arXiv:2607.17770v1 [cs.CV] 20 Jul 2026

Measuring Monosemanticity in Sparse Autoencoders via Latent Activation Coherence

Katarzyna Filus[1] (✉) and Sebastian Pokuciński[2]

[1] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland
[2] Department of Applied Informatics, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

Abstract. Within Explainable Artificial Intelligence, mechanistic interpretability uses Sparse Autoencoders (SAEs) to extract more interpretable features from neural representations. However, assessing their monosemanticity, and thus explanation quality, remains challenging. Existing metrics require external concept labels or depend on pretrained embedding models, making them sensitive to encoder's geometry. We introduce the Tversky Monosemanticity Score (TMS), a label-free metric that operationalizes monosemanticity as activation-set coherence of binarized SAE latents, and does not require external embedding encoders. We evaluate TMS on SAEs trained on features from pretrained vision and vision-language models (DINOv3, CLIP, BLIP2), two common SAE regimes (TopK, BatchTopK), multiple sparsity levels, and expansion factors. Our results show that TMS is less affected by encoder anisotropy than its embedding-based alternative, while remaining aligned with established monosemanticity indicators. TMS also reveals distinct SAE training dynamics across base models. Moreover, under encoder anisotropy, TMS provides a stronger indication of probe-based concept deletion effectiveness, while being competitive otherwise.

Keywords: Mechanistic interpretability · Sparse Autoencoders · Monosemanticity · Explainable Artificial Intelligence Evaluation.

1 Introduction

Within Explainable Artificial Intelligence (XAI), mechanistic interpretability aims to improve the transparency of deep networks by analyzing their internal representations. A crucial challenge is feature entanglement, where individual neurons encode multiple unrelated concepts, making representations difficult to interpret [2]. Sparse autoencoders (SAEs) emerged as a key technique in this area, encouraging more disentangled features by projecting activations into higher-dimensional sparse latent spaces [8,19]. In this context, monosemanticity plays a key role [2], describing whether a neuron corresponds to a single coherent concept and reflects reduced feature entanglement. Beyond serving as interpretability indicator, monosemanticity holds functional importance and aids in targeted manipulations of model behavior [9].

Besides its acknowledged importance, currently there is no consensus on how monosemanticity of SAE neurons should be defined and measured. Existing approaches are grounded in supervised concept-predictive evaluation [9,5] or similarity among stimuli that activate a given neuron [13]. While these methods provide useful proxies, they typically rely on external resources such as annotated concept labels [9,5], auxiliary classifiers [9], or pretrained embedding models [13]. While methods based on embedding models eliminate the need for concept labels [13], their estimates still largely depend on the representational geometry of external encoders, and factors such as anisotropy. As a consequence, monosemanticity estimates may vary substantially across encoders, complicating direct comparisons between models. These observations highlight the need to develop a monosemanticity measure independent of external labels and embedding models.

Motivated by the limitations of existing monosemanticity metrics, we introduce a novel Tversky Monosemantic Score (TMS), a metric that operationalizes monosemanticity as activation-set coherence within sparse autoencoder latent features. TMS is independent of both external concept labels and test-time reference encoders. The proposed formulation uses a set-based, non-geometric similarity measure inspired by symmetric Tversky's similarity model from cognitive psychology [17]. By operating on binary activation patterns rather than continuous embeddings, TMS is less sensitive to factors induced by embedding geometry, including anisotropy, that distort similarity-based measures. This independence from external resources simplifies the evaluation and makes it more self-contained. We evaluate TMS on SAEs trained on representations from pretrained vision and vision-language models across different sparsity regimes and expansion factors, using Mini-ImageNet as a natural image benchmark. Our results show that, despite TMS' simplicity, it remains aligned with established monosemanticity indicators, while showing stronger agreement with the MonoSemanticity Score (MS) [13] in low-anisotropy settings (DINOv3). TMS trajectories show that monosemanticity is not static but evolves during training, with clear differences across SAE regimes and base models, pointing to a direction for further investigation in SAE research. TMS as a model-level measure exhibits stronger association with target drop in linear probes trained on SAE activations, compared to the reference measure for anisotropic embedders such as CLIP. We provide our code in an anonymous Zenodo repository for reproducibility [6].

2 Related work

Mechanistic interpretability aims to uncover the internal structures responsible for neural networks behavior [12]. A key challenge is *polysemanticity*, meaning that individual neurons in a network respond to multiple unrelated concepts, preventing straightforward interpretation. It is often connected to the superposition hypothesis stating that networks encode more features than neurons through overlapping activation directions [10]. Sparse autoencoders (SAEs) are a promising approach to decompose activations into sparse feature dictionaries that are more disentangled and interpretable, capturing more coherent, single-

concept representations – a property commonly referred to as *monosemanticity*. SAE-based analyses have demonstrated monosemantic feature discovery not only in language models, which were the main focus of the initial studies [3,8,15], but also in vision–language or vision settings, where learned features also align with human concepts and can be used for targeted model steering [13,4].

Available monosemanticity measures include Feature Monosemanticity Score (FMS) [9], Jensen-Shannon divergence-based score (JSDS) [5], and MonoSemanticity score (MS) [13]. FMS uses performance difference for removed or added features in concept classification. JSDS measures how activations separate across concepts with Jensen–Shannon divergence. The primary limitation of JSD and FMS is that they need external concept labels. A recent study introduced the Monosemanticity Score (MS) [13] to avoid this limitation. MS treats a latent neuron as monosemantic if the input features that activate it are mutually similar, by embedding each image with an external encoder and computing a weighted mean cosine similarity between embeddings. While this is an important step toward label-free measures, it remains dependent on the external encoder geometry, and on the encoder choice, which adds variability and complicates cross-model comparisons. Contrastingly, our Tversky Monosemanticity Score is independent of both external concept labels and embedding models. Moreover, while prior studies typically focus on SAEs trained over a single-type modality (e.g. image-text [13], text [9]), we provide an analysis across different modality types – purely visual (DINOv3) and multimodal (CLIP, BLIP2), under consistent SAE training. Finally, beyond the predominantly static monosemanticity evaluation [13,9,5], we complement our analysis with a dynamic tracking of monosemanticity training evolution, offering a broader view on SAE interpretability.

3 Tversky Monosemanticity Score

We define our novel metric – **Tversky Monosemantic Score (TMS)** – following the feature-based similarity model of Tversky [17] from cognitive psychology to operationalize monosemanticity. $\mathcal{A} = \{a, b, c, \dots\}$ denotes the domain of stimuli (images), which are fed to the deep learning model under interpretation. In Tversky's theory, each object is represented as a set of binary features, and similarity between two objects depends on their shared and distinctive features. For objects $a, b \in \mathcal{A}$ with feature sets A and B, similarity can be defined as a function $S(a, b) = F(A \cap B,\ A \setminus B,\ B \setminus A)$, increasing with the number of shared features and decreasing with the number of distinctive features.

