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

Lachin Naghashyar
University of Oxford
lachin.naghashyar@cs.ox.ac.uk

Hunar Batra
University of Oxford
hunar.batra@cs.ox.ac.uk

Ashkan Khakzar
University of Oxford
ashkan.khakzar@eng.ox.ac.uk

Philip Torr
University of Oxford
philip.torr@eng.ox.ac.uk

Ronald Clark
University of Oxford
ronald.clark@cs.ox.ac.uk

Christian Schroeder de Witt
University of Oxford
christian.schroeder@eng.ox.ac.uk

Constantin Venhoff
University of Oxford
constantin.venhoff@cs.ox.ac.uk

Abstract

Contemporary Vision–Language Models (VLMs) achieve strong performance on a wide range of tasks by pairing a vision encoder with a pre-trained language model, fine-tuned for visual–text inputs. Yet despite these 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 process. Using stage-wise model diffing, a technique that isolates representational changes introduced during multimodal fine-tuning, we reveal how a language model learns to "see". We first identify vision-preferring features that emerge or reorient during fine-tuning. We then show that a selective subset of these features reliably encodes spatial relations, revealed through controlled shifts to spatial prompts. Finally, we trace the causal activation of these features to a small group of attention heads. Our findings show that stage-wise model diffing reveals when and where 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 models 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 [29, 1]. 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

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability Workshop at NeurIPS 2025.

representations [53, 55, 13, 12]. Yet we lack a mechanistic account of how language representations adapt during multimodal training and when vision-specific capabilities emerge [24, 49, 46].

In this work, we introduce a method 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 features rotate, emerge, or are repurposed, it has been shown to uncover subtle shifts such as sleeper-agent features [21]. 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 VQAv2 dataset samples[17]. This warm-start preserves the original feature basis 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. Figure 3 highlights the filtered spatial features.

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-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 extend stage-wise model diffing to the multimodal setting, providing the first feature-level account of how pretrained language backbones adapt under visual grounding.
- We introduce a systematic pipeline to isolate adapted features, identify those selectively recruited by spatial queries, and filter out lexical artifacts.
- We show that these spatially selective SAE features are functionally involved in reasoning, through empirical evidence and ablation studies, supported by interpretive checks.
- We causally attribute the emergence of spatial features to a small subset of attention heads using scalable attribution patching, highlighting structured pathways for visual grounding.

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 [41, 14], stitching intermediate layers across models [27, 3], or defining new similarity metrics [26, 4]. Later studies examined alignment at the level of individual neurons, showing convergent units across independently trained networks [30, 40].

Sparse autoencoders (SAEs) offered a feature-level lens, and prior work [25] 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 precisely interpret those changes. Stage-wise model diffing [6] offers such fine-grained resolution, revealing sleeper-agent features and distinguishing between base and chat-tuned models [37]. Extensions to multimodal models highlight similar representational shifts, with concept-shift vectors proposed for steering [24] and evidence that alignment converges in middle-to-late layers [49]. These remain semantic-level analyses, whereas our work applies stage-wise diffing with SAEs to the backbone, giving the first mechanistic account of multimodal fine-tuning, showing how it rotates features and induces spatial grounding in 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 and causal analyses aim to explain model behavior at a high level. Approaches include interpretability toolkits based on attention patterns, relevancy maps, and causal interventions [46]. Other work uses interventions to trace how information is stored and transferred [5], or applies causal mediation to study how BLIP integrates visual evidence [43]. Second, probing-based studies focus on the representations themselves. Several works analyzed CLIP, identifying both strengths and limitations [48, 15, 8]. Others reported multimodal neurons responsive to joint visual–textual concepts [45] and examined how VLMs differentiate hallucinated from real objects [22]. More recent methods map visual embeddings into linguistic space, projecting features onto language vocabularies [39] or showing the late emergence of visual signals in LLM backbones [50].

