RT-TopKSAE: Improving Top-k Sparse Autoencoders with the Rotation Trick

Sulayman Yusuf*
University of Washington

Saksham Nirvan
University of Wisconsin-Madison

Aishwarya Balwani†
St. Jude Children's Research Hospital

Abstract

Mechanistic interpretability seeks to reverse-engineer computational circuits learnt by neural networks and sparse autoencoders (SAEs) have emerged as a key tool for this endeavor given their ability to decompose neural activations into interpretable feature dictionaries in an unsupervised manner. However, SAEs face persistent challenges, viz. collapsed feature usage (dead latents), polysemantic features that confound interpretation, and fundamental non-identifiability that admits multiple equally-valid decompositions. We hypothesize these pathologies share a common origin in gradient geometry, specifically, that standard training loses critical directional information when backpropagating through the discrete Top-K selection operation, typically implemented via straight-through estimation. We therefore introduce the RT-TopKSAE, which recovers this directional information by preserving angular relationships at the Top-K selection boundary. This geometric inductive bias yields appreciable improvements in dictionary utilization, feature disentanglement, and activation uniformity while maintaining representational fidelity. Our results suggest that preserving gradient angle information acts as a geometric regularizer, that, while not resolving identifiability in principle, empirically biases optimization toward representations with desirable interpretability properties.

1 Introduction

Mechanistic interpretability seeks to understand neural networks by decomposing their learnt representations into interpretable components that correspond to meaningful computational primitives Braun et al. (2024), with sparse autoencoders (SAEs) having emerged as a leading approach for this decomposition (Templeton et al., 2024; Gao et al., 2025) thanks to their ability to train overcomplete feature dictionaries that reconstruct network activations as sparse linear combinations of learnt features in a completely unsupervised manner. The appeal is clear: if successful, SAEs provide a monosemantic basis in which each feature captures a distinct, interpretable concept, enabling researchers to trace information flow and identify computational circuits.

Top-K SAEs, which enforce sparsity by selecting only the K largest activations, offer particularly favorable reconstruction-sparsity tradeoffs (Gao et al., 2025; Makhzani & Frey, 2014). However, like other SAE variants, they suffer from persistent pathologies that undermine their interpretability promise. *Dead latents*, i.e., dictionary features that never activate, leave large swathes of the representational space unexplored. *Polysemantic features* conflate multiple concepts, defeating the goal of monosemantic decomposition. And even more fundamentally, sparse coding admits *non-unique solutions*: multiple dictionaries can reconstruct the same activations equally well, raising questions about which decomposition, if any, reflects the network's true computational structure.

We hypothesize these pathologies share a geometric origin: in the gradient flow where activations are mapped to sparse feature codes. Top-K selection introduces a discrete, non-differentiable boundary

*First Author
†Corresponding Author

between selected and non-selected features, conventionally handled via the straight-through estimator (STE) (Bengio et al., 2013). While the STE preserves gradient magnitude, it discards angular information, i.e., the geometric relationship between the pre-selection activations (e) and their gradients $\nabla_e L$ is lost. Features that fall just below the Top-K threshold receive no feedback about how to adjust their orientation to become more selectable. Consequently, features that are selected may not receive signals promoting orthogonality or alignment with other features.

High-quality sparse representations, by contrast, should exhibit specific geometric properties: i) uniform coverage across the representational manifold, avoiding dead regions; ii) feature disentanglement, with distinct features capturing orthogonal aspects of variation; and, iii) alignment with underlying generative factors or computational roles. These properties have natural geometric interpretations: uniform distribution on the feature sphere, large angular separation between active features, and basis alignment with factor subspaces. Yet without angular gradient information to guide optimization, standard SAE training fails to discover such structured configurations.

Recent work on vector quantization (Fifty et al., 2025) demonstrated that preserving gradient angles through the "rotation trick" dramatically improves codebook organization in vector-quantized variational autoencoders (VQ-VAEs) (van den Oord et al., 2017). We adapt this geometric principle to SAEs, hypothesizing that angle-preserving gradients provide the directional feedback needed to guide optimization toward geometrically structured solutions.