In our case, we interpret a trained Sparse Autoencoder (SAE) as a *concept extractor*. Each latent neuron of the trained SAE is treated as a learned *concept proxy* (in the sense of Tversky's feature) and as a detector of recurring activation patterns, rather than a ground-truth semantic concept. Each stimulus (features generated with a model under interpretation for a given image) is described by the set of active latent concepts. Let $B_n \subseteq \{1, \dots, K\}$ denote the binary set of active latent neurons for stimulus n, obtained via a predefined activation rule – following recent work showing that binarization of SAE features can improve

interpretability and provide an effective alternative representation [7,1]. For a latent neuron k and its active set of samples $\mathcal{A}_k = \{\, n \;:\; k \in B_n \,\}$, for which k activated, we measure similarity between pairs $n_i, n_j \in \mathcal{A}_k$ using Tversky index:

$$\mathrm{TI}_{\alpha,\beta}(B_{n_i}, B_{n_j}) = \frac{|B_{n_i} \cap B_{n_j}|}{|B_{n_i} \cap B_{n_j}| + \alpha|B_{n_i} \setminus B_{n_j}| + \beta|B_{n_j} \setminus B_{n_i}|}. \tag{1}$$

For the purpose of this paper, we set α and β to 1, which reduces the score to Jaccard index, as a parameter-free default. The Tversky framing, however, provides a principled cognitive-science foundation and naturally accommodates future asymmetric extensions [17]. A latent neuron is considered more *monosemantic* if, on average, stimuli in its active set share more common than distinctive features, which is reflected in high Tversky's within-set similarity: $s_k = \mathbb{E}_{n_i,n_j \in \mathcal{A}_k}\left[\mathrm{TI}_{\alpha,\beta}(B_{n_i}, B_{n_j})\right]$ If the latent neuron's active set is considered empty, i.e., $\mathcal{A}_k = \varnothing$, its monosemanticity score becomes $s_k = 0$. In practice, we compute $\mathrm{TI}_{\alpha,\beta}$ on binarized latent activation vectors. Given activations $Z \in \mathbb{R}^{N \times K}$, we obtain a binary matrix $B \in \{0,1\}^{N \times K}$ by thresholding each latent k from the test set at its mean training activation μ_k (i.e., $B_{n,k} = \mathbb{I}[Z_{n,k} > \mu_k]$) for a parameter-free, neuron-wise and data-derived threshold, following prior work of [1] using mean-based thresholds computed on SAE training data for its individual neurons. For each latent k, and its active set $\mathcal{A}_k = \{n : B_{n,k} = 1\}$, we estimate the within-set similarity by sampling at max M distinct pairs (n_i, n_j) with $n_i \neq n_j$ uniformly at random from $\mathcal{A}_k$, and averaging similarities.

The score reflects internal *conceptual coherence* of stimuli activating a given latent neuron, rather than geometric properties of the embedding models. The intuition for grounding our operationalization of monosemanticity in latent coherence is that a latent encoding a relatively selective concept may tend to activate on stimuli sharing not only that concept, but also its contextual structure. For example, a 'bird wings' latent may often co-occur with 'beaks' and 'feathers', giving consistent activation patterns, whereas polysemantic latents would produce more heterogeneous ones.

4 Experimental Setup

We consider pretrained models from the HuggingFace repository[3]: a purely visual **DINO-v3** with ConvNeXt backbone (*facebook/dinov3-convnext-tiny-pretrain-lvd1689m*)[16], and vision-language **CLIP** (*openai/clip-vit-base-patch32*)[14] and **BLIP2** (*Salesforce/blip2-opt-2.7b*)[11]. We refer to them as **base models** (when used to extract features for SAE training) and **encoders** (when used to embed images for MS [13] computation). The experiments are conducted on MiniImageNet [18], whose natural categories facilitate qualitative inspection of SAE features. We use DINOv3's global average pooled vector, and CLIP's/BLIP's CLS tokens as embeddings. We train sparse autoencoders (SAEs) adopting two

[3] https://huggingface.co/

common strategies: *TopK* and *BatchTopK* [8], using `overcomplete`[4]. We vary the active features number $K \in 10, 32, 64$ and the expansion factor exp $\in 2, 4, 8$, yielding diverse SAE configurations. We train all SAEs for 25 epochs using the Mean Squared Error loss and AdamW optimizer with 3×10^{-4} learning rate and 512 batch size, while applying the `overcomplete's` reanimation loss with weight 10^{-2}. For TMS computation, we set the max number of sampled pairs at $M = 1000$. An auxiliary stability analysis on an example SAE (BatchTopK, DINOv3, $K = 32$, expansion factor $\times 4$) showed that TMS remains stable across 10 runs with $M \in \{50, 100, 250, 500, 1000, 2500, 5000\}$, with values consistent to three decimal places and standard deviation on the order of 10^{-5} at $M = 1000$. The number of active neurons that contribute to TMS after binarization remained substantial in all configurations, with average TMS always computed over at least several hundred neurons. Detailed statistics for all M and results on active neurons are provided in Appendices C and D. We use MonoSemanticity score (MS) [13] as embedding-based monosemanticity reference. The test environment was a PC with an Intel i9 12900 CPU, RTX 3090 24GB GPU, and 196 GB of DDR5 RAM. Our evaluation has several stages:

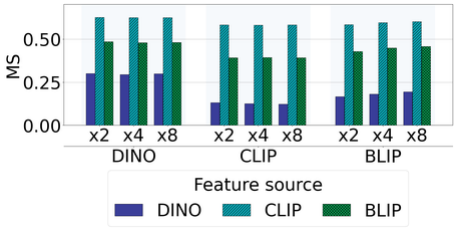
1. **Embedding anisotropy analysis.** We quantify the anisotropy of the base models when used as external encoders for embedding-dependent MS score [13]. This motivates the need for external embedding-free metrics.
2. **Monosemanticity estimation.** For all SAEs, we compute the proposed TMS alongside the reference – MS [13]. To complement quantitative results, we present example top activating images for neurons for all base model.
3. **Metric correlation.** We compute the correlation between monosemanticity estimators by measuring per-neuron Spearman correlations with p-values between TMS and alternative statistics (MS, within-active-set variance with the activation set defined via TMS) for neurons with non-zero TMS.
4. **Training dynamics.** Beyond static evaluation, we examine how monosemanticity evolves during SAE training by tracking TMS and MS values across epochs and visualizing their trajectories using scatter plots.
5. **Concept deletion experiment.** We evaluate the functional relevance of SAE features via a concept deletion experiment. For each SAE, we train a linear probe classifier on Mini-ImageNet and identify the most relevant SAE features for a given class using probe weights (max values), ensuring independence from monosemanticity metrics. We then analyze target accuracy degradation as the top-N features are removed and compute correlations between MS/TMS and the accuracy drops across SAE configurations.To train the probes, we use a scikit-learn's linear probe implemented as Standard Scaler and logistic regression (saga solver, $C = 1.0$, `max_iter`=25).

