statistical analysis.

Dead neurons were defined as dictionary features that did not activate over the evaluation set. Cross-entropy recovery measured how much of the original model performance was preserved after replacing the residual-stream activation with its SAE reconstruction.

C.3. Monosemanticity Score and Dead Features

We follow Pach et al. (2025) and adapt their monosemanticity score (MS) to our setting by using CLIP ViT-B/32 image embeddings as the semantic reference space. For each SAE feature, we collect the evaluation images that produce the strongest feature activations and compute the average similarity of their CLIP embeddings. Intuitively, a feature receives a higher score when its top activating images are more semantically similar to one another.

We also report the number of dead features, defined as features that never activate on the evaluation set. In the main text, we summarize SAE quality using the mean monosemanticity score across features and the number of dead features.

Both the MS and dead feature count itself are computed using the script released by (Pach et al., 2025). Here, the MS is computed as a weighted average of pairwise cosine similarities between CLIP image embeddings. Features with zero total pairwise activation weight are counted as dead.

D. Correlations Between SAE Metrics

To analyze the relationship between representation geometry and SAE behavior, we compute Kendall's τ correlations between several SAE metrics and CE recovery across layers. The following tables report the correlation coefficients and corresponding significance values for monosemanticity and dead-feature counts.

Table 2. Kendall's τ correlations between monosemanticity and CE recovery.

Layer	τ	p
3	-0.1538	0.0725
5	-0.1468	0.0864
7	0.0595	0.4869
8	0.1518	0.0763
9	0.1944	0.0231
10	0.2877	0.0008
11	0.3333	0.0001

Table 3. Kendall's τ correlations between dead features and CE recovery.

Layer	τ	p
3	0.0154	0.8575
5	0.0095	0.9122
7	-0.1778	0.0460
8	-0.2866	0.0015
9	-0.3009	0.0009
10	-0.3964	1.0×10^{-5}
11	-0.7445	6.3×10^{-17}

E. Hyperparameter Effects on SAE Reconstructions

For each layer, we report the group means and medians, the U statistic, and the p-value. Significant effects for learning rate are only observed in layer 3 and 5. For weight decay we find a signifiant effect in all analyzed layers.

Table 4. Mann-Whitney U tests for dead-neuron counts across hyperparameter settings.

Layer	Hyperparameter	Comparison	U	p	Result
3	learning rate	0.0001 vs. 0.001	265.0	0.0009	Significant
3	batch size	256 vs. 512	625.5	0.1292	Not significant
3	patch size	4 vs. 8	636.5	0.0959	Not significant
3	dropout	0.0 vs. 0.2	431.0	0.2797	Not significant
3	weight decay	0.0 vs. 0.1	913.0	0.0000	Significant
3	label smoothing	0.0 vs. 0.2	512.0	1.0000	Not significant
5	learning rate	0.0001 vs. 0.001	342.5	0.0230	Significant
5	batch size	256 vs. 512	598.0	0.2500	Not significant
5	patch size	4 vs. 8	581.0	0.3568	Not significant
5	dropout	0.0 vs. 0.2	459.0	0.4800	Not significant
5	weight decay	0.0 vs. 0.1	1009.0	0.0000	Significant
5	label smoothing	0.0 vs. 0.2	463.0	0.5141	Not significant
7	learning rate	0.0001 vs. 0.001	437.5	0.3104	Not significant
7	batch size	256 vs. 512	540.0	0.7062	Not significant
7	patch size	4 vs. 8	576.0	0.3840	Not significant
7	dropout	0.0 vs. 0.2	420.5	0.2122	Not significant
7	weight decay	0.0 vs. 0.1	1007.0	0.0000	Significant
7	label smoothing	0.0 vs. 0.2	435.5	0.2975	Not significant
8	learning rate	0.0001 vs. 0.001	445.0	0.3532	Not significant
8	batch size	256 vs. 512	507.0	0.9499	Not significant
8	patch size	4 vs. 8	554.0	0.5623	Not significant
8	dropout	0.0 vs. 0.2	408.0	0.1485	Not significant
8	weight decay	0.0 vs. 0.1	1004.0	0.0000	Significant
8	label smoothing	0.0 vs. 0.2	435.0	0.2855	Not significant
9	learning rate	0.0001 vs. 0.001	445.0	0.3532	Not significant
9	batch size	256 vs. 512	499.5	0.8669	Not significant
9	patch size	4 vs. 8	540.5	0.6959	Not significant
9	dropout	0.0 vs. 0.2	431.5	0.2640	Not significant
9	weight decay	0.0 vs. 0.1	1012.0	0.0000	Significant
9	label smoothing	0.0 vs. 0.2	440.0	0.3182	Not significant
10	learning rate	0.0001 vs. 0.001	462.0	0.4931	Not significant
10	batch size	256 vs. 512	483.5	0.6983	Not significant
10	patch size	4 vs. 8	521.5	0.9008	Not significant
10	dropout	0.0 vs. 0.2	439.5	0.3188	Not significant
10	weight decay	0.0 vs. 0.1	1015.0	0.0000	Significant
10	label smoothing	0.0 vs. 0.2	433.5	0.2802	Not significant
11	learning rate	0.0001 vs. 0.001	530.5	0.8051	Not significant
11	batch size	256 vs. 512	500.0	0.8747	Not significant
11	patch size	4 vs. 8	519.0	0.9290	Not significant
11	dropout	0.0 vs. 0.2	529.0	0.8211	Not significant
11	weight decay	0.0 vs. 0.1	1023.0	0.0000	Significant
11	label smoothing	0.0 vs. 0.2	377.0	0.0652	Not significant

