Towards Understanding Multimodal Fine-Tuning: A Case Study into Spatial Features

Anonymous Author(s)
Affiliation
Address
email

Abstract

Vision–Language Models (VLMs) demonstrate strong performance on a wide range of tasks by fine-tuning pretrained language backbones to process projected visual tokens alongside text. Yet despite these empirical gains, it remains unclear how language backbone representations adapt during multimodal training and when vision-specific capabilities emerge. In this work, we present the first mechanistic analysis of VLMs adaptation with stage-wise model diffing, a technique that isolates representational changes introduced during multimodal fine-tuning to reveal how a language model learns to "see". Concretely, we fine-tune sparse autoencoders trained on LLaMA-3.1-8B over multimodal activations from LLaVA-More (based on LLaMA-3.1-8B) using 50k VQAv2 pairs. We first isolate vision-preferring features that appear or reorient during multimodal fine-tuning. We then test for spatial selectivity using a controlled shift to spatial prompts to identify the attention heads that causally activate these units. Our findings show that stage-wise model diffing reveals when and how spatially grounded multimodal features arise. It also provides a clearer view of modality fusion by showing how visual grounding reshapes features that were previously text-only. This methodology enhances the interpretability of multimodal training and provides a foundation for understanding and refining how pretrained language backbones acquire vision-grounded capabilities.

1 Introduction

Large vision–language models (VLMs) have achieved strong performance on multimodal tasks, including visual question answering (VQA), image captioning, object detection, and visual grounding [28, 27, 1, 2, 11]. These gains are typically realized by fine-tuning pretrained language models to process visual inputs through projected token sequences, allowing for seamless fusion of image and text representations [49, 19, 51, 13, 12]. Yet we lack a mechanistic account of how language representations adapt during multimodal training and when vision-specific capabilities emerge [23, 45, 46, 42, 5].

In this work, we introduce a methodology for analyzing multimodal adaptation in VLMs through stage-wise model diffing [6]. This mechanistic interpretability technique isolates representational changes introduced during fine-tuning by comparing sparse autoencoder (SAE) dictionaries across training stages, models, or datasets. By tracking how individual features rotate, emerge, or are repurposed, stage-wise diffing has been shown to uncover subtle shifts such as sleeper-agent features [21, 6]. We extend this approach to the multimodal setting, presenting the first application of stage-wise model diffing to study how pretrained language features evolve under visual grounding.

Concretely, we fine-tune LLaMA-Scope SAEs on activations extracted from the LLaVA-More model [20] on 50k samples from the VQAv2 dataset [17]. This warm-start preserves the original feature basis

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

while adapting to multimodal activations. We isolate features that gain visual preference and undergo strong geometric rotation, serving as anchors for studying spatial representations in the backbone. To identify which adapted features encode spatial reasoning, we apply a controlled dataset shift from general VQA to spatial queries. Features that are preferentially recruited under spatial prompts form a selective subset, which we validate through automatic and manual interpretation. These features consistently activate on questions about object placement, relative position, and orientation.

Finally, we use attribution patching to trace the causal pathways by which these spatial features are activated. Our results reveal a sparse set of mid-to-deep layer heads that consistently drive spatial representations, often localizing to semantically meaningful regions and reappearing across related prompts. These findings support the hypothesis that a small number of specialized attention heads coordinate visual grounding within the model.

Our contributions are as follows:

- We propose stage-wise model diffing as a method for dissecting multimodal adaptation in large language models, and show that it isolates the emergence of vision-specific features within the backbone.
- We identify sparse SAE features that encode spatial relationships and are selectively activated by spatial prompts.
- We causally attribute these features to a small subset of attention heads using scalable patching methods.

By focusing on feature-level change, our approach complements high-level alignment analyses and probing-based methods, providing a deeper mechanistic view of how models "learn to see". More broadly, this work offers a framework for auditing and refining multimodal training regimes, with implications for safety-critical domains and targeted fine-tuning in specialized applications.

2 Related Work

Model Diffing and Representation Dynamics Model diffing techniques aim to isolate how internal representations change across models or training stages. Early work focused on coarse similarity measures, such as visualizing function-space geometry [37, 14], stitching intermediate layers across models [26, 3], or defining new similarity metrics [25, 4]. Later studies examined alignment at the level of individual neurons, showing convergent units across independently trained networks [29, 36].

Sparse autoencoders (SAEs) provided a feature-level lens, and Kissane et al. [24] showed that SAEs largely transfer between base and fine-tuned models—implying most features are preserved and only a minority are altered. This motivates methods that can isolate and interpret precisely those changes. Stage-wise model diffing [6] offers such fine-grained resolution, revealing sleeper-agent features and distinguishing between base and chat-tuned models [33].

Extensions to multimodal models highlight similar representational shifts: Khayatan et al. [23] proposed concept-shift vectors for steering, while Venhoff et al. [45] found vision-language alignment converges in middle-to-late layers. These remain semantic-level analyses, whereas our work applies stage-wise diffing with SAEs directly to the backbone, providing the first mechanistic account of how multimodal fine-tuning rotates features and induces spatially grounded representations within pretrained language models.

Multimodal Mechanistic Interpretability. Compared to the rapidly growing literature on mechanistic interpretability of textual LLMs, relatively few studies have examined the internal mechanisms of multimodal large language models (MLLMs). Existing work falls into two main categories.

First, tool-based or causal analyses aim to explain model behavior at a high level. Stan et al. [42] introduced an interpretability toolkit for VLMs based on attention patterns, relevancy maps, and causal interventions. Basu et al. [5] applied intervention methods to trace how information is stored and transferred, while Palit et al. [39] used causal mediation analysis to study how BLIP integrates visual evidence. Second, probing-based studies focus on the representations themselves. Tong et al. [44], Gandelsman et al. [15], and Chen et al. [8] analyzed CLIP, identifying both strengths and limitations. Schwettmann et al. [41] reported multimodal neurons responsive to joint visual–textual concepts, and Jiang et al. [22] examined how VLMs differentiate hallucinated from real objects.

More recent methods attempt to map visual embeddings into linguistic space, such as Neo et al. [35] who projected visual features onto language vocabulary, or Venhoff et al. [46] who studied the late emergence of visual signals in LLM backbones.

In contrast, these studies primarily analyze patterns, interventions, or probing correlations, but do not directly track how multimodal fine-tuning restructures the backbone's internal features. Our work addresses this gap by providing a mechanistic perspective.

3 Preliminaries

3.1 Vision–Language Models

A vision–language model (VLM) consists of three components: a visual encoder f_V, a pretrained language model f_{LM}, and a trainable projector P. The visual encoder (e.g., a ViT [40]) extracts image patch embeddings $V = f_V(x) = [v_1, \ldots, v_{N_V}]$, which the projector maps into the token space as $\tilde{V} = P(V)$. These projected image tokens are concatenated with tokenized text embeddings $T = [t_1, \ldots, t_{N_T}]$ to form the multimodal sequence $X = [\tilde{v}_1, \ldots, \tilde{v}_{N_V}, t_1, \ldots, t_{N_T}]$. Alignment between modalities is achieved through *visual instruction tuning*, where image–text pairs fine-tune the backbone to follow multimodal instructions.

The language model processes X through a stack of transformer layers, each consisting of multi-head self-attention (MHA) and a feed-forward network. For each head h, attention is computed as

$$\text{Attn}(Q, K, V) = \text{Softmax}\Big(\frac{QK^\top}{\sqrt{d_h}} + M\Big)V, \tag{1}$$

where M is the causal mask that prevents attending to future tokens. The outputs of all heads are concatenated and projected back into the hidden dimension, and the final hidden states are mapped through the unembedding matrix to yield next-token probabilities. For our experiments, we adopt LLaVA-More [9], which extends LLaVA framework [32, 31] by integrating recent language models and diverse visual backbones; specifically, we use the variant combining the CLIP ViT-Large-Patch14–336 encoder with a LLaMA-3.1-8B language model backbone [18].