In Appendices A and B, we report SAE reconstruction and health metrics (R^2, in-out cosine similarity, latent L_0, dead neuron ratio) and pre-deletion probe accuracies (on SAE latents and base model features) to show that the SAEs and probes are sufficiently functional, making our monosemanticity and concept-deletion experiments under a shared setup meaningful.

[4] https://github.com/KempnerInstitute/overcomplete

Model	σ_{cosine}	EVR
DINOv3	0.07	0.02
CLIP	**0.52**	**0.07**
BLIP2	0.33	0.05

(a) Anisotropy statistics of embedding models: mean cosine similarity (σ_{cosine}) and the first principal component's explained variance ratio (EVR). Bold: the highest anisotropy.



(b) Aggregated MS for non-zero neurons (TOPK SAE, K=10).

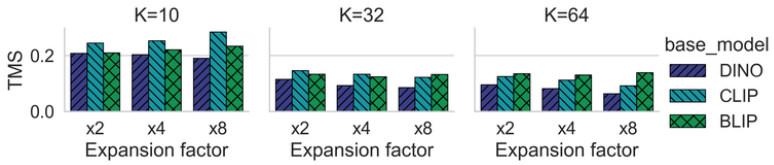
Fig. 1. Anisotropy-induced bias in embedding-based MS. (a) External embedding models exhibit different degrees of anisotropy. (b) MonoSemanticity Score (MS) [13] aggregated over non-zero-MS neurons depends strongly on the embedding source—more anisotropic encoders produce inflated MS results across setups.

5 Experimental results

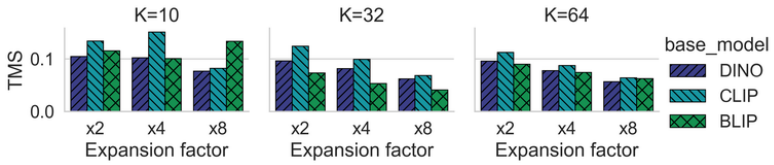
The impact of anisotropy level on monosemanticty level estimation Fig. 1 shows how external encoder geometry can affect monosemanticity estimation based on pairwise sample similarity. We first quantify anisotropy of the feature-source models (Fig. 1a) using mean pairwise cosine similarity between L2-normalized embeddings and the explained variance ratio (EVR) of the first principal component (PC1). Both statistics indicate that CLIP is significantly more anisotropic ($\sigma_{\text{cosine}} = 0.52$, EVR $= 0.07$) than BLIP2 (0.33, 0.05), while DINOv3 has the most isotropic space (0.07, 0.02). Higher σ_{cosine} and EVR suggest a stronger concentration of embeddings along some direction, which can create a bias and inflate similarity estimates. This effect is reflected in embedding-based monosemanticity measures such as MS [13]. Fig. 1 shows MS computed over non-zero values for all pretrained models and configurations for TOPK SAE with $K = 10$. Pretrained models are used both as base models and external encoders for MS (Fig. 1b). MS exhibits a strong dependence on the encoder: CLIP-based MS is consistently the highest, BLIP2 – intermediate, and DINOv3 – the lowest. This ordering follows the anisotropy ranking from Fig. 1, showing that highly anisotropic encoders inflate embedding-based scores, without necessarily reflecting a corresponding increase in monosemanticity. These observations motivate creating a monosemanticity score that is not dependent of external encoders.

Measuring the monosemanticity level In Fig. 2, we present the barplots of TMS values obtained for all the examined base models and SAEs. Mean is computed for the set of non-zero TMS per-neuron values. We present the reference barplot of MS values in Fig. 3 (for the same set of neurons for a fair comparison). In this setup, we use each base model as the encoder for its SAE's MS computation. A clear difference between the two measures emerges. For MS, CLIP dominates with the highest score for all cases. TMS produces a more balanced ordering of models with two examined multimodal models sharing the highest

scores – CLIP (in 13 cases) and BLIP2 (in 5 cases). The overall inter-model gap is notably reduced. When the barplots of TMS and MS are compared in configuration pairs, the agreement between TMS and MS is strongest for DINOv3-based SAEs – likely due to the smallest score inflation due to anisotropy. It suggests that embedding-based MS is less affected by geometric similarity inflation in the more isotropic regime. In contrast, the divergence between MS and TMS for multimodal models indicates that embedding geometry with higher anisotropy can amplify apparent monosemanticity independently of the feature coherence. Overall, the results suggest that TMS captures a notion of within-feature coherence that is more robust to embedding-space anisotropy, leading to a score less dominated by embedding geometry and more reflective of intrinsic feature coherence. Consequently, differences in monosemanticity across SAEs trained on different feature sources appear less pronounced under TMS than under purely geometry-based measures.



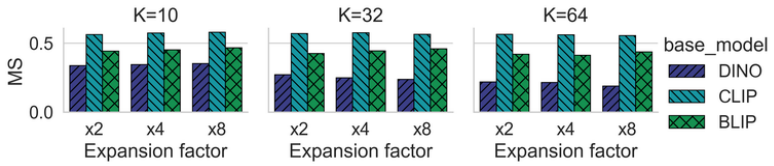
(a) TOPK



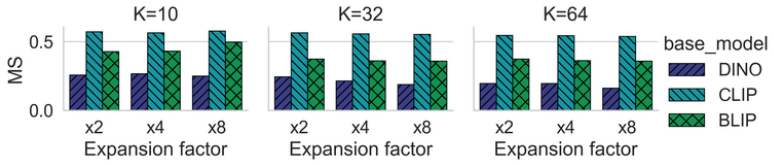
(b) BatchTOPK

Fig. 2. Tversky Monosemantic Score (TMS) results.

Fig. 4 provides qualitative examples of top-activating images for neurons with decreasing TMS values (from left to right) across SAEs trained on DINOv3, CLIP, and BLIP2 features for SAE TopK, $K = 64$, expansion factor 2. A consistent pattern can be observed, in which neurons with high TMS values exhibit visually coherent top activating image sets, sharing a clear semantic/visual motif, whereas lower TMS corresponds to more heterogeneous sets. We highlight neurons for which the TMS and MS orderings differ. While the overall ranking of neurons is largely consistent across measures, the highlighted cases reveal apparent divergences. For example, in the DINOv3 row, neurons with TMS values of 0.12 and 0.05 (showing mixtures of islands/musical instruments, and of animals/textures/boats/food, respectively) receive relatively higher MS scores (approximately 0.3) and an inverse ordering. The latter neuron exhibits a



(a) TOPK



(b) BatchTOPK

Fig. 3. MonoSemantic (MS) [13] score results.

clearly polysemantic activation pattern, and the substantially lower TMS value reflects smaller overlap in its activation sets. For CLIP-based SAEs, neurons with relatively high MS values (e.g. compared to DINOv3) display visually and semantically diverse image sets despite their geometric embedding space proximity. In contrast, TMS assigns lower scores to such neurons, reflecting reduced consistency in their activation behavior. These qualitative observations support the interpretation of TMS as capturing within-feature coherence as proxies for monosemanticity rather than embedding-driven similarity, and provide an intuitive complement to the quantitative results in Figs. 2 and 3.