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 a visual encoder f_V, a pretrained language model f_{LM}, and a trainable projector P. The visual encoder (e.g., a ViT [44]) extracts image patch embeddings $V = f_V(x) = [v_1, \ldots, v_{N_V}]$, which the projector maps into token space $\tilde{V} = P(V)$. 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 transformer layers of multi-head self-attention and feed-forward networks. For each head h, attention is computed as

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

where M is the causal mask preventing attention to future tokens. The outputs of all heads are concatenated and projected into the hidden dimension, and mapped through the unembedding matrix to predict next tokens. For our experiments, we adopt LLaVA-More [9], which extends LLaVA framework [35, 34] 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 $x \in \mathbb{R}^D$ into

$$f(x) = \mathrm{ReLU}(W_{\text{enc}}x + b_{\text{enc}}), \quad \hat{x} = W_{\text{dec}}f(x) + b_{\text{dec}},$$

with $W_{\text{enc}} \in \mathbb{R}^{F\times D}$, $b_{\text{enc}} \in \mathbb{R}^F$, $W_{\text{dec}} \in \mathbb{R}^{D\times F}$, and $b_{\text{dec}} \in \mathbb{R}^D$. Training minimizes

$$\mathcal{L} = \|x - \hat{x}\|_2^2 + \lambda \sum_{i=1}^{F} |f_i(x)|,$$

combining reconstruction with an L_1 sparsity penalty. 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. Variants such as Top-K SAEs [16] further sharpen this tradeoff by enforcing hard sparsity, improving interpretability and reducing feature co-adaptation.

SAEs have been widely applied to uncover monosemantic features and offer a practical lens on model internals, enabling analyses that range from probing knowledge to tracing safety-relevant behaviors [7, 10]. They are not, however, a complete decomposition: interpretability can vary across runs

and training setups, and recent work suggests their practical utility may be more limited in some settings [47, 23]. Even so, SAEs have proven particularly effective for *model diffing*, where they make it possible to track how features shift across training stages and to surface subtle but behaviorally important dynamics—a direction we expand on in the next subsection [6, 36].

3.3 Stage-Wise Model Diffing

A recent line of work in model diffing has introduced *stage-wise model diffing* [6], which extends SAE analysis across training stages by re-training dictionaries on activations from successive checkpoints while keeping feature indices aligned. This makes it possible to compare whether units are preserved, rotated, or repurposed during adaptation. Applied to controlled fine-tuning trajectories, it disentangles changes due to model updates from dataset shifts and highlights features that drive adaptation. Prior work has shown that stage-wise diffing uncovers fine-grained dynamics, including sleeper-agent features that remain dormant in pretraining but activate once safety constraints are lifted [21]. Compared to crosscoder-based methods [32], it provides finer resolution at the feature level, though it remains limited to aligned checkpoints of the same architecture.

4 Stage-wise Model Diffing for Multimodal Adaptation

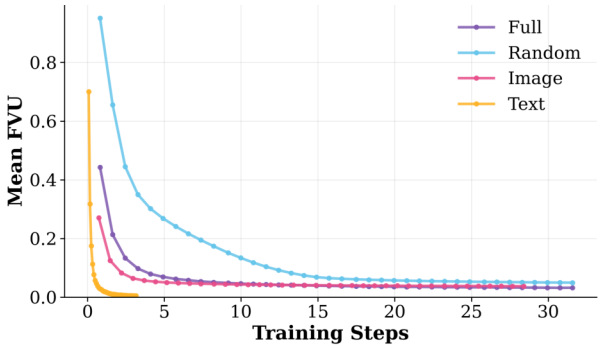
Overview. We aim to understand how multimodal fine-tuning reshapes model representations, using spatial reasoning as a case study of a distinctly multimodal task that integrates both visual and linguistic cues. To this end, we take inspiration from stage-wise diffing 3.3, employing sparse autoencoders (SAEs) as a feature-level lens to track how internal directions shift when a pretrained language backbone is exposed to visual inputs. Our pipeline has three stages. First, we fine-tune SAEs on multimodal activations to obtain a feature dictionary aligned with the vision–language space. Second, we isolate features that prefer visual tokens and undergo substantial geometric rotation, indicating that they have been repurposed by multimodal training. Third, we probe for spatial reasoning by contrasting generic VQA with spatial queries and keeping only features that increase under the shift while remaining active under neutral instructions, ensuring they are not driven by lexical artifacts. In this way, we reduce the original pool of over one million features to a compact set of candidates plausibly recruited for spatial reasoning tasks.