3.2 Sparse Autoencoders (SAEs)

Sparse Autoencoders (SAEs) learn a dictionary of features that approximate hidden states as sparse linear combinations of interpretable directions. mitigating superposition where many features overlap in the same dimensions [7, 10]. Formally, a vanilla SAE encodes an input vector $x \in \mathbb{R}^D$ into a sparse hidden representation

$$f(x) = \text{ReLU}(W_{\text{enc}}x + b_{\text{enc}}), \quad W_{\text{enc}} \in \mathbb{R}^{F\times D}, ; b_{\text{enc}} \in \mathbb{R}^F, \tag{2}$$

which are then decoded back into the input space:

$$\hat{x} = W_{\text{dec}}f(x) + b_{\text{dec}}, \quad W_{\text{dec}} \in \mathbb{R}^{D\times F}, ; b_{\text{dec}} \in \mathbb{R}^D. \tag{3}$$

Sparsity is encouraged via an L_1 penalty on the hidden activations, yielding the objective

$$\mathcal{L} = |x - \hat{x}|2^2 + \lambda \sum i = 1^F |f_i(x)|_1. \tag{4}$$

Here, decoder columns $(W_{\text{dec}})_{:,i}$ define the direction of each feature in input space, while encoder rows $(W_{\text{enc}})_{i,:}$ act as detectors that determine when a feature is present.

Top-K Sparse Autoencoders. Top-K SAEs [16] enforce sparsity by keeping only the K most active hidden units per input and setting the rest to zero. The surviving activations are then decoded as in the vanilla SAE. This hard selection yields a sharper sparsity–fidelity tradeoff, reduces feature co-adaptation, and improves interpretability by ensuring that only a few features contribute to each reconstruction.

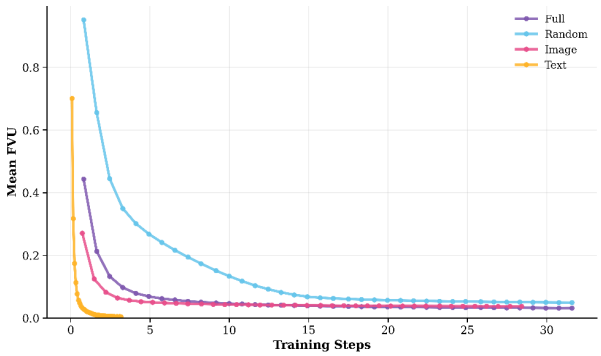
We build on the LLAMA-SCOPE suite of SAEs trained on LLaMA-3.1-8B [20], which refine the Top-K design with norm-aware selection [43], JumpReLU post-processing to stabilize the number of active features, and K-annealing during training. Since our VLM (LLaVA-More) shares the same backbone, we warm-start from these pretrained SAEs rather than retraining from scratch, enabling us to directly leverage millions of monosemantic features across layers.

4 Adapting Language Dictionaries to Vision-Language Space

To study how multimodal fine-tuning reshapes internal representations, we adapt sparse autoencoders (SAEs) of Llama 3.1 8B backbone to the hidden states of LLaVA-MORE (Llama 3.1 8B backbone) [9]. We use 50k image–question pairs from the VQAv2 dataset [17], a widely used benchmark for visual question answering that pairs natural images with open-ended queries. Each SAE is attached to the output of a transformer block and trained on cached activations from these samples. Images are represented by 575 consecutive visual tokens, and questions by variable-length text sequences; this separation allows token-type–specific masking.

We initialize SAEs from the pretrained `llama_scope_lxr_8x` release [20], re-instantiated as a top-k model (k=50), to preserve a meaningful basis while enabling sparse, interpretable codes. This warm-start ensures continuity with the pretrained language feature space. As a control, we also train SAEs from random initialization under identical conditions. Training uses Adam with a layer-scaled learning rate, and cached activations are processed in padded mini-batches. To disentangle modality-specific contributions, we consider four regimes: (i) full sequence, (ii) image-only, using only the visual-token span, (iii) text-only, using only the non-visual span, and (iv) random initialization. In all cases, the SAE receives the full hidden state sequence, but masking controls which token spans contribute to the training signal.

We evaluate reconstruction quality using the fraction of variance unexplained (FVU) and report sparsity to verify that codes remain selective. Evaluation is performed on a held-out split. Figure 1 shows FVU as a function of tokens seen across layers and masking regimes. Text-only SAEs converge rapidly, while image-only and full-token regimes converge slowly and plateau at higher error, reflecting the mismatch between projector embeddings and the LLM basis. Random initialization performs worst, underscoring the importance of starting from a pretrained language dictionary. These findings establish text-only SAEs as a reliable reconstruction baseline, which we later use for stage-wise diffing.



Metric	Full	Random	Image	Text
Mean	0.032	0.050	0.037	**0.005**
Std	0.028	0.041	0.027	**0.009**
Min	0.013	0.020	0.017	**0.000**
Max	0.123	0.198	0.123	**0.037**
Tokens (M)	31.6	31.6	28.4	3.2

Figure 1: **SAE adaptation on LLaVA-MORE.** Left: Mean fraction of variance unexplained (FVU) across layers on the validation set. Right: Summary statistics of FVU values on the validation set, with decimal alignment; the lowest mean is highlighted in **bold**.

Implications for stage-wise model diffing. Stage-wise diffing assumes that fine-tuning induces *localized* (feature-level) changes rather than wholesale rotations. Prior work reports that image-token representations in early layers exhibit higher reconstruction error than text tokens, indicating a distributional gap between projector outputs and the LLM basis [47]. Consistent with this, our decoder–cosine analysis (Appx.Fig.6) shows that *text-only* SAEs remain highly aligned to the base LLM dictionary across layers, whereas *image-only* and *full sequence* SAEs undergo large rotations in shallow layers and only align in later layers. We also note that text-only SAEs begin with slightly higher error in the very first layers but adapt extremely quickly, converging to near-zero reconstruction, while image and full-sequence SAEs plateau at higher error—underscoring the instability of projector-driven spans (see Appx.Fig.7). We therefore avoid stage-wise diffing on image-only or full-sequence SAEs in early layers, and focus on text-only SAEs and on later layers where alignment is stable and feature-level identifiability is more plausible.

5 Identifying Adapted Features

Our goal is to find SAE features that (i) exhibit a *modality preference* for vision input and (ii) reorient geometrically after multimodal adaptation. Such features are the best targets for stage-wise diffing and later causal probes.

5.1 Signals

Modality preference (variance gap). View each SAE feature f as a latent direction whose activation on hidden state x is $h_f(x)$. We quantify f's preference for vision states using the variance gap:

$$\Delta_f \;=\; \mathbb{E}_{\text{vision}}\big[h_f^2\big] \;-\; \mathbb{E}_{\text{text}}\big[h_f^2\big].$$

$\mathbb{E}_{\text{vision}}[\cdot]$ is taken from VQA runs of the VLM (image + question), while $\mathbb{E}_{\text{text}}[\cdot]$ comes from the base LLM on the same prompts, where images are replaced with captions, and the model receives only textual input. A large Δ_f indicates that f shows stronger activation on image-conditioned representations, suggesting visual specialization.

Geometric reorientation (decoder cosine). To test if f has been *repurposed* by multimodal fine-tuning, we compare its decoder direction before and after adaptation. Let $W_{\text{dec},f}^{\text{LLM}}$ be the base SAE decoder vector and $W_{\text{dec},f}^{\text{VLM}}$ the corresponding vector in the VLM-adapted SAE. We compute

$$c_f \;=\; \cos\!\big(W_{\text{dec},f}^{\text{LLM}},\, W_{\text{dec},f}^{\text{VLM}}\big).$$

High c_f means the semantic direction of f stayed aligned with the original language dictionary; low c_f indicates a substantial rotation, consistent with a reallocation of f to encode new multimodal structure. We use decoder vectors rather than encoder parameters because decoder directions more directly index the feature's semantics.