Correlation with existing monosemanticity indicators To further validate TMS as an indicator of monosemanticity through feature coherence, we analyze its relationship with two neuron-level properties: (i) our main reference – the MonoSemanticity Score (MS), and (ii) the embedding variance within the activation sets. The second reference serves as an independent structural proxy, as monosemantic neurons are expected to activate on more homogeneous sets of stimuli (expressed with features of base models), which should result in lower variability in their activation sets (based on TMS binarization).

Tab. 1 presents the per-neuron correlations between TMS and MS across all SAE variants. MS is computed with a base model used as encoder. Overall, a strong positive relationship is observed, particularly for DINOv3-based SAEs (the least anisotropic encoder), where correlations exceed 0.75 in most cases (16/18). This indicates that TMS captures a notion of feature monosemanticity largely overlapping with embedding-based measures, despite being computed without reliance on external representations. For CLIP-based SAEs, correlations exhibit greater variability, reflecting the influence of embedding geometry and anisotropy. Nevertheless, moderate to strong correlations are still present in all configurations, showing partial alignment between measures.

Fig. 4. Example top 9 activating images for neurons with different TMS. Values decrease from left to right. Top row includes examples for SAE-DINOv3, middle row — CLIP, and bottom row – BLIP2. All SAEs are TopK, expansion factor 2 and K = 64. We use bold for cases where ordering is different for TMS and MS.

In Tab. 2 we present correlations between TMS and within-activation-set variance of feature embeddings via the base model. Across all models and SAE setups, the correlations are strongly negative, frequently exceeding −0.9. This result supports the interpretation that higher TMS values correspond to more coherent activation patterns, aligned with the expected behavior of monosemantic features. The effect is particularly pronounced for BLIP2-based SAEs, where correlations approach −0.9 across nearly all configurations. These findings support that TMS reflects both the alignment with existing monosemanticity indicators and coherence of neuron activations.

Evolution of monosemanticity during SAE training To see how monosemanticty changes over the course of optimization, we analyze the dynamics of MS and TMS throughout SAE training. Figs. 5 and 6 report how both measures evolve across epochs for all the base models and SAE configurations. It is visible that both measures have similar trajectory trends for the majority of plots when compared in pairs – again mostly visible for DINOv3-based SAEs (increase at the beginning, followed by drop and stabilization). Trajectories show that monosemanticity in SAEs trained with different base model features evolves differently (e.g. the trend described for DINOv3 vs the TMS decreasing trend for BLIP2 (without the initial "bump"). TMS shows that monosemanticity of TopK SAEs with smaller K regime ($K = 10$) results in significantly higher scores over training than for higher K values. Contrastingly, for the BatchTopK vari-

Table 1. MS correlation grouped by expansion factor and K for TOPK. All reported correlations are statistically significant (p-values $<<$ 0.05).

SAE	Model	x2			x4			x8		
		K=10	K=32	K=64	K=10	K=32	K=64	K=10	K=32	K=64
TopK	DINOv3	**0.88**	**0.92**	**0.87**	**0.87**	**0.86**	**0.86**	**0.86**	**0.78**	0.77
	CLIP	0.54	0.77	0.86	0.53	0.75	0.83	0.47	0.70	**0.78**
	BLIP2	0.68	0.75	0.79	0.62	0.75	0.77	0.64	0.75	0.75
BatchTopK	DINOv3	**0.87**	**0.93**	**0.84**	0.65	**0.89**	0.89	0.50	**0.78**	0.85
	CLIP	0.35	0.60	0.90	0.22	0.58	**0.90**	0.27	0.61	**0.87**
	BLIP2	0.81	0.79	0.82	**0.80**	0.68	0.78	**0.79**	0.69	0.76

Table 2. Variance correlation itself grouped by K and expansion factor for TOPK. All reported correlations are statistically significant (p-values $<<$ 0.05).

SAE	Model	x2			x4			x8		
		K=10	K=32	K=64	K=10	K=32	K=64	K=10	K=32	K=64
TopK	DINOv3	-0.85	-0.85	-0.76	-0.87	-0.83	-0.83	-0.84	-0.80	-0.82
	CLIP	-0.81	-0.90	-0.88	-0.79	-0.90	-0.91	-0.74	-0.90	-0.91
	BLIP2	**-0.90**	**-0.93**	**-0.92**	**-0.90**	**-0.94**	**-0.94**	**-0.91**	**-0.93**	**-0.93**
BatchTopK	DINOv3	-0.81	-0.82	-0.70	-0.64	-0.86	-0.85	-0.49	-0.81	-0.88
	CLIP	-0.45	-0.71	-0.92	-0.47	-0.65	**-0.93**	-0.60	-0.68	-0.90
	BLIP2	**-0.90**	**-0.86**	**-0.88**	**-0.91**	**-0.78**	-0.90	**-0.90**	**-0.84**	**-0.89**

ant, all SAE configurations converge more closely and obtain lower values. MS exhibits more irregular epoch-to-epoch fluctuations than TMS (e.g. MS for the CLIP-based TopK SAE, in which MS increases quickly in early epochs and then oscillates with relatively high-frequency fluctuations). Despite these differences, both measures tend to display synchronized local fluctuations, indicating that certain training phases induce genuine variability in feature emergence (e.g. for DINOv3-based TopK SAE with k=32/expansion factor 8). These observations suggest that monosemanticity is not static during SAE training but undergoes distinct phases of feature development, while its dynamics are different for SAEs trained with different base model features and sparsity. TMS and MS can capture these phenomena, highlighting the need for such dynamic examinations.

Concept deletion effectiveness To assess whether model-level TMS monosemanticity translates into effective feature-level interventions for a given SAE, we perform a probe-based concept deletion experiment. For each SAE, we train a linear probe on SAE representations and mini-ImageNet classes. We zero the top-N latent features most relevant to a given class, ranked by the magnitude of the probe weights, treating classes as higher-level semantic targets. The resulting drop in target-class accuracy (target drop) serves as a measure of the functional relevance of SAE features. Fig. 7 shows target drop as a function of the number

of removed features N. Distinct behaviors emerge across base models. For DINOv3 SAEs, smaller K and lower expansion factors yield the strongest deletion effects. For CLIP and BLIP2 SAEs, smaller K also tends to produce stronger target drops at small N (even $N = 1$). This is consistent with our earlier TMS results, where these configurations exhibited higher monosemanticity, suggesting that tighter sparsity yields more concentrated features. For CLIP, the $K = 10$ settings show comparable or smaller drops at larger N, likely due to their lower probe accuracies ($\sim$ 60–70%). Similar behavior for BLIP2 BatchTopK SAEs despite high probe accuracies ($> 90\%$) may indicate greater feature overlap.