F. Effective Rank

To analyze the intrinsic dimensionality of learned representations across transformer layers, we computed effective rank on CLS-token activations extracted from the *hook_resid_post* hook point at each layer. This hook point corresponds to the residual-stream representation after the complete transformer block (including attention, MLP, layer norm, and residual connections).

For each layer l, we collected CLS-token activations across a batch of images, forming a centered data matrix:

$$X \in \mathbb{R}^{N \times d}$$

where N is the number of samples and d is the residual-stream dimension, and X has been centered per feature (i.e., $X \leftarrow X - \mathbb{E}[X]$).

Singular value decomposition (SVD) was applied to the centered data:

$$X = U \Sigma V^\top$$

yielding singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d \geq 0$.

Effective rank computation. Singular values with magnitude less than 10^{-10} were removed as numerically insignificant. The remaining singular values were normalized into a probability distribution:

$$p_i = \frac{\sigma_i}{\sum_j \sigma_j}$$

The Shannon entropy of this distribution was computed:

$$H = -\sum_i p_i \log p_i$$

Effective rank was then defined as:

$$r_{\text{eff}} = \exp(H)$$

Effective rank quantifies the number of actively used dimensions in the representation. A value of $r_{\text{eff}} = d$ indicates that all dimensions are equally used (high-dimensional, distributed representation), while $r_{\text{eff}} = 1$ indicates representational collapse into a single dimension. Low effective rank may indicate representational collapse or feature suppression, while high effective rank indicates rich, multi-dimensional representations.

G. Feature Ablation Across ZOD

To evaluate whether SAE features capture class information differently for wd $= 0.0$ vs. wd $= 0.1$, we perform feature-level ablations on ZOD samples. For each class, SAE features are ranked by average activation magnitude, after which the top-k features are iteratively ablated by setting their latent activations to zero before reconstruction. We then measure the resulting decrease in target-class logit margin. Larger margin drops indicate that the ablated SAE features encode information that is more important for the model's prediction.