5.2 Selection Procedure

We identify adapted features using a *two-stage filter*. All features from every layer are pooled together, and thresholds are computed over this global set. Stage one retains the top $p_{\text{gap}} = 20\%$ of features by variance gap Δ_f, ensuring a preference for vision-conditioned activations. Stage two further narrows this pool to the bottom $p_{\cos} = 20\%$ by cosine similarity c_f, isolating those that underwent the strongest decoder rotations. This procedure produces a single globally defined adapted set comprising under 5% of all features. The joint distribution of variance gap and cosine similarity is shown in Fig. 2, with selected adapted features highlighted. Additional summaries, such as counts of adapted features per layer and their mean cosine similarities, are provided in Appx. Fig. 8a and Appx. Fig. 8b.

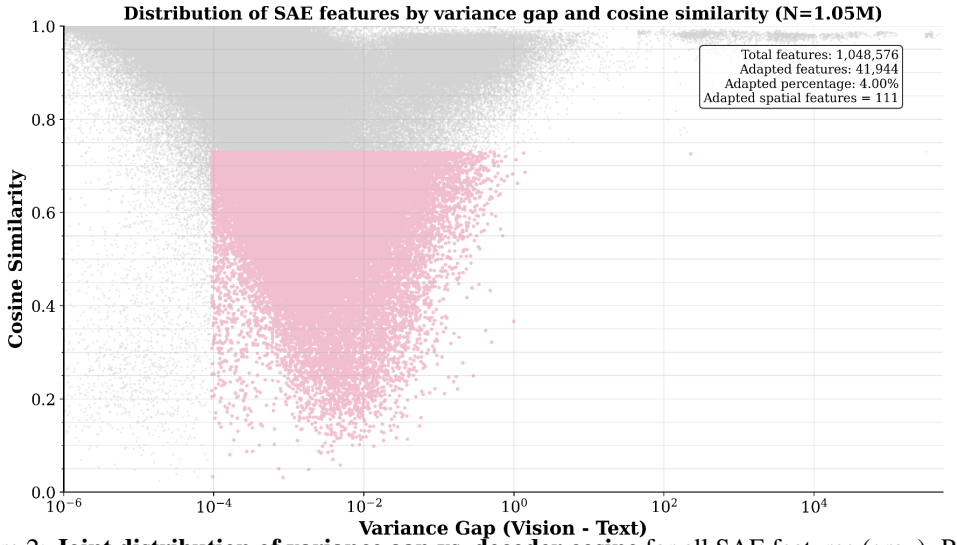


Figure 2: **Joint distribution of variance gap vs. decoder cosine** for all SAE features (gray). Points highlighted in pink are retained by our two-stage filter yielding the globally defined adapted set.

6 Case Study: Identifying Spatial Reasoning Features

We aim to isolate SAE features that encode spatial grounding by comparing firing patterns under a controlled dataset shift from general VQA to spatial queries.

Datasets. We consider two evaluation sets derived from VQAv2. The baseline is the full validation split, denoted $\mathcal{D}_{\text{base}}$. To induce a targeted shift, we construct a spatial subset $\mathcal{D}_{\text{sp}}$ by filtering questions that contain spatial cues (e.g., *left/right/above/behind*). This contrast tests whether some SAE features are selectively recruited under spatial reasoning.

Firing frequencies. Let $h_f(x_t) \geq 0$ denote the activation of feature f on token t of input x. For a dataset $\mathcal{D}$, the firing frequency of f is

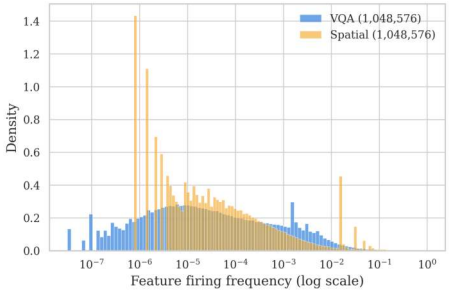
$$p_f(\mathcal{D}) \;=\; \frac{1}{n(\mathcal{D})} \sum_{x \in \mathcal{D}} \sum_{t} \mathbf{1}\{h_f(x_t) > 0\},$$

where $n(\mathcal{D})$ is the total number of tokens.

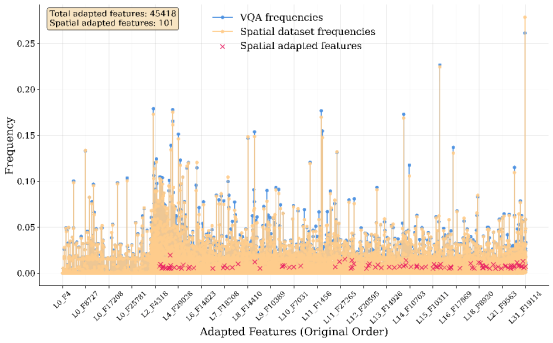
Distribution shift. Figure 3a compares the empirical distributions of feature firing frequencies under $\mathcal{D}_{\text{base}}$ and $\mathcal{D}_{\text{sp}}$. The spatial subset exhibits a heavier right tail, suggesting selective recruitment under spatial queries.

For each feature f, we compute the frequency gap $\Delta p_f \;=\; p_f(\mathcal{D}_{\text{sp}}) - p_f(\mathcal{D}_{\text{base}})$, and the odds ratio OR_f comparing firing counts across the two splits. Features with large Δp_f and $\text{OR}_f > 1$ are flagged as spatial *candidates*.

Selection and outcome. From the spatial candidates, we retain only those that also lie in the adapted set $\mathcal{A}$ from Sec. 5, ensuring they both reorient under multimodal fine-tuning and respond to spatial distribution shifts (with scatter plot shown in Appx.Fig.9). To remove prompt-lexical artifacts, we further probe with a neutral instructions such as *"Describe the positions of all objects in the image."* and Features that remain active and image-token–dominant are preserved, while prompt-specific units are discarded. Figure 3 visualizes the result: across all adapted features, it compares firing frequencies on $\mathcal{D}_{\text{base}}$ and $\mathcal{D}_{\text{sp}}$, highlighting the subset that survives this full pipeline. The plot shows that adapted features span a wide dynamic range, with the retained spatial set concentrated in the high-frequency tail under $\mathcal{D}_{\text{sp}}$.



(a) Overall distribution shift in feature firing frequencies when moving from generic VQA to spatial queries.



(b) Adapted features under both splits, with spatially selective survivors highlighted.

Figure 3: Identifying spatial reasoning features. Evidence of a distribution shift from $\mathcal{D}$base to $\mathcal{D}$sp, with adapted features highlighted after the full selection pipeline.

7 Auto-Interp and Manual Inspection

To further characterize the adapted features, we developed an automated interpretation pipeline. For each feature, we collect its top-activating samples from two sources: general VQA questions from VQAv2 (not limited to spatial reasoning) and the Visual Spatial Reasoning (VSR) dataset, which is inherently spatial. This pairing highlights whether the same feature meaning—such as object orientation or relative position—emerges consistently across both settings (Fig. 4 and 10).

The combined VQA–VSR samples are passed to the `gpt-4o-mini` [38] API in JSON mode, which assigns a confidence score and generates a short description with common patterns and cue counts. Outputs are stored with the selection metrics from Sec. 5 and lightly checked by hand. The retained set thus reflects both automatic labeling and human verification.

