Analyzing Representational Shifts in Multimodal Models: A Study of Feature Dynamics in Gemma and PaliGemma

Aaron Friedman **Trinabh Gupta** **Raine Ma**
Cole Blondin* Sean O'Brien* Kevin Zhu*

Abstract

Understanding internal representational shifts that occur from the adaptation of large language models (LLMs) to vision-language models (VLMs) provides insight into trade-offs in model interpretability, feature reuse, and task specialization. This paper presents an empirical study on representational shifts that occur when extending the LLM Gemma2-2B into its multimodal successor, PaliGemma2-3B. Our initial performance analysis reveals that sparse autoencoders (SAEs) trained on Gemma struggle to reconstruct PaliGemma's activations, motivating a deeper investigation into its activation patterns. Across 26 layers, 37% of SAE features show reduced activation in PaliGemma relative to Gemma. Further experiments on CIFAR-100 and TruthfulQA reveal that PaliGemma relies heavily on visual inputs, activating substantially fewer features for text alone. Additional analyses—including *Residual Stream SAE Performance Analysis*, *Activation Frequency and Dead Feature Quantification*, *Cross-Modal Feature Activity Patterns*, and *Semantic Robustness under Label Perturbations*—provide consistent evidence that PaliGemma's internal representations are more visually grounded and less aligned with purely textual features. Our findings suggest key representational trade-offs in feature dynamics when transitioning from unimodal to multimodal models.

1 Introduction

Artificial Intelligence (AI) systems have rapidly progressed in multimodal tasks such as image captioning and visual question answering (Radford et al., 2021; Alayrac et al., 2022; Chen et al., 2022). Vision-language models (VLMs) extend large language models (LLMs) with vision encoders and cross-modal fusion layers (Li et al., 2023; Wang et al., 2023), but little is known about how internal representations shift during this adaptation. While prior work has focused on improving the performance of VLMs (Radford et al., 2021; Alayrac et al., 2022), few studies have examined how feature representations change when LLMs are adapted to VLMs. While SAEs have uncovered interpretable features in LLMs (Chaudhary & Geiger, 2024), their behavior in VLMs remains understudied. We address this gap by analyzing representational changes as Gemma2-2B is adapted into PaliGemma2-3B. Using SAEs trained on Gemma, we compare both models on CIFAR-100 (Krizhevsky, 2009) and TruthfulQA (Lin et al., 2022), examining reconstruction loss, activation patterns, layer-wise similarity, and robustness under noisy supervision and mismatched labels. Our results reveal substantial shifts in feature dynamics and task behavior across modalities.

2 Related Works

Recent work has focused on adapting large language models (LLMs) into vision-language models (VLMs) by incorporating visual processing capabilities. Studies such as Radford et al. (2021), Alayrac et al. (2022), and Li et al. (2023) have demonstrated that integrating vision components into LLM architectures can significantly improve task performance on multimodal benchmarks. However, these works primarily evaluate end-task outcomes and

*Senior Author

do not investigate the internal representational changes or feature shifts that occur during this adaptation process. Our work addresses this gap by directly analyzing feature dynamics within the residual stream when LLMs are converted into VLMs. Sparse autoencoders (SAEs) have emerged as a promising tool for interpretability by learning compressed, sparse representations of internal model activations. By enforcing sparsity, SAEs encourage the discovery of monosemantic features that are easier to analyze. Huben et al. (2024) leveraged SAEs to reveal interpretable features in LLMs, but their work did not examine how these internal features shift in multimodal settings. Our study extends this line of research by applying SAEs to both text-only and vision-language models, quantifying cross-modal feature reuse, specialization, and alignment. While prior research on models like CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) has explored cross-modal embedding alignment, these methods primarily operate at the output level. In contrast, we analyze sparse internal activations, offering a more granular view of how feature usage patterns change across modalities. Our work complements these embedding-level studies by providing a feature-level perspective on multimodal representation learning.

3 Methodology

Our computational and implementation setup is detailed in Appendix A.3.