Hence, we introduce the Rotation Trick Top-K Sparse Autoencoder (RT-TopKSAE), which preserves angular relationships at the Top-K selection boundary. Our specific contributions are:

1. **Geometric gradient flow for SAEs**: We adapt the rotation trick to Top-K sparse autoencoders, preserving the angle $\angle(e, \nabla_e L) = \angle(q, \nabla_q L)$ where $q = \text{TopK}(e, k)$. This provides richer geometric feedback during optimization, particularly to non-selected features, while maintaining gradient information for selected features.
2. **Dramatic improvements in dictionary organization**: RT-TopKSAE achieves 100% dictionary utilization compared to TopKSAE's 47%, eliminating dead latents entirely. Beyond utilization, we demonstrate $6.1\times$ reduction in feature overlap (Jaccard similarity: 0.071 vs 0.431) and $1.5\times$ more uniform activation distributions (Gini coefficient: 0.036 vs 0.054), indicating that the full dictionary is not only used but well-organized with disentangled, uniformly distributed features.
3. **Geometric regularization toward interpretability**: Our results show that preserving gradient angles acts as an implicit geometric regularizer. While not resolving non-identifiability in principle, this bias empirically steers optimization toward solutions with properties desirable for mechanistic interpretability: comprehensive coverage, minimal polysemanticity, and uniform utilization. We maintain similar representational fidelity (CKA stability: 0.88 vs 0.90), demonstrating these gains come without sacrificing reconstruction quality.

2 Methods

2.1 Background and Overview

Let $x \in \mathbb{R}^d$ denote the SAE input and $e \in \mathbb{R}^m$ the encoder output, where $m > d$ is the dictionary size. Top-K selection yields $q = \text{TopK}(e, k) \in \mathbb{R}^m$ with sparsity level k, and the decoder produces reconstruction $\hat{x} \in \mathbb{R}^d$. Encoder and decoder weights are $W_{\text{enc}} \in \mathbb{R}^{m \times d}$ and $W_{\text{dec}} \in \mathbb{R}^{d \times m}$. We denote gradients as $\nabla_x L$ and define normalized vectors as $\hat{v} = \frac{v}{\|v\|_\epsilon}$ where $\|v\|_\epsilon = \sqrt{\|v\|_2^2 + \epsilon}$.

The "Rotation Trick" modifies gradient flow through non-differentiable operations by preserving gradient geometry at discrete selection steps. Unlike the STE, it preserves angular information at the non-differentiable step during backpropagation by normalizing pre- and post-selection vectors and applying a rotation-based correction term that maintains angular relationships.

Our proposed RT-TopKSAE consequently is a modification to the traditional top-k SAE that applies the rotation trick. Standard top-k SAEs apply a top-k mask, which introduces a non-differentiable point in the computation graph. To circumvent this for backpropagation, they surrogate the gradient through the straight-through estimator (Bengio et al., 2013). Although this method preserves

gradient magnitude, it completely loses vital information in gradient angles (Fifty et al., 2025). In particular, for an encoder vector e in d-dimensional space and $q = \mathrm{TopK}(e, k)$, information about the angle between q and $\nabla_q L$ is lost. The RT-TopKSAE preserves this angle while only returning the gradient of e through the non-differentiable point. We expect that maintaining this gradient angle information provides the encoder with a richer set of gradient signals from non-chosen features.

2.2 Rotation Trick Implementation

For an arbitrary vector e, the k largest elements of vector e ordered from highest to lowest are denoted by $q = \mathrm{TopK}(e, k)$. Each vector is then normalized using the ϵ-norm: $\hat{e} = \frac{e}{\|e\|_\epsilon}, \hat{q} = \frac{q}{\|q\|_\epsilon}$. The resulting two normalized vectors are combined into one unit vector: $r = \frac{\hat{e}+\hat{q}}{\|\hat{e}+\hat{q}\|_\epsilon}$. Using this new unit vector r, we can now define the rotated gradient of L with respect to e as $\nabla_e L = \frac{\|q\|_\epsilon}{\|e\|_\epsilon}[\nabla_q L - 2(r \cdot \nabla_q L)r + 2(\hat{q} \cdot \nabla_q L)\hat{e}]$, where $\cdot$ is the dot product. Consequently, the rotation trick preserves the angle relationship during backpropagation as follows: $\angle(q, \nabla_q L) = \angle(e, \nabla_e L)$. The algorithmic implementation is provided in Appendix A.