4.1 Adapting Language Dictionaries to Vision-Language Space

We start by adapting sparse autoencoders (SAEs) trained on the 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 VQA benchmark of images and open-ended questions. 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 release [20], re-instantiated as a Top-K model (k=50), preserving a meaningful, interpretable basis. Since our VLM shares the same backbone, this warm-start ensures continuity with the pretrained language feature space and avoids retraining from scratch, allowing us to directly leverage millions of monosemantic features across layers. 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 code selectivity. 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 more slowly to 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 model 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 [51]. 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. In contrast, image and full-sequence SAEs plateau at higher error, highlighting the instability of projector-driven spans (see Appx.Fig.7). We therefore focus stage-wise diffing on text-only SAEs, where alignment is stable and feature-level identifiability is more plausible.

4.2 Identifying Adapted Features

We aim to isolate SAE features that (i) undergo geometric reorientation after multimodal adaptation and (ii) show a clear *modality preference* for vision input. Such features are the most informative for model diffing and subsequent causal analysis. To identify them, we rely on two signals:

1. 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.

2. Modality preference (visual energy). Given the sparsity of SAE activations, we score each feature f by its mean squared activation under vision inputs,

$$E_v(f) \;=\; \mathbb{E}_{\text{vision}}\big[h_f^2\big],$$

measured on VQA runs of the VLM. Since nearly half of features have $E_v = 0$, a simple cutoff $E_v > \epsilon$ suffices to discard inactive directions and retain those that carry visual signal.

Selection Procedure We define adapted features as those that meet both criteria: $E_v > \epsilon$, ensuring reliable visual responsiveness, and a cosine similarity c_f in the bottom $p_{cos} = 25\%$, indicating strong decoder rotation. Applying these filters jointly yields a globally defined set comprising about 5% of all features. The joint distribution of E_v and c_f is shown in Fig. 3, with the selected subset highlighted in pink. Details on threshold choices, together with per-layer counts and mean cosine similarities, are provided in Appx. Fig. 8a and Appx. 8b.

4.3 Case Study: Identifying Spatial Reasoning Features

We identify spatial features using two signals: (i) recruitment under a shift to spatial queries, and (ii) persistence under neutral prompts that rule out lexical artifacts.

Datasets. Our analysis uses two evaluation sets 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.

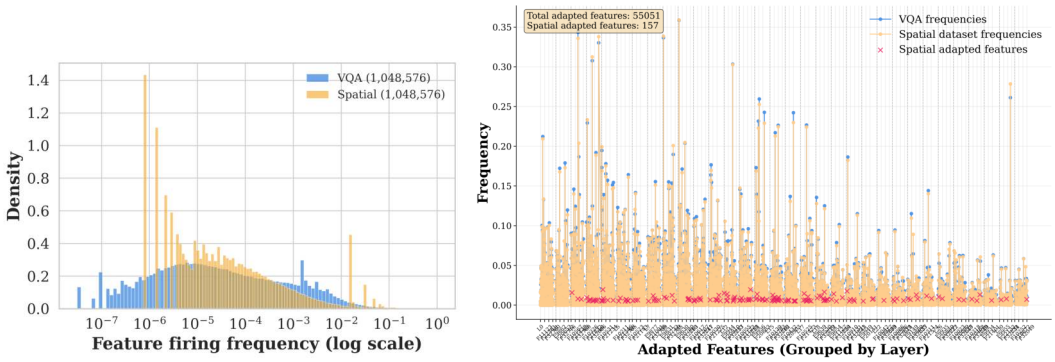
1. Distribution shift Let $h_f(x_t) \geq 0$ denote the activation of feature f on token t of input x. For a dataset $\mathcal{D}$, the firing frequency of f is

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

where $n(\mathcal{D})$ is the total number of tokens. Fig 2a 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 meaningful Δp_f and OR_f are flagged as spatial *candidates* in Fig.3.