3.1 Evaluating SAE Generalization Across Models

To evaluate whether an SAE trained on one model can interpret another, we compared reconstruction loss using matched residual activations. Two SAEs with identical architecture were trained separately: one on Gemma's residuals, the other on PaliGemma's using text-only inputs. We then passed shared text inputs through PaliGemma and fed the resulting residuals into both SAEs. Reconstruction loss was measured via mean squared error between original and reconstructed residuals. This experiement tests whether the Gemma-trained SAE generalizes well enough to serve as a diagnostic tool for PaliGemma's representations. A small loss gap would suggest that the Gemma-trained SAE generalizes sufficiently well to be used as a diagnostic lens across both models. We list our SAE hyperparamters used for training in A.4.

3.2 Residual Stream SAE Performance Analysis

To quantify SAE performance, we compare reconstruction, sparsity, and total losses across all 26 layers of both Gemma and PaliGemma. We process inputs in batches of 16 to optimize memory usage.

3.2.1 Loss Metrics

Reconstruction Loss. We use Mean Squared Error to measure the difference between original activations and SAE reconstructions.

Sparsity Loss. The sparsity penalty ensures that most neurons remain inactive, encouraging monosemanticity. z_i represents the SAE's latent activations, and $\|z_i\|_0$ denotes the L_0 norm, and $\lambda = 10^{-3}$:

$$\text{Sparsity} = \lambda \sum_{i=1}^{N} \|z_i\|_0. \tag{1}$$

Total Loss. The final loss function balances reconstruction and sparsity losses with a regularization weight α:

$$\mathcal{L} = \text{MSE} + \alpha\, \text{Sparsity}. \tag{2}$$

Delta Loss. Delta Loss measures the change in model performance after SAE-based intervention. It is defined as the difference between the patched loss—computed after injecting

SAE-reconstructed activations—and the original (base) loss on the same input. A higher Delta Loss indicates greater disruption to the model's behavior caused by the intervention:

$$\Delta L = L_{\text{patched}} - L_{\text{base}}. \tag{3}$$

3.2.2 *Feature Extraction Pipeline*

We register forward hooks on model layers, then use SAEs to encode activations and calculate reconstruction losses. We then compute and analyze losses for each transformer layer, storing layer-wise losses and comparisons in structured datasets.

3.3 Activation Frequency and Dead Feature Quantification

We evaluated SAE feature activations by running questions from the TruthfulQA dataset on Gemma and PaliGemma. For each model, we aggregated the total activation counts of all SAE features across the entire dataset, layer by layer. This resulted in two datasets, one per model, containing the global count of activations for each SAE feature (indexed by `Layer` and `Feature Index`). We merged these datasets on shared SAE identifiers to enable feature-wise comparison. For each feature, we computed the relative activation frequency (RAF), where $\epsilon = 1 \times 10^{-9}$ prevents division by zero:

$$\text{RAF} = \frac{\text{PaliGemma Count}}{\text{Gemma Count} + \epsilon} \tag{4}$$

Features with zero PaliGemma activations were labeled *dead*, and those with RAF < 0.1 were labeled *low-activation*. To visualize trends, we sampled SAE layers 0, 5, 10, 15, and 20—spanning early to late depths—and plotted RAF-sorted features on a log scale to reveal differences in activation dynamics. All 16,384 features were included per layer, padding missing entries with RAF = 0. Layers were selected arbitrarily to cover early, middle, and late model depth, and not performance-driven.

3.4 Cross-Modal Feature Activity Patterns

3.4.1 *Global Feature Activation Trends (Layer-wise Counts)*

To analyze the distribution of SAE activations across different modalities and models, we conducted experiments on the CIFAR-100 dataset using three setups: Gemma using CIFAR Labels, PaliGemma using CIFAR Labels, and PaliGemma using CIFAR Images corresponding to labels used in the other models. For each CIFAR-100 class, either the class name or the corresponding image was passed into the model, and we captured residual activations at every transformer layer. Each captured residual stream was then encoded using pre-trained `Gemma-Scope` SAE models, yielding a latent representation $z \in \mathbb{R}^d$ per layer, where d is the SAE latent dimension. To determine active features, we applied a thresholding rule:

$$\text{Active}(z_i) = \begin{cases} 1 & \text{if } |z_i| > 0.1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

For each layer l, we computed the total count of active features by summing over all SAE latent dimensions:

$$A_l = \sum_{i=1}^{d} \text{Active}(z_i) \tag{6}$$

We repeat this across all layers l and for all classes and inputs of CIFAR-100. We then aggregated the activation counts to visualize layer-wise activation trends across the three conditions. The resulting trends reveal how feature activations vary across modalities.