To quantify the link between monosemanticity and deletion effectiveness, we compute correlations between SAE-level mean TMS/MS and target drop across all trained SAEs for each N (Tab. 3). For small N, both metrics show strong positive correlations with target drop for all models, indicating that more monosemantic SAEs enable more effective interventions. For CLIP-based SAEs, correlations become non-significant for $N = 16$ and $N = 32$, likely due in part to lower downstream probe performance. Comparing the two measures, TMS gives stronger correlations for CLIP-based SAEs (where MS is impacted by higher anisotropy), while comparable, but slightly weaker results in less anisotropic settings (DINOv3 and BLIP2 at smaller N). Overall, both metrics capture functional feature relevance, with TMS offering better predictive value when embedding geometry may distort similarity-based assessments.

Table 3. Correlation analysis of TMS/MS-target drop across different models and N values. All correlations computed with corresponding p-values. Best results for each model marked in bold. Non-significant results (p-value > 0.05) in italics.

Model	N	Target drop - TMS		Target drop - MS	
		Correlation	p-value	Correlation	p-value
DINOv3	1	0.837	0.000015	**0.878**	0.000002
	2	0.878	0.000002	**0.915**	0.000000
	4	0.876	0.000002	**0.893**	0.000001
	8	0.843	0.000011	**0.905**	0.000000
	16	0.825	0.000026	**0.917**	0.000000
	32	0.868	0.000003	**0.917**	0.000000
CLIP	1	**0.783**	0.000121	0.721	0.000728
	2	**0.827**	0.000023	0.707	0.001037
	4	**0.864**	0.000004	0.653	0.003286
	8	**0.678**	0.001985	0.534	0.022588
	16	*0.406*	*0.094953*	*0.257*	*0.303309*
	32	*0.158*	*0.531483*	*-0.001*	*0.996757*
BLIP2	1	0.719	0.000767	**0.818**	0.000033
	2	0.674	0.002166	**0.785**	0.000113
	4	0.622	0.005819	**0.750**	0.000335
	8	0.670	0.002360	**0.771**	0.000181
	16	**0.697**	0.001318	0.662	0.002791
	32	**0.732**	0.000558	0.577	0.012195

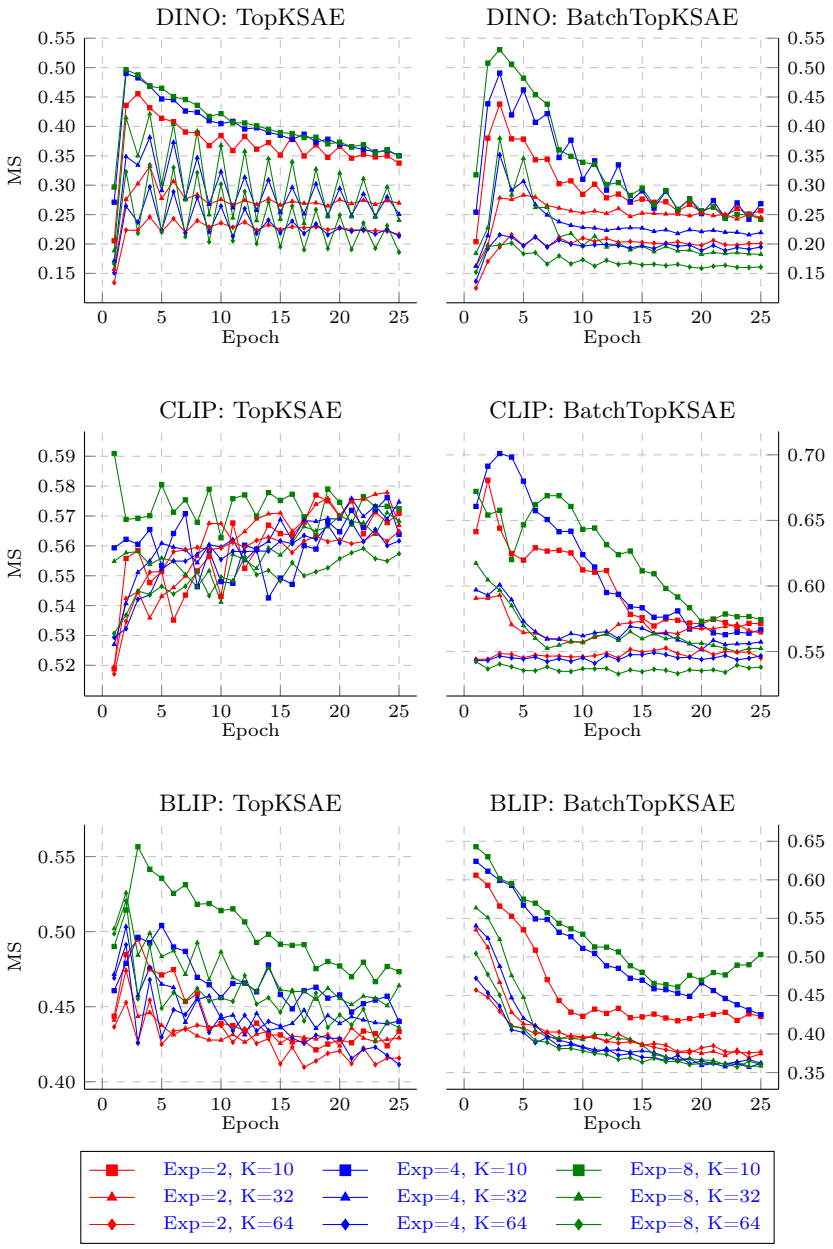


Fig. 5. Epoch evolution of MS for TopKSAE (left column) and BatchTopKSAE (right column) across DINO, CLIP, and BLIP models.