2. Filtering lexical artifacts. To rule out prompt-lexical effects, we replace the original questions in each top-activating sample with neutral spatial prompts such as *"Describe the positions of objects in the image."*. Features that continue firing under these generic instructions are preserved as genuinely image-grounded, while those that fail to activate are discarded. This ensures that the surviving units reflect spatial reasoning rather than memorized lexical cues.

From these filtered candidates, we retain only those also in the adapted set $\mathcal{A}$ (Sec. 4.2), ensuring they reorient under multimodal fine-tuning and respond to spatial shifts. The surviving features are shown in Fig. 2 and in Fig. 3 (blue). A subset, marked with red crosses, is further analyzed via automated interpretation, attribution patching, and ablations (Sec. 5.1, 5.2).



(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 2: 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.

5 Experiments

5.1 Auto-Interp and Preliminary Inspection

As an initial step toward understanding the selected features, we carried out a preliminary inspection using an automated interpretation pipeline. For each feature, we collect its top-activating samples from two sources: general VQA questions from VQAv2 (not restricted to spatial reasoning) and the

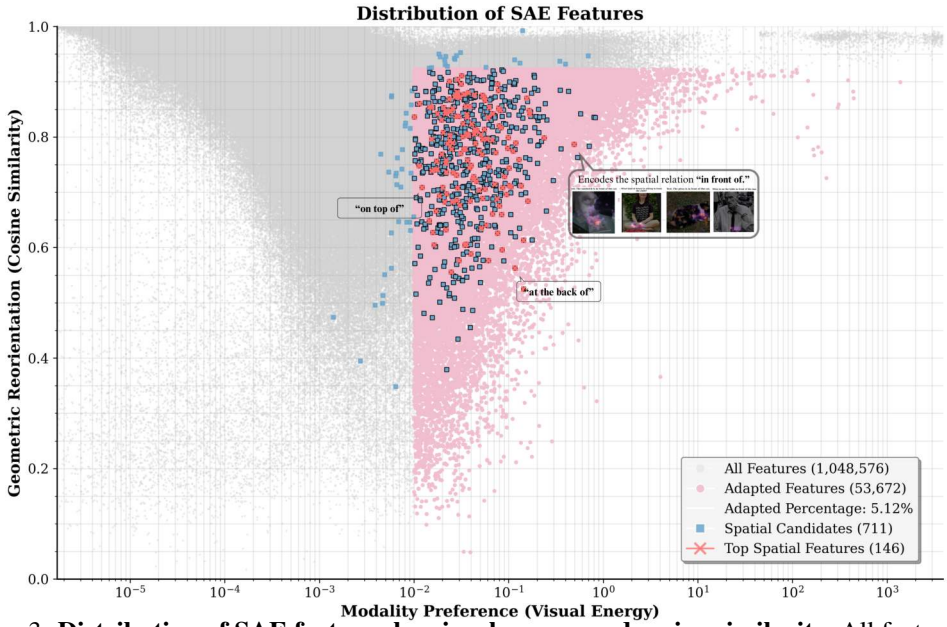


Figure 3: **Distribution of SAE features by visual energy and cosine similarity.** All features are shown in gray; adapted features are highlighted in pink. Spatial candidates are marked with blue squares, and the subset used for downstream analysis is shown as red crosses.

Visual Spatial Reasoning (VSR) dataset [33], which is inherently spatial. This pairing allows us to check whether the same underlying meaning emerges consistently across both settings (Fig. 4). A subset of the combined samples are then passed to the `gpt-4o-mini` [42] API, which proposes a concise one-sentence description for each feature and assigns an interpretability confidence score based on F1 from a validation classification task. The resulting outputs are stored together with the selection metrics from Sec. 4.2, and are lightly reviewed by hand, so that the retained set reflects both automatic labeling and human verification (see App. B for additional examples and scoring details)..