3.4.2 *Top-Activated SAE Features Across Modalities*

For each modality (Gemma–Text, Pali–Text, Pali–Image), we identified the top 10 most frequently activated SAE features by counting how often each feature appeared across the dataset. Features were ranked by activation frequency and compared across modalities to assess overlap and divergence.

3.4.3 *SAE Representation Alignment via Correlation*

To assess cross-modal and cross-model representational similarity, we computed vector correlations between SAE-encoded residual activations a shared concept. We selected the label *apple* and extracted the corresponding CIFAR-100 image. This label was processed through (1) Gemma on text, (2) PaliGemma on text, and (3) PaliGemma on image. Residuals were extracted at transformer layer 12 and encoded using a pretrained SAE from the `Gemma-Scope` suite. To quantify alignment between representations, we computed cosine similarity and Pearson correlation for Gemma–text vs. Pali–text, Gemma–text vs. Pali–image, and Pali–text vs. Pali–image. This approach complements activation count comparisons by measuring directional alignment in SAE space rather than feature frequency alone.

3.4.4 *t-SNE Visualization of SAE Activations*

To qualitatively assess modality-driven representational shifts, we used t-SNE to visualize SAE-encoded activations from Gemma text, PaliGemma text, and PaliGemma image inputs at layers 8, 12, and 16. Full preprocessing and projection details are provided in Appendix A.2.

3.5 Semantic Robustness under Label Perturbations

To investigate how image-label alignment affects SAE feature activations, we analyzed PaliGemma under paired image-text input conditions. Using CIFAR-100 images of two distinct classes (e.g., *apple* and *fox*), we constructed matched and mismatched pairs (e.g., "apple" image with "apple" or "fox" label) and processed the inputs through PaliGemma's multimodal encoder. Residual activations were extracted at every transformer layer within PaliGemma's text-vision fusion architecture. Each layer's activations were encoded into SAE latent vectors $z \in \mathbb{R}^d$ using pre-trained SAEs. We applied a sparsity threshold of $\epsilon = 0.1$ to determine active SAE features:

$$\text{Active}(z_i) = \begin{cases} 1 & \text{if } |z_i| > 0.1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

For each pairing condition (e.g., *apple image + apple label* or *fox image + fox label*), we computed the total number of active SAE features per layer. We also identified the specific activated feature indices to compare how matching vs. non-matching pairs influence the selection of particular SAE features in PaliGemma's encoder.

4 Results and Analysis

4.1 Evaluating SAE Generalization Across Models

We evaluate the reconstruction quality of two SAEs—one trained on Gemma and one trained on PaliGemma—by measuring their performance on residual activations from PaliGemma in response to a shared text input. The reconstruction loss, computed as mean squared error (MSE), quantifies how well each SAE captures the underlying representation space.

As shown in Table 1, the SAE trained on Gemma achieves slightly lower reconstruction loss on PaliGemma residuals than the SAE trained directly on PaliGemma. This suggests that the Gemma-trained SAE generalizes well enough to serve as a reliable probe for analyzing PaliGemma, despite differences in training data or model architecture.

SAE Type	Trained On	Recon Loss (MSE)
Gemma SAE	Gemma	2.55
Pali SAE	PaliGemma	3.10

Table 1: Reconstruction loss on PaliGemma text residuals using Gemma- and Pali-trained SAEs.

4.2 Performance Analysis

Layer	Patching Loss		Delta Loss	
	G	PG	G	PG
0	3.81	1.13	0.05	0.06
5	5.44	1.05	1.69	-0.02
10	5.85	1.13	2.09	0.06
15	6.56	2.24	2.80	1.17
20	5.42	2.22	1.67	1.15
AVG	5.49	1.67	1.73	0.586

Table 2: Delta Loss (Patched - Baseline) for Gemma (G) and PaliGemma (PG). * AVG is across all layers.