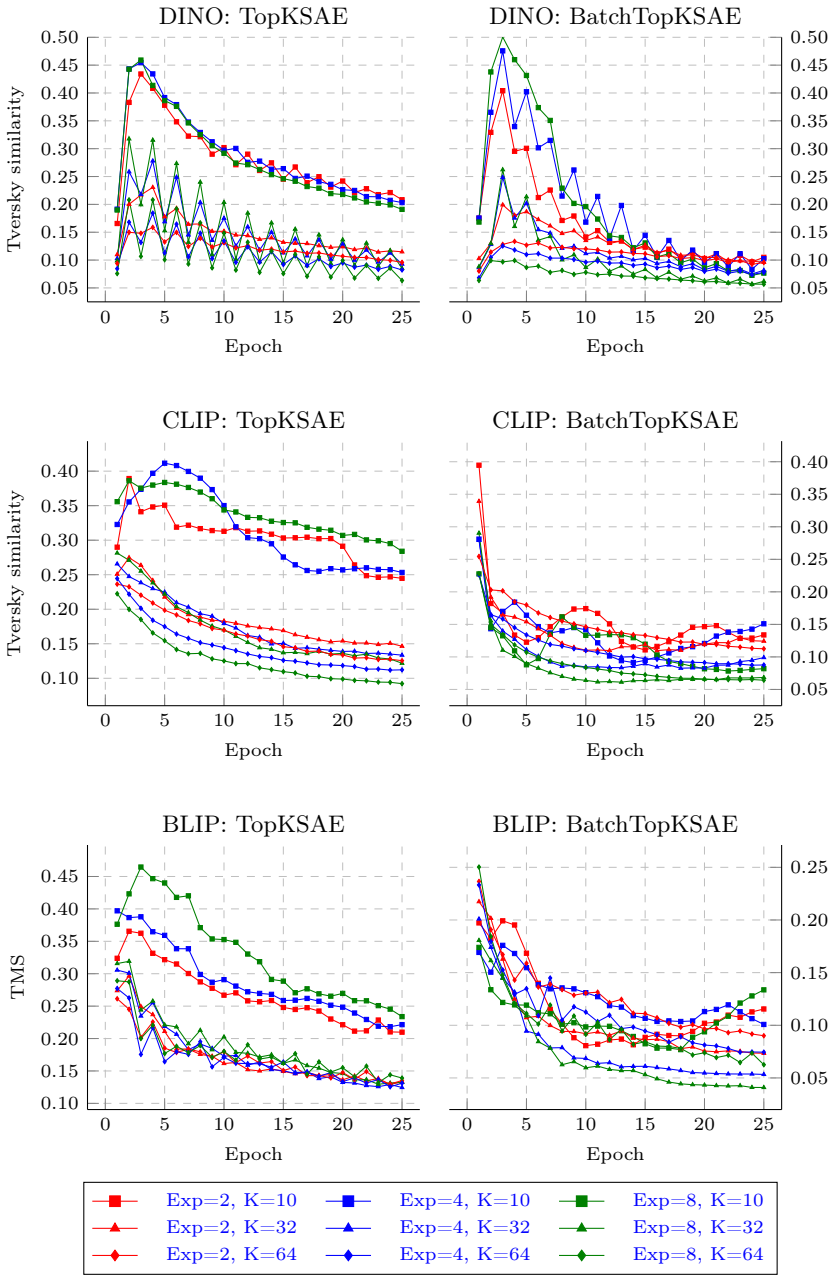


Fig. 6. Epoch evolution of TMS for TopKSAE (left column) and BatchTopKSAE (right column) across DINO, CLIP, and BLIP models.

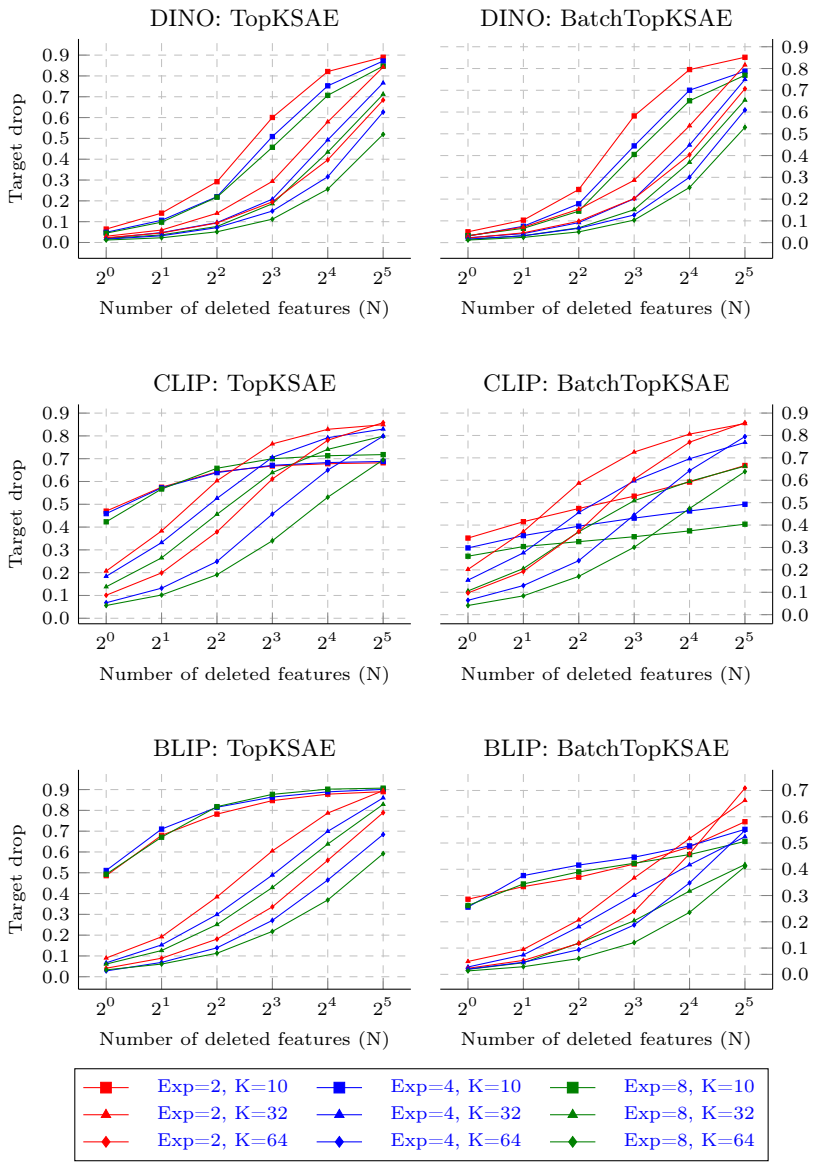


Fig. 7. Deletion curves showing target drop as a function of number of deleted features (N). Left column: TopKSAE, right column: BatchTopKSAE. Rows: DINO, CLIP and BLIP models, respectively.

6 Conclusions

In this paper, we introduced the Tversky Monosemanticity Score (TMS), a metric that operationalizes monosemanticity as activation-set coherence of binarized SAE latents. Across multiple SAEs trained on DINOv3, CLIP, and BLIP2 features, we showed that TMS is less affected by encoder anisotropy than the reference embedding-based metric, while remaining correlated with established monosemanticity indicators. Moreover, TMS more clearly revealed differences in training dynamics across base models and SAE regimes during optimization. Finally, our probe-based concept deletion study showed that TMS as a model-level metric was more strongly associated with deletion effectiveness under higher encoder anisotropy than the reference while remaining competitive otherwise. By operating on SAE latents, we eliminate the cost connected to external embedding generation of methods relying on external encoders, varying depending on hardware used, which in our experiments was up to app. 30 mins on the BLIP2 test set. Although our method operates on higher-dimensional SAE representations, its similarity computation is based on sparse vectors, thresholding and binary operations, which are typically computationally cheaper than the floating-point arithmetic on dense vectors, and allow implementation-level optimization.