3 Results and Analyses

3.1 Overview of Results

Our results show a consistent trend of RT-TopKSAE outperforming TopKSAE across metrics related to representation quality, while incurring a slight loss in reconstruction performance. The most important finding is the complete utilization of the dictionary by RT-TopKSAE, and our other metrics indicate that this full usage does not limit the expressiveness of the learnt representations.

Category	Metric	RT-TopKSAE	TopKSAE	Improvement
Performance	Reconstruction Loss ↓	0.0010	**0.0006**	—
	CKA Stability ↑	0.88	**0.90**	—
Utilization	Dictionary Usage ↑	**100.0**	47.4	**2.1**×
	Gini Coefficient ↓	**0.0360**	0.0544	**1.5**×
Disentanglement	Jaccard Similarity ↓	**0.071**	0.431	**6.1**×

Table 1: **Comparison of Dictionary Utilization and Fidelity.** Results averaged across three seeds. Arrows (↑, ↓) indicate desired direction.

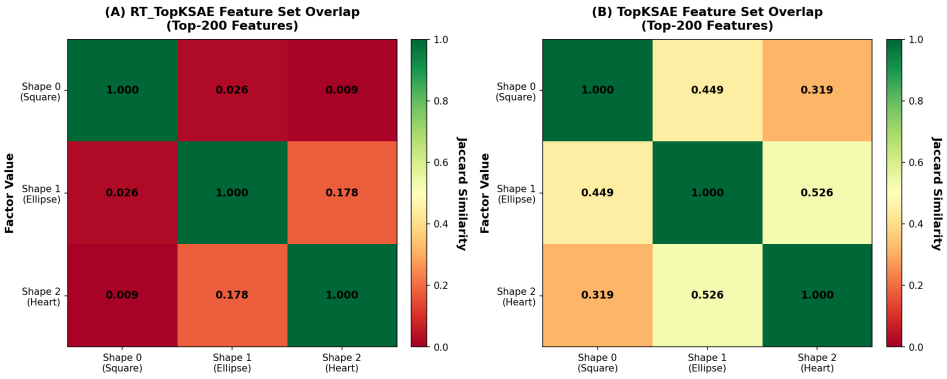

Figure 1: Jaccard Similarity Heatmap. Jaccard similarity measures feature overlap between shapes (lower = more specialized features). RT-TopKSAE achieves 6.1× lower overlap (0.071 vs 0.431), indicating greater monosemanticism where each shape activates distinct features rather than shared polysemantic ones.

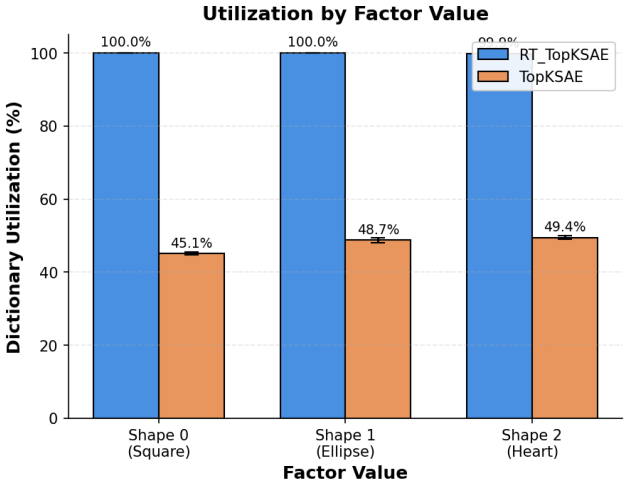

Figure 2: Dictionary Utilization Across Shape Factor Values. Bars represent percentage of dictionary features activated at least once for each shape category. RT-TopKSAE achieves near-complete utilization (100%) while TopKSAE shows ∼50% dead features that never activate.

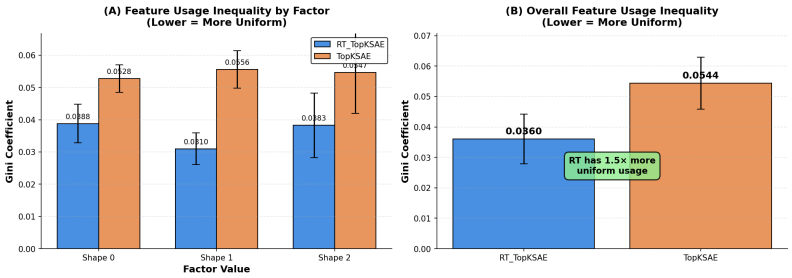

Figure 3: Gini Coefficient of Feature Activation Distribution. RT-TopKSAE achieves 1.5× more uniform activation distribution (0.036 vs 0.054), indicating all features contribute meaningfully rather than having a few dominant features.