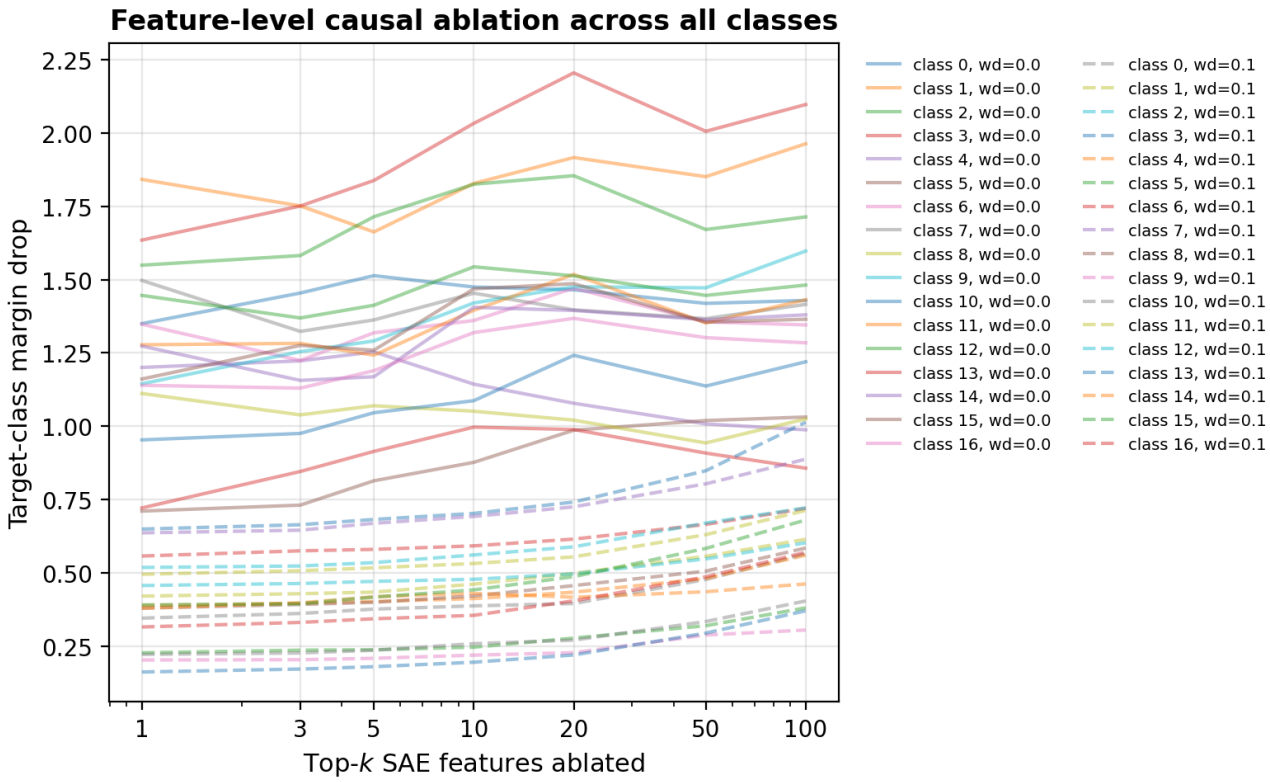

Figure 5. Top-k SAE feature ablation in layer 10 across all ZOD classes.

Across nearly all classes, models trained with wd $= 0.0$ exhibit larger margin drops than models trained with wd $= 0.1$. This suggests that low-weight-decay models rely more strongly on a small set of highly influential SAE features, whereas higher weight decay leads to more distributed representations in which information is spread across a broader set of features. The effect remains consistent as the number of ablated features increases, indicating a systematic shift in representation structure rather than isolated feature dependencies.

H. SAE Feature Usage

Table 5. Layer-wise SAE feature usage statistics for different weight decay values. Statistics are computed using the top-$k = 5$ active features per image over 1700 images.