We adopted Tversky's feature-based similarity model because it provides the conceptual basis for representing latent activations and their comparisons through shared and distinctive components, as a coherence-based measure. TMS's correlation with other monosemanticity indicators supports its use as a useful estimation aid, especially for natural-image data, where concepts often co-occur with reasonable contextual structure. Practically, against degenerate co-activations, the presented within-active-set variance can be used as a complementary sanity check. Our future work will focus on extending the proposed parameter-free baseline with the asymmetric Tversky variant, more advanced binarization [7,1], and a broader evaluation across tasks. Nevertheless, our current study with multiple base models, SAE regimes, sparsity levels, expansion factors, a natural-image task and from different perspectives, shows the potential of TMS for SAE verification without external concept labels or encoders.

Preprint notice This is a preprint version. A shorter version of this paper has been accepted for presentation and publication in the post-workshop proceedings of the 8th International Workshop on eXplainable Knowledge Discovery in Data Mining (XKDD 2026), co-located with ECML PKDD 2026. The appendix is included only in this preprint and is not part of the peer-reviewed proceedings paper.

Acknowledgments. This research was supported by the START Scholarship of the Foundation for Polish Science (FNP) for outstanding young scholars, agreement No. START 017.2025, the Polish Minister of Science and Higher Education scholarship for outstanding young researchers, agreement No. SMN/20/1546/2024, and the Research Funds for Young Scientists at the Silesian University of Technology (agreements No. 02/100/BKM25/0045 and 02/100/BKM26/0055). The authors utilized LLMs for language polishing under human supervision and take fully responsible for the manuscript.

Disclosure of Interests. The authors have no competing interests.

References

1. Aswal, D., Hudelot, C.: Llmsymguard: A symbolic safety guardrail framework leveraging interpretable jailbreak concepts. arXiv e-prints pp. arXiv–2508 (2025)
2. Bereska, L., Gavves, E.: Mechanistic interpretability for ai safety–a review. arXiv:2404.14082 (2024)
3. 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., Olah, C.: Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread (2023)
4. Cywiński, B., Deja, K.: Saeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders. arXiv:2501.18052 (2025)
5. Fereidouni, M., Haider, M.U., Ju, P., Siddique, A.: Evaluating sparse autoencoders for monosemantic representation. arXiv:2508.15094 (2025)
6. Filus, K., Pokuciński, S.: Measuring monosemanticity in sparse autoencoders via latent activation coherence - source code (2026), Zenodo repository (private review link, will be made fully public on acceptance)
7. Gallifant, J., Chen, S., Sasse, K., Aerts, H., Hartvigsen, T., Bitterman, D.: Sparse autoencoder features for classifications and transferability. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 29927–29951 (2025)
8. Gao, L., la Tour, T.D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., Wu, J.: Scaling and evaluating sparse autoencoders. arXiv:2406.04093 (2024)
9. Härle, R., Friedrich, F., Brack, M., Wäldchen, S., Deiseroth, B., Schramowski, P., Kersting, K.: Measuring and guiding monosemanticity. arXiv:2506.19382 (2025)
10. Huben, R., Cunningham, H., Smith, L.R., Ewart, A., Sharkey, L.: Sparse autoencoders find highly interpretable features in language models. In: The Twelfth International Conference on Learning Representations (2023)
11. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742 (2023)
12. Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., Mordvintsev, A.: The building blocks of interpretability. Distill **3**(3), e10 (2018)
13. Pach, M., Karthik, S., Bouniot, Q., Belongie, S., Akata, Z.: Sparse autoencoders learn monosemantic features in vision-language models. arXiv:2504.02821 (2025)
14. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763 (2021)
15. Rajamanoharan, S., Lieberum, T., Sonnerat, N., Conmy, A., Varma, V., Kramár, J., Nanda, N.: Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. arXiv:2407.14435 (2024)
16. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt,

L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: Dinov3 (2025)
17. Tversky, A.: Features of similarity. Psychological review **84**(4), 327 (1977)
18. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in Neural Information Processing Systems **29** (2016)
19. Zaigrajew, V., Baniecki, H., Biecek, P.: Interpreting clip with hierarchical sparse autoencoders. arXiv:2502.20578 (2025)

Appendices: Measuring Monosemanticity in Sparse Autoencoders via Latent Activation Coherence

A Accuracy of SAEs

In Tab. A.1, we provide the standard training metrics obtained for all the SAE setups used in our study after the final epoch 25: R^2, L_0, input-output Cosine Similarity and dead neurons ratio. R^2 and dead neurons ratio are obtained directly via overcomplete pipeline, while we compute L_0 and input-output Cosine Similarity on the 10% split of our test dataset. Overall, the results indicate that the large majority of SAEs trained with our unified procedure are at least relatively functionally strong. In particular, the minimum input-output Cosine Similarity is 0.6636, 20 out of 27 BatchTopK models exceed 0.8, all TopK models exceed 0.8, and the best-performing setups achieve values above 0.9. Similarly, most models obtain $R^2 > 0.7$, while dead neurons are effectively eliminated due to the reanimation loss recommended in the overcomplete library. Therefore, although a configuration-specific strategy to tune the parameters would likely provide us with even stronger reconstruction metrics, the obtained models are sufficiently functional to support a fair comparative study of monosemanticity under a shared setup.

B Accuracy of linear probes

In Tab. B.1, we report the probe accuracy obtained for the concept deletion experiment before the deletion. Specifically, we include the accuracy of probes trained on raw features extracted from the base models (DINOv3, CLIP, and BLIP), as well as on the latent activations of all SAEs considered in our experiments. The results show that, across all data sources, it is possible to train relatively accurate probes in the 100-class classification setting via mini-ImageNet. The best overall performance is obtained for BLIP, for which all accuracies exceed 0.9, and for DINO, where 18/19 cases achieve accuracy above 0.9. Weaker results are observed for CLIP embeddings and the latents of SAEs trained on CLIP representations. However, even in this case, the majority of accuracies remain above 0.85. Overall, although both the SAEs and the probes were trained under a unified procedure rather than being individually tuned for a particular configuration, the obtained results are sufficiently strong to support the experiments regarding the concept deletion experiment.

Table A.1. Training metrics of SAE models across different feature extractors and configurations. R^2 denotes explained variance, L_0 is the fraction of zero activations in latent codes, CS denotes mean cosine similarity between input and reconstruction, and Dead features ratio is the fraction of latent dimensions never activated during training. Best values per feature extractor marked in bold.