3.2 Dictionary Utilization and Feature Disentanglement

As Figure 2 shows, RT-TopKSAE achieves 100% utilization, while TopKSAE at its highest achieves 47% utilization, meaning effectively half of the activations are dead during training. In isolation, these results don't mean that RT-TopKSAE is learning a richer representation; in fact, one could argue that its utilization is due to the backward pass adding noise to the dictionary, as a result of which the model may not be learning new, effective features, but random, unstable, meaningless features. Our other metrics, however, dismantle this argument.

Beginning with the Jaccard similarity (Figure 1), the most robust evidence supporting the opposite of this argument: a 6.1-times reduction is seen between RT-TopKSAE and TopKSAE, which strongly implies that the rotation trick negatively affects polysemanticism between features and promotes monosemanticism, contradicting the claim that the rotation trick adds noisy instability to the dictionary weights. The next metric that suggests our reconstructions are effective dictionary usages, not just monosemantic, is the Gini coefficient (Figure 3), which was 1.5 times lower on average between the two models, implying that the features are more uniformly utilized. Both models maintain similar CKA stability, indicating global representational fidelity is not compromised at the cost of utilization. All in all, the combination of these three metrics shows that the features are highly specialized to specific factors, a hallmark of disentangled representations.

4 Conclusions and Future Work

We introduced the RT-TopKSAE, which applies geometric constraints to gradient flow in Top-K sparse autoencoders by preserving angular relationships at the selection boundary. This simple modification resolves the persistent dead latent problem, achieving complete dictionary utilization while producing representations with substantially improved disentanglement and uniformity. Critically, these gains in geometric structure come with maintained representational fidelity, suggesting that gradient angle preservation acts as an effective geometric regularizer that guides optimization toward well-organized latent spaces. By ensuring that non-selected features receive directional feedback about how to adjust their orientations, the rotation trick enables the discovery of comprehensive, monosemantic dictionaries better suited for mechanistic interpretability.

The consistent success of gradient geometry preservation—from VQ-VAE codebooks to SAE dictionaries—suggests this principle may benefit other sparse coding methods where non-differentiable operations disrupt gradient flow. Our work demonstrates that incorporating geometric inductive biases can address fundamental organizational challenges in learnt representations. Future work should investigate whether improved dictionary structure translates to more interpretable features through direct visualization and steering experiments, and explore whether similar geometric constraints can mitigate identifiability issues in other sparse decomposition settings. As mechanistic interpretability increasingly relies on sparse autoencoders to reverse-engineer neural circuits, ensuring these tools produce geometrically structured, comprehensive dictionaries becomes essential for faithful interpretation.

REFERENCES

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. Identifying functionally important features with end-to-end sparse dictionary learning. *arXiv preprint arXiv:2405.12241*, 2024.

Christopher Fifty, Ronald Guenther Junkins, Dennis Duan, Aniketh Iyengar, Jerry Weihong Liu, Ehsan Amid, Sebastian Thrun, and Christopher Ré. Restructuring vector quantization with the rotation trick. In *International Conference on Learning Representations (ICLR)*, 2025.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations (ICLR)*, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 3519–3529, 2019.

Alireza Makhzani and Brendan J Frey. k-sparse autoencoders. 2014.

Loic Matthey, Irina Higgins, Christopher P Burgess, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. GitHub repository, 2017.

Adly Templeton, Tom Conerly, Jonathan Marcus, Joshua Batson, Nikhil Choudhury, et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Anthropic*, 2024.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

A Algorithmic Framework and Implementation Details.