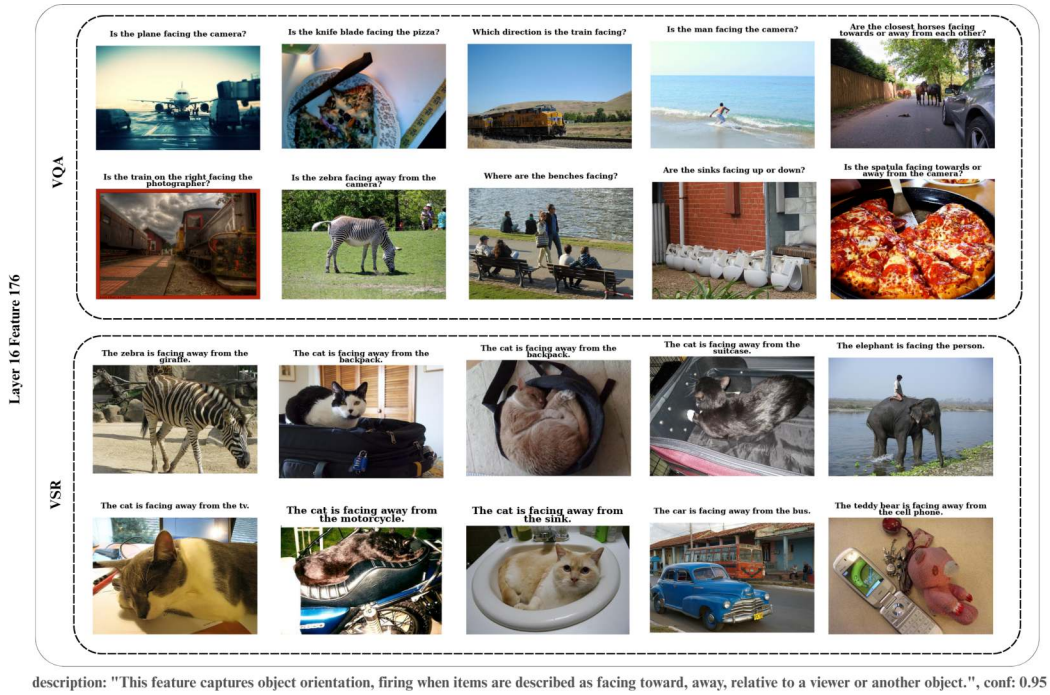


Figure 4: **Qualitative Auto-Interp example.** Layer 16, Feature 176 (conf. 0.95). Top VQA (general) and VSR (spatial) samples both highlight the same concept of *object orientation*—firing when items are described as facing toward, away, or relative to another object.

8 Attribution Patching to Identify Spatial Heads

Method. Attribution patching [34] is a scalable alternative to activation patching [50], which measures causal effects by replacing activations with counterfactual values. While activation patching requires a separate forward pass per intervention, attribution patching uses a gradient-based linear approximation to estimate the effect of all interventions with only two forward passes and one backward pass. This makes it practical to probe attribution scores across all layers and attention heads in large multimodal models.

We adapt attribution patching to identify which attention heads drive spatially selective SAE features. For a target feature f at layer L, we define a scalar objective by projecting the layer-L activations onto the SAE decoder vector. Gradients of this objective with respect to upstream query/key activations indicate how strongly each attention head contributes to f.

We compare two runs:

- **Clean run:** the original image–text input.

- **Corrupt run:** the same input, but with layer-0 visual token embeddings replaced by a *mean embedding* computed over many VQA samples. This corruption preserves plausible distributional statistics while deliberately suppressing spatial information.

We then compute two attribution variants, differing in whether the perturbation direction is taken from the corrupted or the clean representation:

$$\text{Method A:} \quad (\text{corr} - \text{clean}) \cdot \nabla_{\text{clean}},$$
$$\text{Method B:} \quad (\text{clean} - \text{corr}) \cdot \nabla_{\text{corr}}.$$

Method A measures how strongly the clean gradients indicate that ablating spatial detail affects the feature, whereas Method B measures how strongly the corrupted gradients indicate that retaining spatial detail matters. In both cases, we obtain per-layer and per-head attribution scores, averaged over the top-k VQA samples that most strongly activate f.

Results. Across the spatially selective features we examined, attribution patching with both methods reveals consistent trends. Layer-wise attribution curves typically peak in mid-to-deep layers, consistent with the emergence of spatial features in Sec. 6. At the head level, both methods generally highlight a small subset of heads with notably high scores, and the top heads identified are often consistent across the two attribution methods. This suggests that spatial information is mediated by a specialized group of heads rather than being spread uniformly across the model.

To illustrate the effect of attribution patching on individual features, Appx 11-12 provide detailed examples. In each case, attribution scores isolate a handful of mid- to deep-layer heads, and qualitative maps confirm that high-scoring heads focus on regions consistent with the queried relation (e.g., "on top of," "behind"), whereas low-scoring heads fail to do so. Interestingly, when we look across multiple related spatial features together, we find that some of the same heads recur across related spatial relations. Figure 5 illustrates this pattern. In the top row, L13H1 attends to semantically relevant regions across queries. As a control, the middle row shows that bottom-ranked heads on the same samples fail to localize meaningfully. The bottom row further confirms that irrelevant queries do not trigger spurious activation. More generally, these same heads also attend to meaningful regions such as salient objects or attributes under custom prompts (Appx. Fig. 13), underscoring that attribution patching identifies a small set of heads that reliably carry spatial–semantic signal.

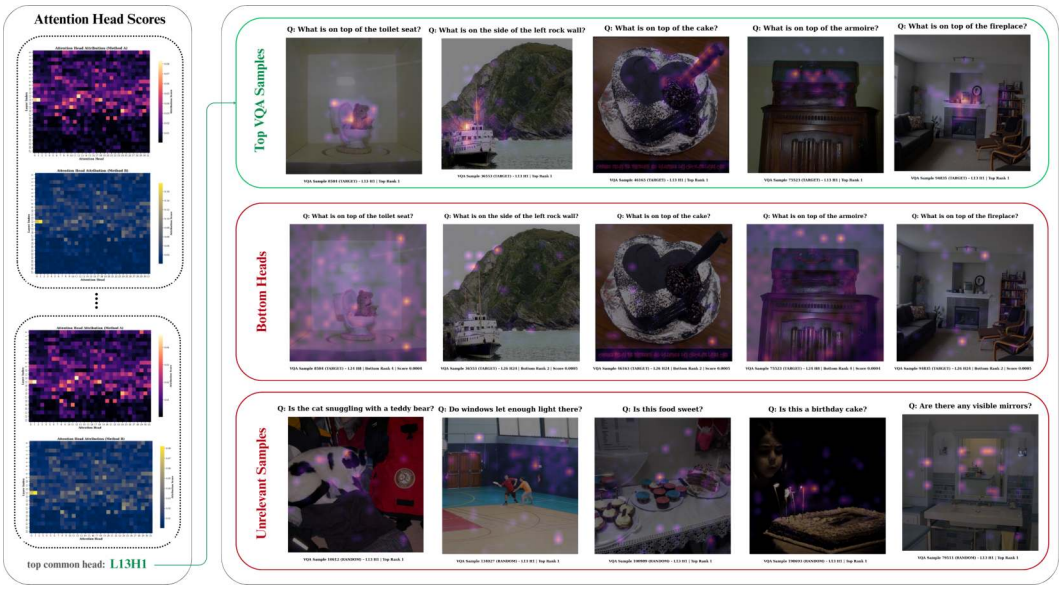


Figure 5: Attribution patching across related spatial features. Top: recurring top-scoring head (L13H1) localizes to relevant regions in queries about "on top of" relations. Middle: bottom-ranked heads on the same samples fail to capture spatial structure. Bottom: unrelated queries confirm that the top head does not spuriously activate.

9 Ablation Study

We test whether adapted SAE features and attention heads are *causally involved* in spatial reasoning by ablating them during inference and comparing performance on Visual Spatial Reasoning (VSR) and a general Yes/No subset of VQAv2. For both datasets, we use Yes/No prompts and evaluate with accuracy and mean $P(\text{correct})$. Baseline and ablation runs always use identical cached indices for fairness.

Feature ablation. For a target feature f at layer L, we project out its decoder direction v (unit norm) from the residual stream at text positions, leaving image tokens unchanged:

$$y \leftarrow y - (y^\top v)\, v.$$

We compare performance with and without ablation, and also run controls on randomly chosen features or on relation-mismatched subsets. This allows us to test whether the feature is specifically used for spatial reasoning.

Attention head ablation. We test causality at the head level by replacing the activations of selected attention heads with mean values computed from a small calibration set. This "mean-patching" removes head-specific signals while leaving the rest of the model intact, allowing us to measure their contribution to VSR performance.

Interpretation. Ablations provide additional evidence that spatial features and heads carry causal weight. Removing a recurring spatial feature (e.g., L26F807) on relation-matched VSR items consistently reduces accuracy (approximately 10–20 percentage points) and lowers mean $P(\text{correct})$ by a few points. Similar trends are observed when ablating the corresponding attention heads, suggesting that both feature- and head-level pathways mediate spatial information. In contrast, random or mismatched ablations show little effect, supporting the specificity of the result.