Layer	Residual		MLP	
	G	PG	G	PG
0	0.155	0.29	1.11	0.64
5	1.10	1.61	0.62	0.66
10	1.59	1.95	1.85	1.19
15	2.84	5.00	2.72	2.64
20	6.14	9.26	5.90	4.83
AVG	3.70	6.33	4.18	4.14

Table 3: Reconstruction Loss for Gemma (G) and PaliGemma (PG). *AVG is across all layers.

Layer	Gemma	PaliGemma	Diff
0	934.06	792.05	142.01
5	512.22	114.05	398.17
10	303.25	116.26	187.00
15	325.96	61.10	264.86
20	406.22	37.65	368.56
AVG	458.10	172.49	285.60

Table 4: Residual Sparsity Loss for Gemma and PaliGemma. * AVG is across all layers.

Layer	Gemma	PaliGemma	Diff
0	120.05	105.22	14.83
5	98.72	75.11	23.61
10	87.31	65.04	22.27
15	77.89	50.33	27.56
20	70.45	45.98	24.47
AVG	90.08	68.34	21.74

Table 5: MLP Sparsity Loss for Gemma and PaliGemma. * AVG is across all layers.

PaliGemma exhibits consistently higher SAE residual reconstruction loss than Gemma's across observed layers, as reported in Table 3. However, MLP reconstruction remain comparable. The higher reconstruction loss in PaliGemma compared to Gemma indicates that the SAE trained on Gemma does not capture PaliGemma's residual activations as effectively, suggesting a divergence in representational structure between the two models. Despite lower sparsity loss in PaliGemma, as reported in Tables 4 and 5, this does not reflect increased sparsity in PaliGemma's representations. Rather, it indicates that the Gemma-trained SAE fails to reliably detect features in PaliGemma due to a shift in representation space, explaining both the reduced sparsity loss and higher residual reconstruction loss. While Table 2 shows lower Delta Loss in PaliGemma, interpretation is complicated by its lower baseline loss. Without normalization, the comparison does not clearly indicate whether PaliGemma is less affected by feature patching.

4.3 Activation Frequency and Dead Feature Quantification

Table 6 shows that PaliGemma maintains full feature coverage but suppresses a substantial portion of the feature space compared to Gemma, particularly in deeper layers where fewer features remain highly active. Figure 1 further demonstrates this by showing that PaliGemma concentrates activations on a small subset of features while suppressing most others. This shift suggests that PaliGemma relies on a narrower, potentially more specialized subset of features, likely shaped by its multimodal training. The over-activation of select

Metric	Value
Total SAE Features	283,107
Dead Features (0 activations in PaliGemma)	0 (0.00%)
Significantly Lower Activation Features	105,112 (37.13%)
Layers with Most Suppressed Features	Layers 10, 15, 20
Typical Activation Ratio (Suppressed Features)	< 0.1
Layers with Most Over-activated Features	Layers 10, 15, 20
Typical Activation Ratio (Over-activated Features)	> 10

Table 6: Feature suppression statistics for PaliGemma relative to Gemma.

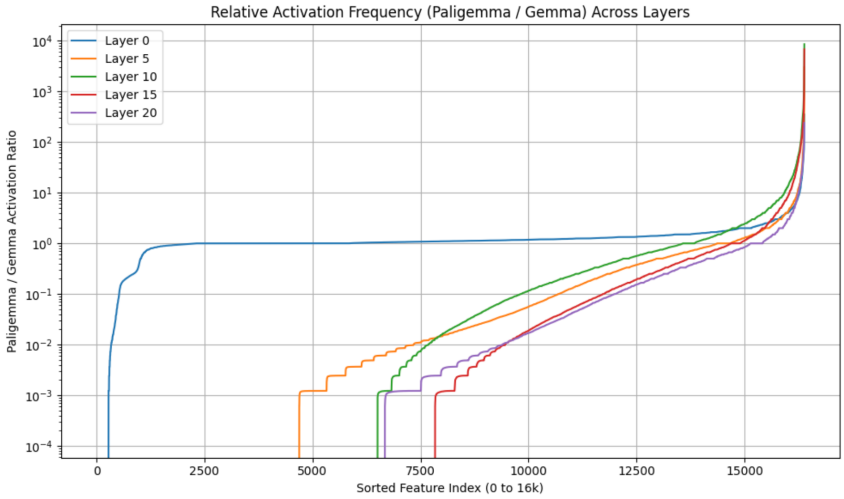


Figure 1: Relative Activation Frequency (PaliGemma / Gemma) across SAE layers.

features in deeper layers may indicate a redistribution of representational load, with certain features dominating due to cross-modal objectives. These results indicate that PaliGemma trades feature diversity for more selective activation, potentially improving alignment between vision and language inputs at the cost of representational breadth.