Our architecture for RT-TopKSAE and TopKSAE are two untied sparse autoencoders, with inputs being d-dimensional vectors. Weight matrices are initialized as $W_{\text{enc}} \in \mathbb{R}^{d_{\text{hidden}} \times d_{\text{feature}}}$ and $W_{\text{dec}} = W_{\text{enc}}^T$. The former maps the input vector to an overcomplete latent vector; subsequently, a top-k sparsity constraint is applied to that vector. At that point is where a custom autograd function applies the rotation trick for RT-TopKSAE. For our specific experiments, the input was the second hidden layer of an MLP trained on dSprites (Matthey et al., 2017) with MNIST. Both SAEs were tasked with reconstructing the input across three random seeds (17, 42, 123) for 30 epochs using the Adam optimizer (Kingma & Ba, 2015).

Algorithm 1 RT-TopKSAE Algorithm

Forward Pass:
$\mathbf{e} = (\mathbf{h} - \mathbf{b}_{dec})\mathbf{W}_{enc} + \mathbf{b}_{enc}$
$M = (\mathbf{e} \in \text{top-}k(\mathbf{e}))$
$\mathbf{f} = M \odot \mathbf{e}$
$\hat{\mathbf{h}} = \mathbf{f}\mathbf{W}_{dec} + \mathbf{b}_{dec}$
Backward Pass:
$\hat{\mathbf{e}} = \frac{\mathbf{e}}{\|\mathbf{e}\|+\epsilon}, \hat{\mathbf{f}} = \frac{\mathbf{f}}{\|\mathbf{f}\|+\epsilon}, \mathbf{v} = \frac{\hat{\mathbf{e}}+\hat{\mathbf{f}}}{\|\hat{\mathbf{e}}+\hat{\mathbf{f}}\|+\epsilon}$
$\mathbf{g}_{rot} = \nabla_{\mathbf{f}}\mathcal{L} - 2(\mathbf{v}^\top \nabla_{\mathbf{f}}\mathcal{L})\mathbf{v} + 2(\hat{\mathbf{e}}^\top \nabla_{\mathbf{f}}\mathcal{L})\hat{\mathbf{f}}$
$\lambda = \frac{\|\mathbf{f}\|+\epsilon}{\|\mathbf{e}\|+\epsilon}$
$\nabla_{\mathbf{e}}\mathcal{L} = \lambda \mathbf{g}_{rot}$

B Experimental Setup

To evaluate how the rotation trick affected the quality and health of learnt representations in top-k sparse autoencoders, we tested RT-SAE across four metrics; utilization, Jaccard similarity, Gini coefficient, and CKA; against a baseline top-k SAE model. Our 3-layer MLP was trained on dSprites (Matthey et al., 2017), model weights were frozen after 10 epochs, and the second hidden layer of the MLP was passed as input to both SAEs, training them across three seeds (17, 42, 123).

B.1 Datasets

The dataset we evaluated on was dSprites (Matthey et al., 2017), which notably has 3 latent factors relevant to disentanglement research: shape, scale, and orientation. For factor-wise evaluation, we created factor-conditioned subsets of the dSprites dataset. Each subset contains data points with a fixed value for one factor and freely varying values for all other factors. For a given factor f and a value v, we chose images that satisfied $f = v$. This resulted in disjoint subsets corresponding to different values of the same factor.

B.2 MLP Implementation Details

A 3-layered Multi-Layer Perceptron (MLP) Auto-Encoder ($4096 \to 2048 \to 2048 \to 4096$) with ReLU activations that was trained using the Adam optimizer (Kingma & Ba, 2015) for 10 epochs at a learning rate of 1×10^{-3}. The activation weights from the training process were then frozen to provide constant input to the SAE regardless of the seed used during initialization. Both SAE variants shared the same architectural details and used a top-k value of 16. We varied only the application of the custom autograd function, which was applied to the RT-TopKSAE at the top-k masking.

B.3 Baselines

We evaluate how RT-TopKSAE differs from a standard version of TopKSAE. The two architectures are otherwise identical in terms of architecture except for the backward pass: the baseline uses the standard straight-through estimator (Bengio et al., 2013), while our method applies the rotation trick.

The same configurations were used for both models: 8192 features, Top-K sparsity (with $K = 16$), separately learnt encoder and decoder weight matrices, unit-normed decoder column vectors, and Adam optimization with a learning rate of 3×10^{-4}.

B.4 Evaluation Metrics

We evaluated the learnt sparse representations using five metrics to understand how the rotation trick affects the latent space. Each metric was chosen to highlight one aspect of the rotation trick we believe is relevant to its impact on the latent representations.