Layer	Weight decay	Diversity	Most common feature	Max freq. (%)	Gini
0	0.000	8	1176	20.00	0.0164
0	0.001	8	862	20.00	0.0248
0	0.005	20	3539	20.00	0.0376
0	0.010	51	974	15.93	0.1189
0	0.050	688	595	7.52	0.5208
0	0.100	386	1161	8.91	0.5987
1	0.000	7	102	20.00	0.0106
1	0.001	6	926	20.00	0.0128
1	0.005	15	1964	20.00	0.0209
1	0.010	14	1716	20.00	0.0324
1	0.050	673	1991	7.19	0.4340
1	0.100	520	1407	9.25	0.5196
2	0.000	9	2476	20.00	0.0077
2	0.001	7	1923	20.00	0.0105
2	0.005	9	1264	20.00	0.0203
2	0.010	12	1708	20.00	0.0256
2	0.050	685	2468	5.29	0.3420
2	0.100	862	3265	4.24	0.5788
3	0.000	6	2157	20.00	0.0043
3	0.001	5	3139	20.00	0.0054
3	0.005	8	1457	19.94	0.0163
3	0.010	13	3224	19.60	0.0302
3	0.050	803	2297	6.53	0.2736
3	0.100	1041	137	3.67	0.4737
4	0.000	6	121	20.00	0.0031
4	0.001	8	2097	20.00	0.0035
4	0.005	12	291	20.00	0.0095
4	0.010	16	41	20.00	0.0177
4	0.050	740	709	7.08	0.2984
4	0.100	1133	169	2.58	0.3896
5	0.000	6	3124	20.00	0.0013
5	0.001	5	324	20.00	0.0020
5	0.005	7	3770	20.00	0.0059
5	0.010	12	1039	19.99	0.0159
5	0.050	584	2338	4.14	0.3447
5	0.100	1126	1067	7.48	0.3739
6	0.000	6	1673	20.00	0.0005
6	0.001	6	1670	20.00	0.0007
6	0.005	6	318	20.00	0.0029
6	0.010	8	3457	20.00	0.0121
6	0.050	641	484	5.66	0.3306

Layer	Weight decay	Diversity	Most common feature	Max freq. (%)	Gini
6	0.100	1211	2296	2.99	0.4081
7	0.000	5	2953	20.00	0.0003
7	0.001	5	2088	20.00	0.0005
7	0.005	8	1525	20.00	0.0019
7	0.010	11	2911	20.00	0.0096
7	0.050	554	1595	5.54	0.3312
7	0.100	1161	2997	4.34	0.4476
8	0.000	5	3555	20.00	0.0003
8	0.001	6	3592	20.00	0.0004
8	0.005	7	3270	20.00	0.0018
8	0.010	7	1884	20.00	0.0094
8	0.050	477	1257	6.78	0.2952
8	0.100	1093	446	2.46	0.4272
9	0.000	5	428	20.00	0.0003
9	0.001	5	2539	20.00	0.0003
9	0.005	6	3037	20.00	0.0015
9	0.010	10	1821	20.00	0.0055
9	0.050	422	3240	13.35	0.2449
9	0.100	881	3448	4.96	0.3836
10	0.000	5	2836	20.00	0.0002
10	0.001	5	2594	20.00	0.0003
10	0.005	5	2126	20.00	0.0015
10	0.010	8	576	20.00	0.0049
10	0.050	370	117	8.52	0.2590
10	0.100	795	1238	4.00	0.3679
11	0.000	7	1228	20.00	0.0003
11	0.001	6	440	20.00	0.0003
11	0.005	8	1650	20.00	0.0017
11	0.010	9	3270	20.00	0.0049
11	0.050	217	2840	10.54	0.1594
11	0.100	789	2809	3.41	0.3238

I. Gini Coefficient

To measure spatial concentration of SAE features across patches within each image, we computed a Gini coefficient on patch-level feature activations.

For each image:

1. The SAE feature with maximum mean activation across patches was identified.
2. Patch-level absolute activations $a_1, \ldots, a_n$ for this feature were collected.
3. A Gini coefficient was computed over these activations.

The activations were sorted in ascending order, and the Gini coefficient was computed as:

$$G = \frac{2\sum_{i=1}^{n} i \cdot a_i}{n\sum_{i=1}^{n} a_i} - \frac{n+1}{n}$$

where $i \in \{1, \ldots, n\}$ is the rank position and a_i are the sorted activations.

The Gini coefficient ranges from 0 (perfectly uniform activation across patches) to approximately 1 (activation concentrated on a single patch). Higher Gini values indicate that a feature is spatially specialized to a small set of patches, while lower values indicate distributed activation across all patches.

I.1. Class Complexity

To study the effect of task complexity on SAE reconstruction quality, we vary the number of classes while controlling the number of samples per class. Table 6 reports Kendall's τ correlations between class count and average reconstruction CE across different sampling regimes.

Table 6. Kendall's τ between number of classes C and average reconstruction CE on GTSRB.

Samples/class	τ	p	H_0
200	0.810	0.011	reject
400	0.905	0.003	reject
600	0.619	0.069	fail to reject
800	0.810	0.011	reject
1000	0.810	0.011	reject
1200	0.905	0.003	reject