4.4 Cross-Modal Feature Activity Patterns

4.4.1 Global Feature Activation Trends (Layer-wise Counts)

Figure 2 shows that PaliGemma activates substantially fewer features on text-only labels compared to Gemma, but activates many more while processing images, particularly in deeper layers. This suggests that PaliGemma handles text and image inputs differently: remaining sparse and selective for text, while engaging a broader set of features for images. This behavior likely reflects PaliGemma's multimodal training, which encourages stronger reliance on visual patterns when they are available. Overall, PaliGemma appears biased towards image-heavy or multimodal tasks, allocating fewer features when processing text alone.

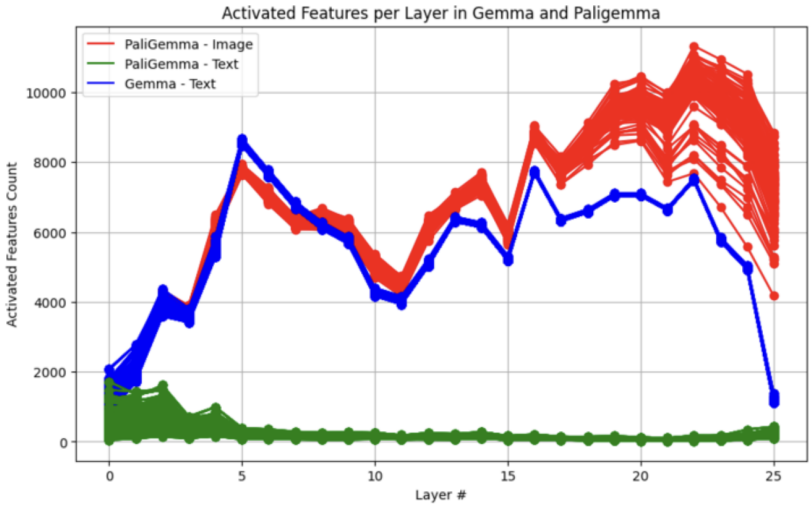


Figure 2: Activated SAE features per layer across three conditions on CIFAR100: Gemma on text labels (blue), PaliGemma on text labels (green), and PaliGemma on image inputs (red).

4.4.2 *Top-Activated SAE Features Across Modalities*

Modality	**Top Features**
Gemma Text	373, 9606, 4328, 8153, 12415, 209 2340, 9309, 10291, 16223
PaliGemma Text	15887, 5877, 15537, 14599, 4467, 1099 6810, 15030, 8578, 2234
PaliGemma Image	3020, 8072, 8920, 9746, 11794, 13886 16300, 5628, 5695, 6918

Table 7: Top-10 features per modality.

We observed substantial variation in the top-10 most frequently activated SAE features across modalities. For example, features most active in PaliGemma's vision encoder differ from those used by Gemma on text, despite semantically identical inputs. This suggests that each model-modality pair relies on a distinct subset of SAE features, reinforcing our claim that internal representations are not aligned, even when SAE structure and total activation volume appear consistent.

4.4.3 *SAE Representations Alignment via Correlation*

Input Pair	**Cosine**	**Pearson**
Gemma Text vs Pali Text	0.0538	0.0520
Gemma Text vs Pali Image	0.0076	0.0066
Pali Text vs Pali Image	0.0879	0.0860

Table 8: SAE similarity between model-modality pairs for the class *apple* at layer 12.

Table 8 quantifies representational similarity between different model-modality pairs for the concept *apple*. While Figure 2 suggests that PaliGemma on images and Gemma on text exhibit similar overall activation frequencies—implying comparable SAE usage—the correlation results reveal a different picture. All pairwise cosine and Pearson correlations are low, indicating that the specific features being activated are not aligned across models or modalities. This suggests that even when the volume of SAE activity appears similar (as

in Figure 2), the underlying representations remain semantically and structurally distinct. Notably, the strongest similarity occurs between PaliGemma's own text and image inputs, but even this correlation remains weak, underscoring a fundamental representational gap.