5.2 Attribution Patching to Identify Spatial Heads

Method. Attribution patching [38] is a scalable alternative to activation patching [54], which measures causal effects by replacing activations with counterfactuals. While activation patching requires a separate forward pass per intervention, attribution patching uses a gradient-based linear approximation to estimate interventions with two forward and one backward pass. This makes it practical to probe attribution scores across layers and heads in MLLMs.

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 w.r.t. 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.

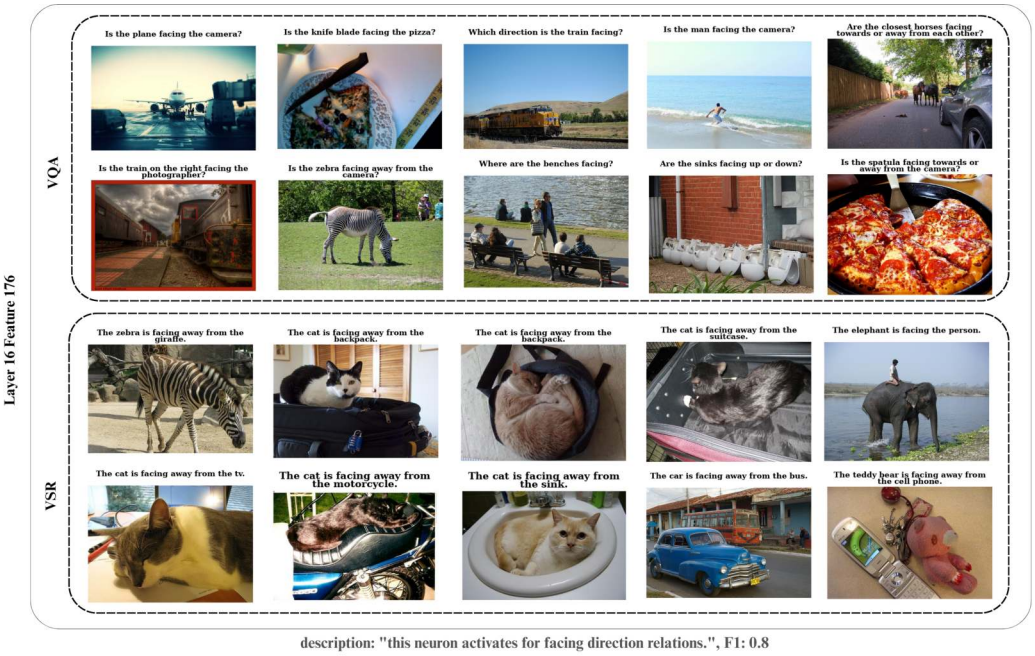


Figure 4: **Auto-Interp example (Layer 16, Feature 176).** Top VQA and VSR samples both highlight *facing direction*, with activation on objects described as facing toward, away, or relative to another.

Results. Across the spatially selective features we examined, attribution patching with both methods reveals consistent trends. Layer-wise attribution curves typically peak in middle layers, consistent with the emergence of spatial features in Sec. 4.3 (Appx. Fig. 11). 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 (Appx. Fig. 12). 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. Fig. 13 provides detailed examples. In each case, attribution scores isolate a handful of 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. These head-level overlays can also be used to (i) improve the confidence of automated feature interpretation by coupling sample activations with attention visualizations, and (ii) examine failure cases by checking whether the top spatial features and heads attend to valid regions in misclassified samples. Interestingly, when we look across multiple related spatial features together, we find that some of the same heads recur across related spatial relations. Fig 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. 15), underscoring that attribution patching identifies a set of heads that reliably carry spatial–semantic signal.