B.4.1 Dictionary Utilization

Our first metric is the dictionary utilization, which measures the percentage of dictionary features activated at least once across all samples in a given data slice. This metric measures the extent to which the model utilizes the codebook, as greater usage of the codebook directly indicates more efficient or diverse representation. We defined the dictionary utilization as utilization_pct $= \frac{\text{number of features with activation_rate}>0}{\text{total_features}} \times 100\%$, where activation_rate$[i] = \frac{\text{number of samples activating feature } i}{\text{total_samples}}$. However, dictionary utilization does not tell a full story; a high dictionary utilization can show more features are being used by the model, but it doesn't tell us if the features are particularly useful and, more importantly, if the additional features are just noise.

B.4.2 Jaccard Similarity

To circumvent this, we examined whether the features activated for different shapes are shared or specialized. This was evaluated via Jaccard Similarity, a set-based measure of overlap between the top-200 features for different data slices. A lower similarity can suggest more distinct features across different factors, greater disentanglement, and less reliance on shared, polysemantic features. Mathematically, the Jaccard similarity is simply defined as $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$.

B.4.3 Gini Coefficient

The Jaccard similarity shows how features overlap with each other, but this raises another question, especially in interpretability, where sparse autoencoders are used; we don't just want disentangled feature sets but also feature sets with a uniform distribution, allowing for a clearer and more interpretable representation. We decided to investigate this property with the Gini constant, which we used to measure inequality in continuous feature activation distributions. Let $x = (x_1, x_2, \ldots, x_n)$ be the activation rates of features, sorted in ascending order with zeros excluded, and let n be the number of features with an activation rate > 0. The Gini coefficient is defined as $G = \frac{2\sum_{i=1}^{n} i x_i}{n \sum_{i=1}^{n} x_i} - \frac{n+1}{n}$, which lies in the interval $[0, 1]$. Intuitively, G measures the inequality of feature activations: the lower bound of the function at 0 indicates a uniform distribution of features, while a value closer to 1 indicates that a few features dominate the activation.

B.4.4 Centered Kernel Alignment (CKA)

So far, all of the metrics have measured how features are activated across different data subsets, but they do not provide insight into the fundamental geometry of the features or whether these patterns are consistent across seeds, which is particularly important in interpretability, as ideally, we would need consistency to understand features better. Given the unordered permutation of top-K features, we reasoned that metrics such as cosine similarity would be insufficient to properly evaluate the similarity of outputs across different seeds. Therefore, we used Centered Kernel Alignment (CKA) (Kornblith et al., 2019), which measures the similarity between representations in a permutation-invariant manner. The CKA is defined in the range $[0, 1]$, where a value of 1 corresponds to identical patterns, while a value of 0 indicates completely unrelated representations.

B.5 Reconstruction Performance

A key limitation is the slightly higher reconstruction loss for RT-TopKSAE. However, as Figure 4 shows, the reconstructed outputs are visually similar between the two models, which supports the idea that the small increase in loss does not directly mean a lower reconstruction quality.

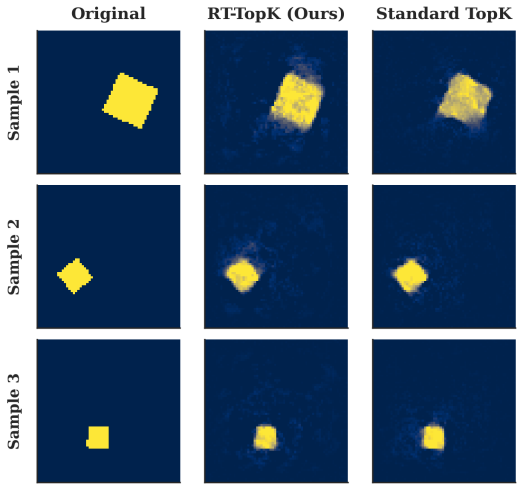

Figure 4: Comparison between the quality of original images and the visual reconstructions of RT-TopKSAE and TopKSAE on the dSprites Dataset. Both models produce visually similar outputs despite RT-TopKSAE's slightly higher numerical reconstruction loss, suggesting the quantitative difference doesn't translate to perceptual degradation.

Acknowledgments

This work was conducted under Algoverse AI Research. We used Claude for assistance with code development and debugging. All experimental design, methodology, analysis, and interpretation are the authors' own work.