4.5 Semantic Robustness under Label Perturbations

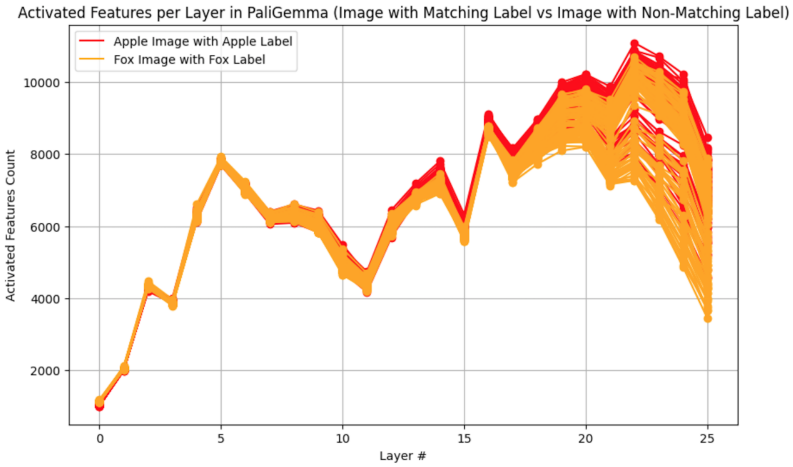
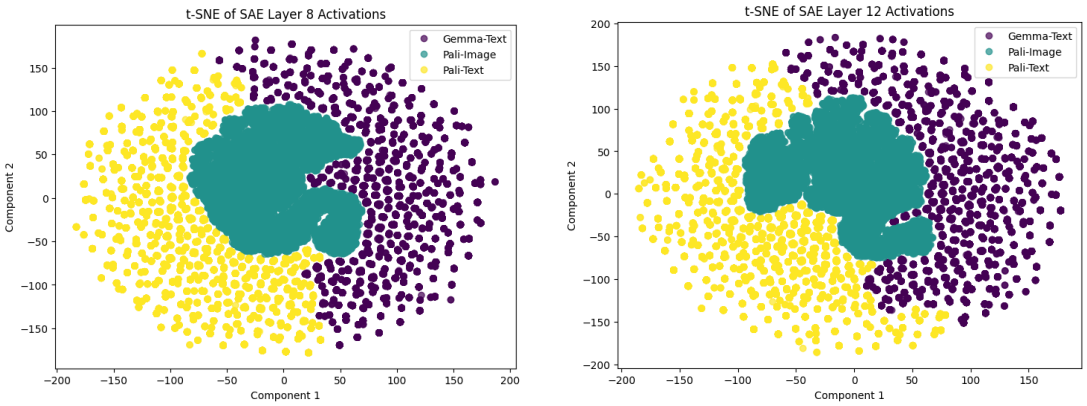


Figure 3: Activated SAE features per layer in PaliGemma when the image-label pair is correct (red) vs. when labels are randomly shuffled (orange).

Figure 3 shows that PaliGemma activates nearly the same number of SAE features per layer, regardless of whether the image-text pair is correct. This suggests that SAE activations are largely image-driven and insensitive to label correctness in terms of feature count. The near-identical activation patterns suggest that PaliGemma's SAE activations are overwhelmingly driven by the image modality. The text label, even when mismatched, does not strongly influence feature activation count at each layer. This could mean that PaliGemma processes image and text inputs in parallel, with the image stream dominating the SAE feature space, while the text has a weaker or more downstream influence. The slight divergence in deeper layers may reflect cross-modal interaction where some late-stage SAE activations adjust based on the label. However, this adjustment appears small, suggesting that much of the cross-modal reasoning happens after SAE processing (e.g., in the MLP head or multimodal fusion modules). Overall, this result reinforces the idea that the SAE primarily encodes image-driven representations and that feature sparsity is not highly sensitive to label correctness at the activation count level.