10 Limitations

Our analyses indicate spatial selectivity, but more detailed ablation and steering studies are needed to fully validate causality. Moreover, our experiments are limited to a single model (LLaVA-More with a LLaMA-3.1-8B backbone); applying the method to other backbones and larger corpora will be key to assessing generality.

11 Conclusion

We set out to understand how a pretrained language backbone learns to "see" under multimodal fine-tuning. By extending stage-wise model diffing to the vision–language setting, we isolated vision-preferring features that undergo strong rotations during training, showed that a subset reliably encodes spatial relations, and traced their causal drivers to a small number of mid-to-deep attention heads. These results show that multimodal adaptation is structured and interpretable: it can be localized, probed, and explained at the feature level. Beyond spatial reasoning, our methodology offers a general framework for uncovering how new capabilities emerge in large models, with practical implications for auditing, safety, and domain-specific fine-tuning. We view this work as an early step toward a mechanistic science of multimodal training, where models can be interpreted both in terms of their outputs and the internal features that support them.

References

[1] Mistral AI. *Pixtral 12B: A New Frontier in Image and Text Understanding*. https://mistral.ai/news/pixtral-12b/. Accessed: 2024-12-21. Sept. 2024.

[2] Jinze Bai et al. "Qwen-vl: A frontier large vision-language model with versatile abilities". In: *arXiv preprint arXiv:2308.12966* (2023).

[3] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. "Revisiting model stitching to compare neural representations". In: *Advances in neural information processing systems* 34 (2021), pp. 225–236.

[4] Serguei Barannikov et al. "Representation topology divergence: A method for comparing neural network representations". In: *arXiv preprint arXiv:2201.00058* (2021).

[5] Samyadeep Basu et al. "Understanding information storage and transfer in multi-modal large language models". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 7400–7426.

[6] Trenton Bricken et al. "Stage-Wise Model Diffing". In: (2024). https://transformer-circuits.pub/2024/model-diffing/index.html.

[7] Trenton Bricken et al. "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning". In: *Transformer Circuits Thread* (2023). https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[8] Haozhe Chen et al. "Interpreting and controlling vision foundation models via text explanations". In: *arXiv preprint arXiv:2310.10591* (2023).

[9] Federico Cocchi et al. "LLaVA-MORE: A Comparative Study of LLMs and Visual Backbones for Enhanced Visual Instruction Tuning". In: *arXiv preprint arXiv:2503.15621* (2025).

[10] Hoagy Cunningham et al. "Sparse autoencoders find highly interpretable features in language models". In: *arXiv preprint arXiv:2309.08600* (2023).

[11] Matt Deitke et al. "Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models". In: *arXiv preprint arXiv:2409.17146* (2024).

[12] Guanting Dong et al. "Progressive Multimodal Reasoning via Active Retrieval". In: 2024. URL: https://api.semanticscholar.org/CorpusID:274859457.

[13] Yuhao Dong et al. "Insight-V: Exploring Long-Chain Visual Reasoning with Multimodal Large Language Models". In: *arXiv preprint arXiv:2411.14432* (2024).

[14] Dumitru Erhan et al. "Why Does Unsupervised Pre-training Help Deep Learning?" In: 11 (Mar. 2010), pp. 625–660. ISSN: 1532-4435.

[15] Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. "Interpreting clip's image representation via text-based decomposition". In: *arXiv preprint arXiv:2310.05916* (2023).

[16] Leo Gao et al. "Scaling and evaluating sparse autoencoders". In: *arXiv preprint arXiv:2406.04093* (2024).

[17] Yash Goyal et al. "Making the v in vqa matter: Elevating the role of image understanding in visual question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6904–6913.

[18] Aaron Grattafiori et al. "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783* (2024).

[19] Jiayi He et al. "Self-correction is more than refinement: A learning framework for visual and language reasoning tasks". In: *arXiv preprint arXiv:2410.04055* (2024).

[20] Zhengfu He et al. "Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders". In: *arXiv preprint arXiv:2410.20526* (2024).

[21] Evan Hubinger et al. "Sleeper agents: Training deceptive llms that persist through safety training". In: *arXiv preprint arXiv:2401.05566* (2024).

[22] Nick Jiang et al. "Interpreting and editing vision-language representations to mitigate hallucinations". In: *arXiv preprint arXiv:2410.02762* (2024).

[23] Pegah Khayatan et al. "Analyzing Fine-tuning Representation Shift for Multimodal LLMs Steering alignment". In: *arXiv preprint arXiv:2501.03012* (2025).

[24] Connor Kissane et al. "SAEs (usually) Transfer Between Base and Chat Models". Interim report on AI Alignment Forum. AI Alignment Forum post. July 2024.

[25] Simon Kornblith et al. "Similarity of neural network representations revisited". In: *International conference on machine learning*. PMlR. 2019, pp. 3519–3529.

[26] Karel Lenc and Andrea Vedaldi. "Understanding image representations by measuring their equivariance and equivalence". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 991–999.

[27] Bo Li et al. "Llava-onevision: Easy visual task transfer". In: *arXiv preprint arXiv:2408.03326* (2024).

[28] Feng Li et al. "Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models". In: *arXiv preprint arXiv:2407.07895* (2024).

[29] Yixuan Li et al. "Convergent learning: Do different neural networks learn the same representations?" In: *arXiv preprint arXiv:1511.07543* (2015).

[30] Jack Lindsey et al. *Sparse Crosscoders for Cross-Layer Features and Model Diffing*. Published on Transformer Circuits Thread; https://transformer-circuits.pub/2024/crosscoders/index.html. Oct. 2024.

[31] Haotian Liu et al. "Improved baselines with visual instruction tuning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 26296–26306.

[32] Haotian Liu et al. "Visual instruction tuning". In: *Advances in neural information processing systems* 36 (2023), pp. 34892–34916.

[33] Julian Minder et al. "Robustly identifying concepts introduced during chat fine-tuning using crosscoders". In: *arXiv preprint arXiv:2504.02922* (2025).

[34] Neel Nanda. *Attribution Patching: Activation Patching at Industrial Scale*. https://www.neelnanda.io/mechanistic-interpretability. Accessed: 2025-08-23. 2023.

[35] Clement Neo et al. "Towards interpreting visual information processing in vision-language models". In: *arXiv preprint arXiv:2410.07149* (2024).

[36] Chris Olah et al. "Zoom In: An Introduction to Circuits". In: *Distill* (2020). https://distill.pub/2020/circuits/zoom-in. DOI: 10.23915/distill.00024.001.

[37] Christopher Olah. *Visualizing Representations: Deep Learning and Human Beings*. https://colah.github.io/posts/2015-01-Visualizing-Representations/. Accessed: 2025-08-23. 2015.

[38] OpenAI. *GPT-4O-Mini: Advancing Cost-Efficient Intelligence*. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2024-12-21. 2024.

[39] Vedant Palit et al. "Towards vision-language mechanistic interpretability: A causal tracing tool for blip". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 2856–2861.

[40] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PmLR. 2021, pp. 8748–8763.

[41] Sarah Schwettmann et al. "Multimodal neurons in pretrained text-only transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 2862–2867.

[42] Gabriela Ben Melech Stan et al. "LVLM-Interpret: an interpretability tool for large vision-language models". In: *arXiv preprint arXiv:2404.03118* (2024).

[43] Adly Templeton et al. "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet". In: *Transformer Circuits Thread* (2024). URL: https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[44] Shengbang Tong et al. "Eyes wide shut? exploring the visual shortcomings of multimodal llms". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9568–9578.

[45] Constantin Venhoff et al. "How Visual Representations Map to Language Feature Space in Multimodal LLMs". In: *arXiv preprint arXiv:2506.11976* (2025).

[46] Constantin Venhoff et al. "Too Late to Recall: The Two-Hop Problem in Multimodal Knowledge Retrieval". In: *Mechanistic Interpretability for Vision (Non-proceedings Track), CVPR 2025*. 2025. URL: https://openreview.net/forum?id=VUhRdZp8ke.