5.3 Ablation Study

We test whether adapted SAE features are *causally involved* in spatial reasoning by ablating them during inference and measuring performance on VSR [33], a dataset of text–image pairs spanning dozens of spatial relations, and on a *Yes/No* subset of VQAv2 (general). Each feature is evaluated on a *relation-specific subset* of VSR constructed from its top-activating samples, so that the ablation directly targets the relation it most strongly encodes. To ablate a target feature f at layer L, we orthogonally remove its decoder direction v (unit norm) from the residual stream at *text* token

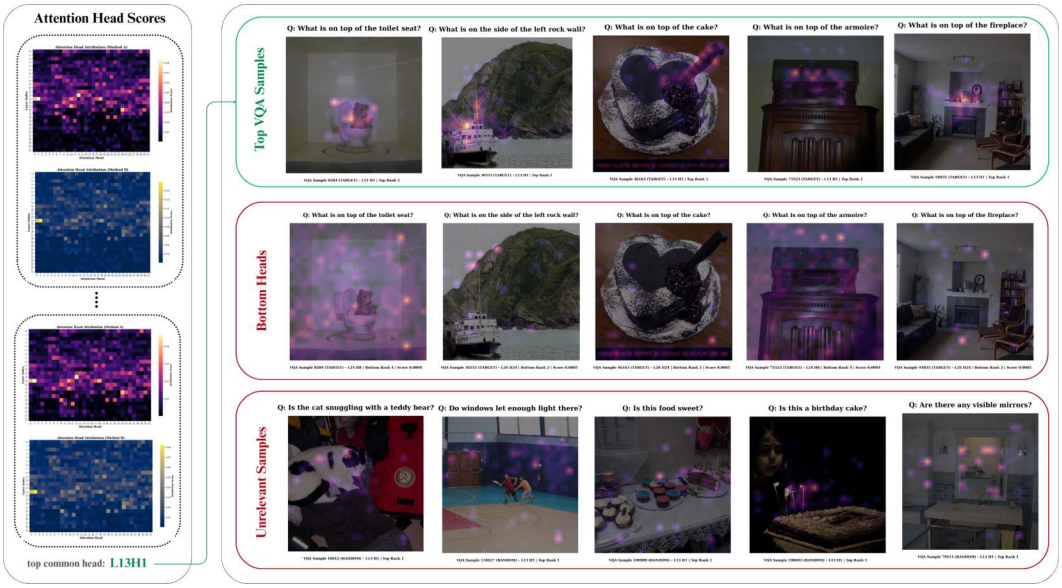


Figure 5: Attribution patching across related spatial features. Top: recurring top-scoring head (L13H1) localizes to relevant regions in queries about "on top of" relations. Middle: bottom-ranked heads on the same samples fail to capture spatial structure. Bottom: unrelated queries confirm that the top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

positions, leaving image tokens unchanged:

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

Evaluation metrics. We report: (i) accuracy drop on VSR (ΔVSR Acc; $\downarrow$ is worse), (ii) accuracy drop on VQA (ΔVQA Acc), (iii) accuracy drop from ablating same-layer random features (ΔCtrl), and (iv) odds ratio under the spatial distribution shift (VSR OR; $\uparrow$ is better). All runs use identical cached indices, and results are averaged over seeds.

Layer	Feature	ΔVSR Acc ($\downarrow$)	ΔVQA Acc ($\downarrow$)	ΔCtrl	VSR Relations	VSR OR ($\uparrow$)
7	15870	-15.54	-0.10	-0.88	above	4.32
11	27061	-12.77	-0.40	0.00	across from	8.03
9	15404	-11.19	-0.80	1.08	below	5.60
14	17873	-10.21	-0.30	-1.71	at the right side of	7.17
12	23874	-9.05	-0.40	-0.95	left of	9.10
18	29948	-7.98	-0.30	0.00	beside	8.36

Table 1: **Top ablated SAE features** ranked by VSR accuracy drop. Columns 2–4 show accuracy drops on VSR, VQA, and random-feature controls; the final column gives odds ratio (VSR OR) as a measure of selective recruitment. Large ΔVSR Acc with small ΔVQA Acc indicates spatial specificity, while near-zero ΔCtrl confirms robustness.