4.5 t-SNE Visualization of SAE Activations



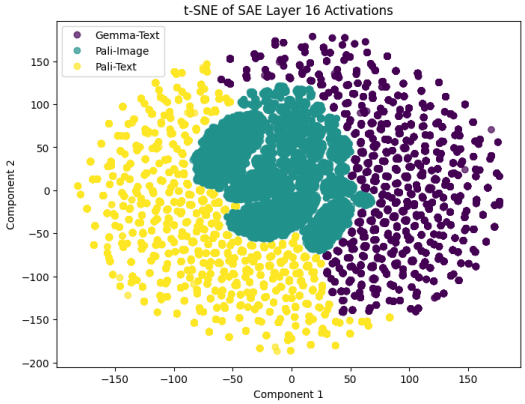


Figure 4: t-SNE projections of SAE activations. Top row: Layer 8 (left) and Layer 12 (right). Bottom row: Layer 16. Each point is a latent vector from a Gemma-trained SAE, colored by input modality (Gemma-Text, Pali-Text, Pali-Image).

We visualize SAE-encoded latent space using t-SNE at layers 8, 12, and 16. The resulting clusters are separable by modality: PaliGemma's image-based activations form tight clusters, while text-based activations from both PaliGemma and Gemma occupy broader, distinct regions. Notably, PaliGemma's image and text representations remain disjoint, indicating minimal feature reuse. This supports the hypothesis that PaliGemma encodes inputs in separate subspaces rather than a shared representation. These results suggest that VLMs represent visual input in a semantically abstract space overlapping both textual modalities, offering a potential mechanism for cross-modal alignment via shared latent representations.

5 Limitations and Future Work

While our analysis reveals clear representational shifts between Gemma and PaliGemma, our inputs are limited to single-token text labels and CIFAR-100 images, which do not capture the complexity of natural language prompts or multimodal reasoning. Our experiments also focus on two datasets—CIFAR-100 and TruthfulQA—which may not generalize to tasks like captioning or Visual Question Answering (VQA; (Antol et al., 2015)). Future work will explore training SAEs directly on PaliGemma's image and joint activations, interpret high-activation features using tools like Neuronpedia (Neuronpedia Contributors, 2024), and evaluate longer or compositional prompts. We also plan to investigate cross-layer alignment and use causal patching to trace the influence of specific SAE features on model behavior.

6 Conclusion

In this work, we investigate the internal feature shifts that occur when LLMs are adapted into VLMs. By using SAEs, we explore how feature representations evolve during the transition from the text-only Gemma2-2B to the VLM PaliGemma2-3B. Our findings reveal substantial differences in how internal representations are structured across models and modalities. PaliGemma exhibits higher reconstruction loss, lower activation frequency for many features, and a shift toward modality-specific feature usage. For example, PaliGemma activates fewer SAE features on text than Gemma, and its top features differ completely between text and image inputs, even for the same concept. We also observe that PaliGemma concentrates activations in a smaller subset of SAE layers and features, and shows reduced sensitivity to text-label mismatches at the activation level. These results suggest a fundamental reorganization of internal representations during multimodal adaptation. Our analysis shows that SAEs can capture modality-specific structural shifts, offering a scalable tool for probing the internals of vision-language models.

References

Jean-Baptiste Alayrac, Jeff Donahue, Paolo Lucarella, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2204.14198.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2425–2433, 2015.

Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. Saelens. https://github.com/jbloomAus/SAELens, 2024.

Maheep Chaudhary and Atticus Geiger. Evaluating open-source sparse autoencoders on disentangling factual knowledge in gpt-2 small. *arXiv preprint arXiv:2409.04478*, 2024. URL https://doi.org/10.48550/arXiv.2409.04478.

Yen-Chun Chen, Chenfei Luo, Zhe Hu, et al. Pali: A jointly-scaled multilingual language and vision model. In *arXiv preprint arXiv:2209.06794*, 2022. URL https://arxiv.org/abs/2209.06794.

Bryan Huben et al. Sparse autoencoders discover interpretable features in large language models, 2024. arXiv preprint arXiv:2403.00001.

Chao Jia, Yinfei Yang, et al. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Junnan Li, Dongxu Hu, et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Stephanie Lin, Jacob Hilton, and Amanda Askell. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2022. URL https://arxiv.org/abs/2109.07958.