[47] Constantin Venhoff et al. "Too Late to Recall: The Two-Hop Problem in Multimodal Knowledge Retrieval". In: *CVPR 2025 Workshop on Mechanistic Interpretability of Vision (MIV)*. Non-proceedings Track Poster. 2025.

[48] Zhiyu Wu et al. "DeepSeek-VL2: Mixture-of-Experts Vision-Language Models for Advanced Multimodal Understanding". In: *arXiv preprint arXiv:2412.10302* (2024).

[49] Guowei Xu et al. “LLaVA-o1: Let Vision Language Models Reason Step-by-Step”. In: *arXiv preprint arXiv:2411.10440* (2024).

[50] Fred Zhang and Neel Nanda. “Towards Best Practices of Activation Patching in Language Models: Metrics and Methods”. In: *International Conference on Learning Representations (ICLR)*. arXiv:2309.16042. 2024. URL: https://doi.org/10.48550/arXiv.2309.16042.

[51] Ruohong Zhang et al. “Improve vision language model chain-of-thought reasoning”. In: *arXiv preprint arXiv:2410.16198* (2024).

A Appendix

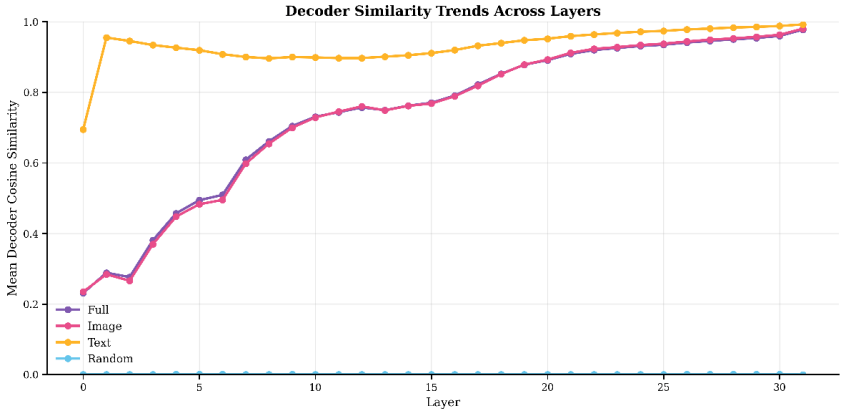


Figure 6: **Decoder cosine similarity vs. layer (LLM SAE vs. VLM SAE).** Text-only stays highly aligned across layers; image-only and full-sequence rotate in shallow layers and align later; random remains near zero. Higher cosine indicates closer alignment of SAE decoder directions.

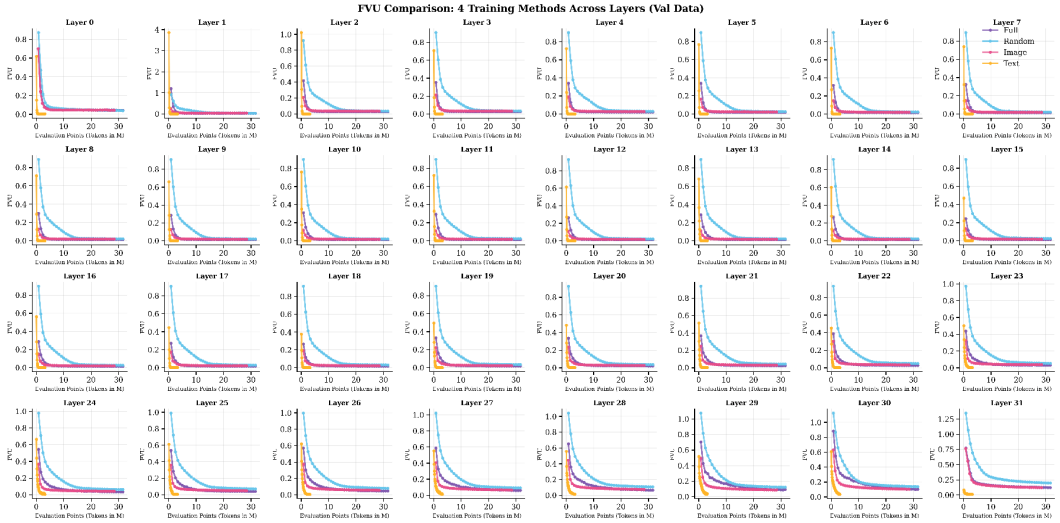
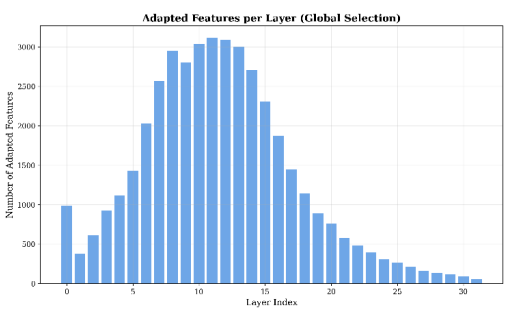
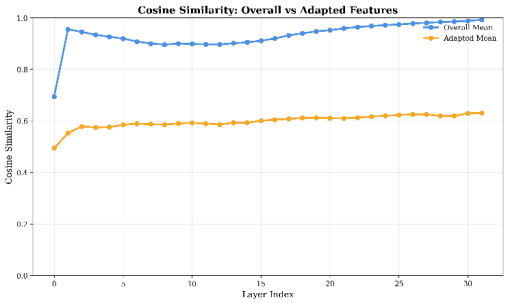


Figure 7: **Per-layer FVU across regimes.** Each panel shows the convergence of SAEs trained with different masking regimes for a specific layer. Text-only SAEs begin with slightly higher error in the shallowest layers but adapt almost immediately to near-zero reconstruction. Image-only and full-sequence SAEs converge more slowly and plateau at higher error, while random initialization performs worst throughout. This confirms that projector-driven spans remain off-distribution in early layers and only align with the LLM basis in later layers.



(a) **Adapted features per layer.** Most concentrate in mid layers, tapering in deeper blocks.



(b) **Decoder cosine by layer.** Adapted features remain less aligned to the base dictionary than the overall pool.

Figure 8: **Per-layer statistics of adapted features.** (a) Distribution of adapted feature counts across depth. (b) Mean decoder cosine similarity for adapted features vs. the overall pool.

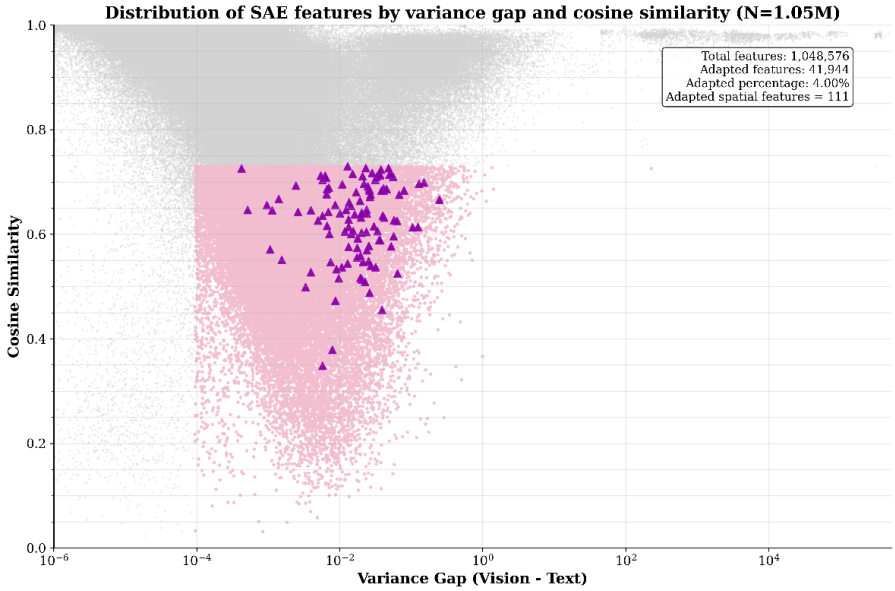


Figure 9: **Joint distribution of SAE features by variance gap and cosine similarity.** Adapted features (pink) are highlighted, with the retained spatial subset (purple) concentrated in the high-frequency tail.