Model	Exp. Factor	K	R^2		L_0		CS		Dead Features	
			Batch TopK	TopK	Batch TopK	TopK	Batch TopK	TopK	Batch TopK	TopK
DINO	×2	10	0.67	0.68	0.99	0.99	0.80	0.81	0.001	0.000
		32	0.71	0.71	0.97	0.98	0.84	0.83	0.000	0.000
		64	0.72	0.72	0.95	0.96	0.85	0.84	0.000	0.000
	×4	10	0.69	0.70	0.99	1.00	0.81	0.83	0.003	0.000
		32	0.73	0.73	0.99	0.99	0.85	0.84	0.000	0.000
		64	0.75	0.74	0.98	0.98	0.86	0.85	0.000	0.000
	×8	10	0.70	0.72	1.00	1.00	0.82	0.83	0.036	0.003
		32	0.76	0.75	0.99	0.99	0.86	0.85	0.000	0.000
		64	0.77	0.76	0.99	0.99	0.86	0.86	0.000	0.000
CLIP	×2	10	0.67	0.71	0.96	0.99	0.76	0.84	0.000	0.000
		32	0.85	0.85	0.92	0.97	0.91	0.92	0.000	0.000
		64	0.89	0.89	0.92	0.94	0.94	0.94	0.000	0.000
	×4	10	0.67	0.71	0.96	1.00	0.74	0.84	0.000	0.000
		32	0.85	0.85	0.95	0.98	0.89	0.92	0.000	0.000
		64	0.89	0.90	0.95	0.97	0.94	0.94	0.000	0.000
	×8	10	0.69	0.70	0.98	1.00	0.75	0.84	0.004	0.000
		32	0.86	0.86	0.97	0.99	0.88	0.92	0.000	0.000
		64	0.90	0.90	0.97	0.98	0.94	0.94	0.000	0.000
BLIP	×2	10	0.62	0.65	0.97	1.00	0.68	0.81	0.004	0.001
		32	0.80	0.79	0.97	0.99	0.88	0.89	0.001	0.002
		64	0.83	0.81	0.97	0.98	0.91	0.90	0.004	0.030
	×4	10	0.62	0.67	0.98	1.00	0.68	0.82	0.012	0.006
		32	0.80	0.80	0.98	0.99	0.87	0.89	0.006	0.004
		64	0.83	0.83	0.98	0.99	0.91	0.91	0.011	0.058
	×8	10	0.64	0.67	0.99	1.00	0.66	0.82	0.044	0.020
		32	0.81	0.81	0.99	1.00	0.87	0.89	0.018	0.013
		64	0.83	0.79	0.99	0.99	0.91	0.90	0.030	0.121

Table B.1. Reconstruction accuracy of SAE models across different feature extractors and configurations. Baseline Acc denotes linear probe accuracy on raw features. SAE Acc denotes linear probe accuracy on SAE latent representations. Best SAE result per feature extractor marked in bold.

Model	Exp. Factor	K	Baseline Acc	SAE Acc	
				BatchTopK	TopK
DINO	×2	10	93.08%	90.90%	90.31%
		32		91.85%	91.55%
		64		**92.36%**	92.09%
	×4	10		90.49%	90.26%
		32		91.65%	91.45%
		64		92.06%	91.99%
	×8	10		89.80%	90.05%
		32		91.50%	91.54%
		64		91.82%	91.77%
CLIP	×2	10	89.83%	70.93%	68.36%
		32		86.56%	85.34%
		64		87.47%	87.25%
	×4	10		59.67%	68.96%
		32		86.93%	84.87%
		64		87.87%	86.89%
	×8	10		56.45%	72.06%
		32		86.67%	84.55%
		64		**88.24%**	86.84%
BLIP	×2	10	96.49%	90.14%	89.42%
		32		95.35%	94.85%
		64		**95.47%**	95.09%
	×4	10		91.07%	90.33%
		32		95.14%	94.71%
		64		95.34%	94.89%
	×8	10		89.46%	90.75%
		32		94.85%	94.55%
		64		95.09%	94.82%

C TMS stability under sampling

To examine the stability of the sampling-based TMS computation, we conducted an additional experiment on an example SAE configuration: BatchTopK trained on DINOv3 features with sparsity $K = 32$ and expansion factor $\times 4$, using the same training setup as for all other SAEs considered in the paper. Table C.1 reports the mean and standard deviation of the estimated mean TMS for different values of samples used as an upper limit on the number of pairs M over 10 runs. The results show that the estimate is rather stable across M values, with the mean remaining unchanged at approximately 0.084 to three decimal places across all tested values. However, as could be predicted, the variability slightly decreases as the number of sampled pairs increases: for $K = 1000$, which is the value used throughout the paper, the standard deviation is already on the order of 10^{-5}, indicating sufficient stability. Even smaller budgets appear reasonably stable for fast comparisons in our case, while larger values provide even higher consistency across repetitions.

Table C.1. Stability of mean TMS for different numbers of upper limit on the number of sampled pairs (M).

M	mean	std
50	0.084	1.88×10^{-4}
100	0.084	1.99×10^{-4}
250	0.084	9.30×10^{-5}
500	0.084	6.78×10^{-5}
1000	0.084	6.14×10^{-5}
2500	0.084	2.94×10^{-5}
5000	0.084	2.46×10^{-5}

D Active latent neurons after binarization

To better contextualize the Tversky Monosemanticity Score (TMS), we report the number of latent neurons with non-empty active sets A_k after applying our mean-based binarization rule in Tab. D.1. For reference, the original feature dimensionalities of the analyzed base models are 768 for DINO, 512 for CLIP, and 1408 for BLIP, which correspond to latent dimensionalities scaled by the considered expansion factors $\times 2$, $\times 4$, and $\times 8$. The reported counts indicate how many latent neurons are considered active after binarization, therefore, how many neurons effectively participate in TMS computation. These statistics reveal a difference between TopK and BatchTopK SAEs regimes. For BatchTopK, the number of active neurons is typically closer to the full latent dimensionality, indicating the expected broad utilization of the SAE dictionary, which is especially visible for CLIP-based SAEs. In this case, nearly all neurons become active across configurations. DINOv3-based SAEs also exhibit high coverage, while BLIP2-based

SAEs, although still broadly utilized under BatchTopK, remain visibly below full saturation, especially for larger expansion factors. In contrast, TopK SAEs often use only a subset of their latent space after binarization, which depends on both K and expansion factor. For all base models, increasing K leads to a visible increase in the number of active neurons, whereas increasing the expansion factor reduces the fraction of neurons that are active. This means that although larger expansion increases the overall latent dimensionality, the effective binarized dictionary size does not grow proportionally under TopK, which is mostly observed for BLIP-based SAEs. Overall, the raw latent dimensionality can differ from the number of neurons effectively contributing to TMS. Active-neuron statistics therefore can help contextualize the reported TMS values by clarifying and confirming sufficiently large neuron sets for model-level aggregation.

Table D.1. Active neurons grouped by K and expansion factor for TOPK.

SAE	Model	x2			x4			x8		
		K=10	K=32	K=64	K=10	K=32	K=64	K=10	K=32	K=64
TopK	DINO	1131	1533	1535	1460	2908	3043	1811	4376	5732
	CLIP	569	954	1021	659	1606	2015	677	2466	3766
	BLIP	670	1639	2300	633	2192	3627	630	2475	4200
BatchTopK	DINO	1535	1535	1536	2714	3058	3070	4492	5928	6095
	CLIP	1024	1024	1024	2048	2047	2046	4080	4084	4088
	BLIP	2644	2735	2685	5071	5282	5096	9104	9613	9243