Neuronpedia Contributors. Neuronpedia: An open atlas of interpretable neurons in language models. https://www.neuronpedia.org, 2024. Accessed: 2024-05-18.

Paszke et al. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Alec Radford, Jong Wook Kim, M Hallacy, Aditya Ramesh, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. URL https://arxiv.org/abs/2103.00020.

Zhiwei Wang, Licheng Xie, Xiaohua Chen, et al. Image as a foreign language: Beit-3 pretraining for multimodal understanding and generation. *arXiv preprint arXiv:2301.06216*, 2023. URL https://arxiv.org/abs/2301.06216.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

A Appendix

A.1 Model and Autoencoder Background

A.1.1 Sparse Autoencoders (SAEs)

Sparse autoencoders (SAEs) are unsupervised neural networks trained to compress and reconstruct input data while enforcing sparsity in the latent space. In our context, SAEs are applied to the residual stream activations of transformer-based language models.

Each SAE consists of:

- An encoder that maps a high-dimensional activation vector $\mathbf{a} \in \mathbb{R}^n$ to a sparse latent representation $\mathbf{z} \in \mathbb{R}^d$.
- A decoder that reconstructs the input as $\hat{\mathbf{a}} = f_{\text{dec}}(\mathbf{z})$.

Sparsity is enforced using a regularization penalty (e.g., L_0 or L_1 norm) to encourage monosemantic, interpretable features. All SAEs in this work are trained using the `sae-lens` framework (Huben et al., 2024).

A.1.2 Gemma and PaliGemma Architectures

Gemma 2-2B is a 2-billion-parameter decoder-only transformer model released by Google DeepMind. It contains standard components such as multi-head self-attention, feedforward MLPs, and residual connections.

PaliGemma 2-3B builds upon Gemma by incorporating a vision encoder and multimodal training objectives. It supports both text and image inputs through a fusion architecture that integrates visual and linguistic signals at various layers. Despite structural similarities, PaliGemma's internal activations differ substantially from Gemma due to its multimodal training regime.

In this paper, we train SAEs on residual activations from Gemma and apply them to both Gemma and PaliGemma to examine representational shifts resulting from multimodal adaptation.

A.2 Elaboration on t-SNE Methodology

To investigate representational divergence across unimodal and multimodal contexts, we used t-distributed stochastic neighbor embedding (t-SNE) to project SAE latent activations into two dimensions. We extract hidden states from Gemma text-only prompts, PaliGemma text prompts, and PaliGemma image inputs at SAE layers 8, 12, and 16. These hidden states were passed through pre-trained SAEs to obtain bottleneck activations, which serve as compressed representations of the model's internal features.

For each input modality:

1. We obtained the hidden state $h \in \mathbb{R}^d$ from the target transformer layer.
2. The hidden state was optionally padded or truncated to match the SAE input dimension d', and passed through the encoder:
$$z = \text{ReLU}(W_{\text{enc}} h + b_{\text{enc}}), \quad z \in \mathbb{R}^k \tag{8}$$
3. We collected z vectors from each modality and visualized their distribution using t-SNE with default parameters (perplexity = 30, random seed = 42).

To isolate the effect of modality, we used matched prompts (e.g., "A photo of a dog") and paired each image with its corresponding label for Gemma and PaliGemma text. The resulting 2D embeddings allow qualitative inspection of how SAEs separate or align representations across modalities. Clear clusters imply modality-specific encoding, while overlap suggests feature reuse or shared abstraction.

A.3 Computational Setup

All experiments are conducted on an NVIDIA A100 GPU. The implementation utilizes Hugging Face Transformers (Wolf et al., 2020), SAE-Lens (Bloom et al., 2024), and PyTorch (Paszke et al., 2019). This study involves no model training or sampling. All evaluations are conducted on fixed, pretrained model checkpoints without hyperparameter tuning. All analyses rely solely on publicly available model checkpoints and standard datasets, independent of any project-specific codebase.

A.4 SAE Hyperparameters

We set the SAE latent dimension to $d = 16{,}384$ (matching the model's residual width), used a sparsity weight $\lambda = 1 \times 10^{-5}$ and balance term $\alpha = 0.1$, and trained with Adam (learning rate 1×10^{-3}, batch size 16) for 50 epochs.