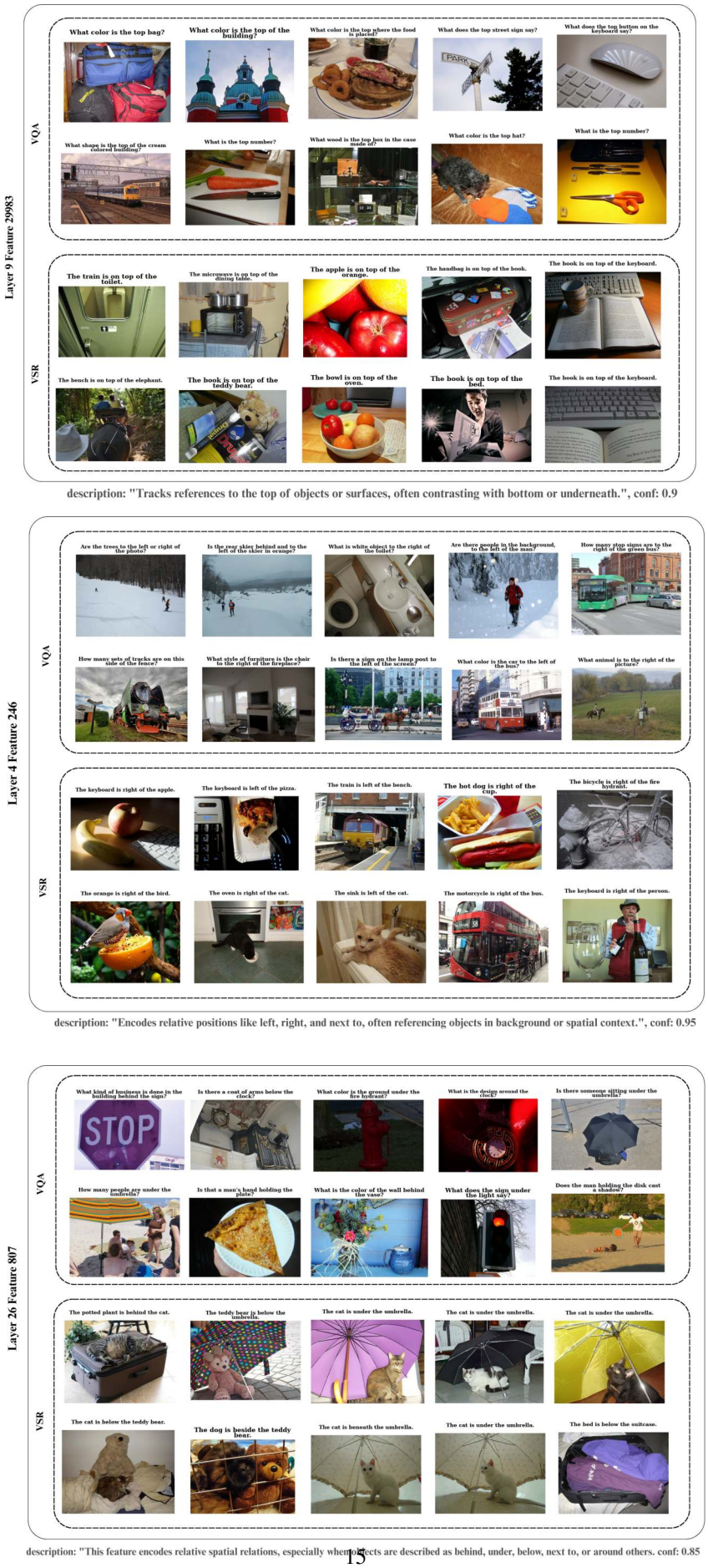
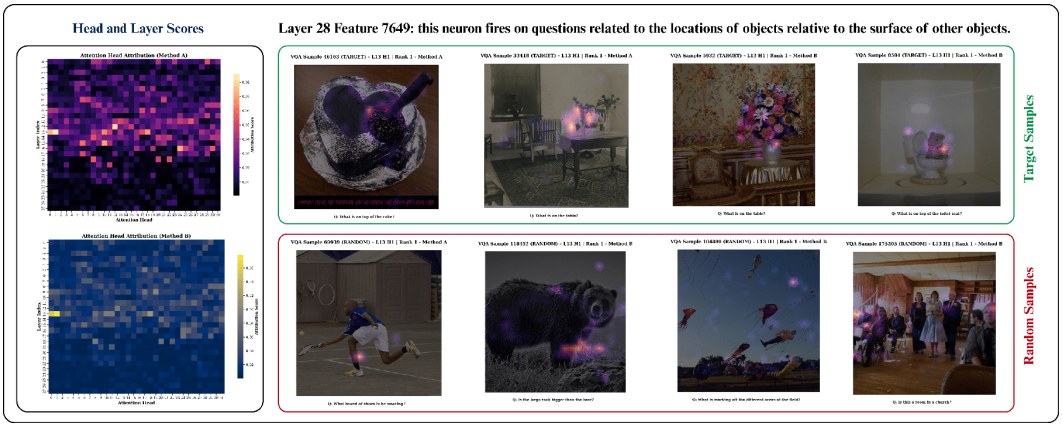
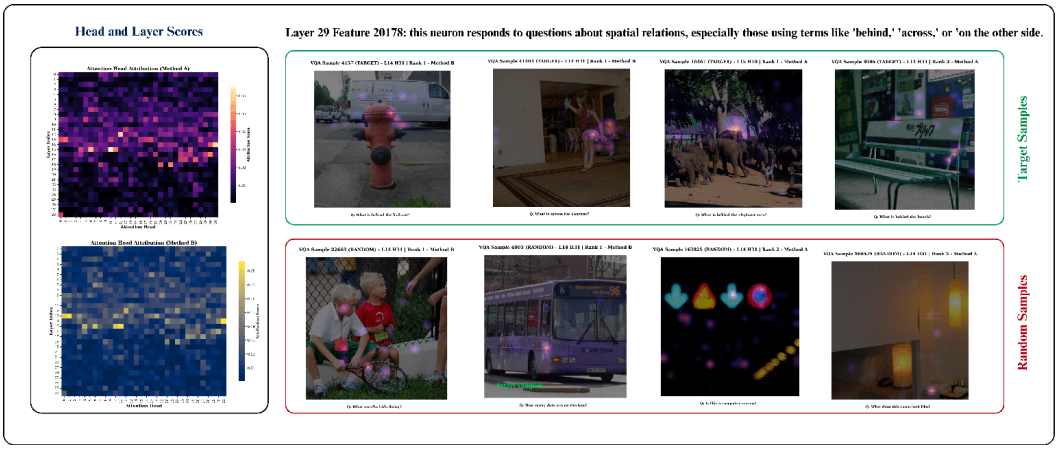


Figure 10: **Qualitative Auto-Interp examples.** Top-activating VQA and VSR samples for three adapted features with short GPT-4o-mini–generated descriptions.



(a) Object placement relations (‘on’, ‘on top of’, ‘on the side of’).



(b) Spatial relation queries (‘behind’, ‘across’, ‘on the other side’).

Figure 11: Neuron interpretability examples of object placement and spatial relations.

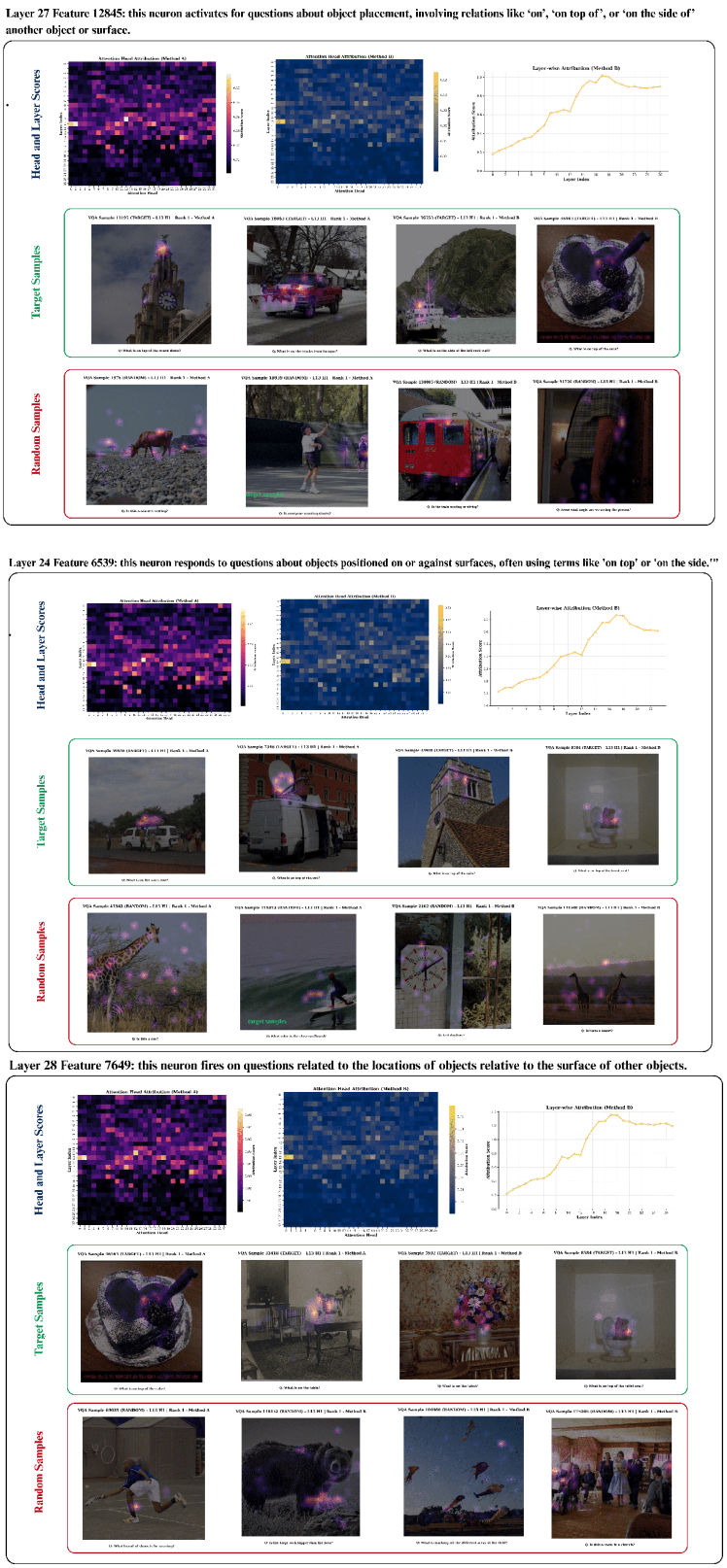


Figure 12: Consistency of attribution results across related spatial features. In all three cases, the same attention head (**L13H1**) is identified as the top contributor under both methods.

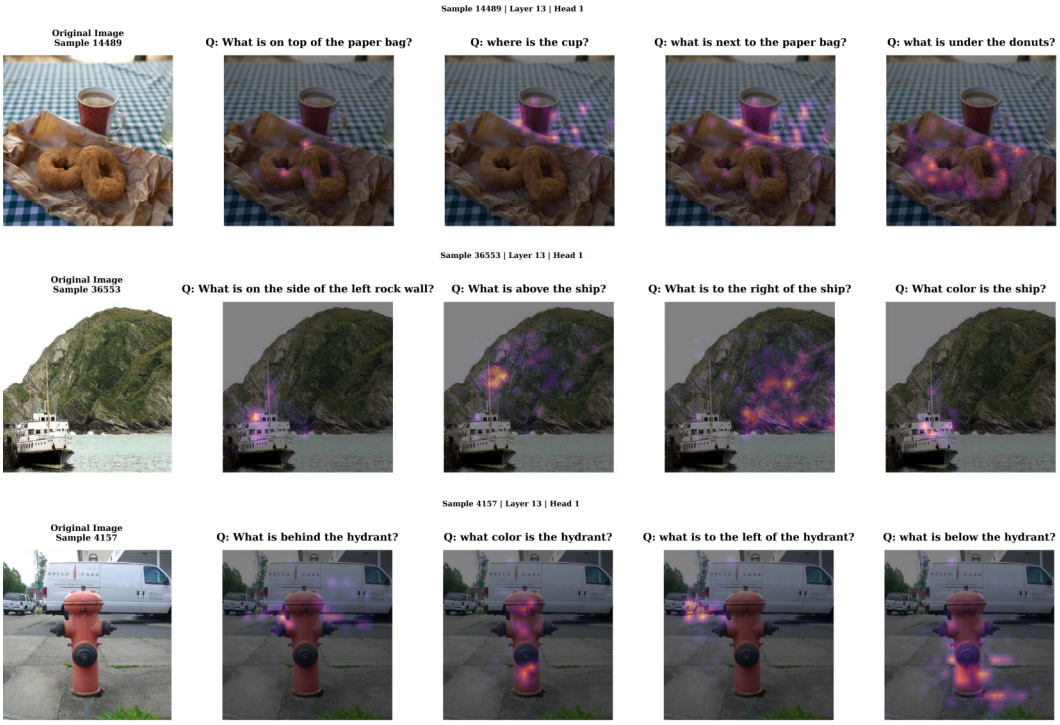


Figure 13: **Attention head visualizations across queries.** Each row shows one image with attention overlays from a single high-attribution head across multiple spatial and non-spatial custom queries. The same heads consistently focus on semantically relevant regions.

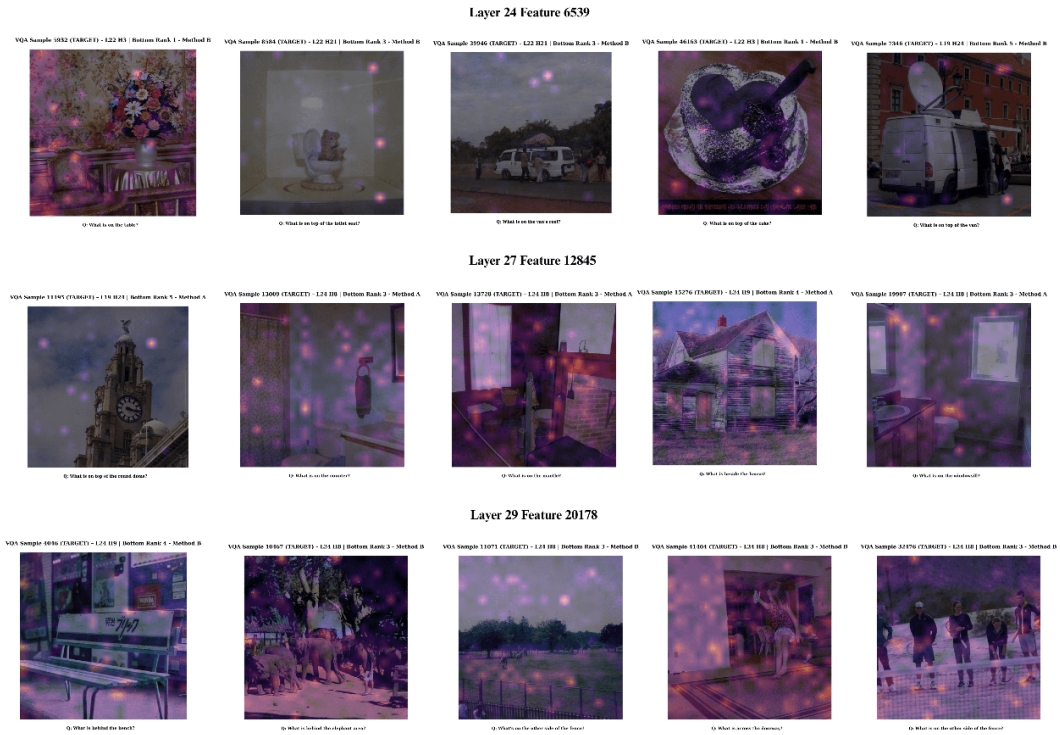


Figure 14: **Low-attribution heads.** Bottom-ranked heads yield diffuse or irrelevant attention, showing little relation to the spatial queries.