Interpretation. Ablating the top spatial features lowers VSR accuracy by 9–16 points on average while leaving general VQA nearly unchanged (≤ 1 pp), indicating that these directions are functionally used for spatial reasoning rather than general behavior. This shows that probing or switching off a single feature can selectively disable spatial reasoning without harming overall ability. High odds ratios further show selective recruitment under spatial prompts. Random-feature controls yield effects near zero or inconsistent in sign, supporting specificity. Full per-feature results, probability deltas, and seed-wise summaries are reported in Appx. D.

6 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.

7 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-layer attention heads. These results show that multimodal adaptation is structured and interpretable as it can be localized, probed, and explained at the feature level. Beyond spatial reasoning, our methodology offers a general framework for uncovering when and where new capabilities emerge in large models, showing that multimodal adaptation follows structured patterns rather than diffuse changes. 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] Subhash Kantamneni et al. "Are Sparse Autoencoders Useful? A Case Study in Sparse Probing". In: *arXiv preprint arXiv:2502.16681* (2025).

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

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

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

[27] 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.

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

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

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

[31] 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.

[32] Jack Lindsey et al. “Sparse Crosscoders for Cross-Layer Features and Model Diffing”. In: *Transformer Circuits Thread* (2024). URL: https://transformer-circuits.pub/2024/crosscoders/index.html.

[33] Fangyu Liu, Guy Edward Toh Emerson, and Nigel Collier. “Visual Spatial Reasoning”. In: *Transactions of the Association for Computational Linguistics* (2023).

[34] 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.

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

[36] Julian Minder, Clément Dumas, and Neel Nanda. “What We Learned Trying to Diff Base and Chat Models (And Why It Matters)”. In: *LessWrong* (2025). URL: https://www.lesswrong.com/posts/xmpauEXEerzYcJKNm/what-we-learned-trying-to-diff-base-and-chat-models-and-why.

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

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

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

[40] 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.

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

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

[43] 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.

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

[45] 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.

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

[47] 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.

[48] 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.

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

[50] 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.

[51] 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.

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

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

[54] 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.

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

A Appendix

A.1 Geometry divergence: decoder cosine trends

To quantify how SAE feature geometry shifts across training regimes, we track cosine similarity between decoder directions from SAEs trained on different input types. Fig 6 shows that text-only SAEs remain closely aligned across layers, while image-only and full-sequence SAEs diverge in early layers before realigning deeper in the model. Randomly initialized SAEs stay largely uncorrelated, confirming the stability of the observed trends.

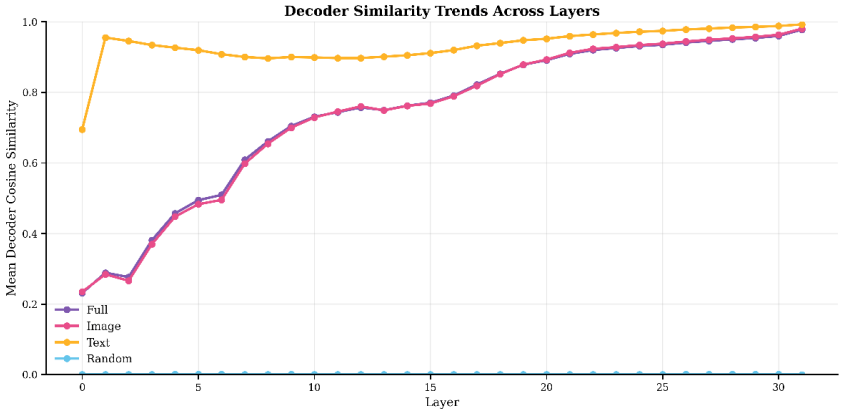


Figure 6: **Decoder cosine similarity vs. layer (LLM SAE vs. VLM SAE).** Text-only stays highly aligned across layers; image-only and full-sequence rotate in shallow layers and align later; random remains near zero. Higher cosine indicates closer alignment of SAE decoder directions.

A.2 Per-layer FVU trajectories

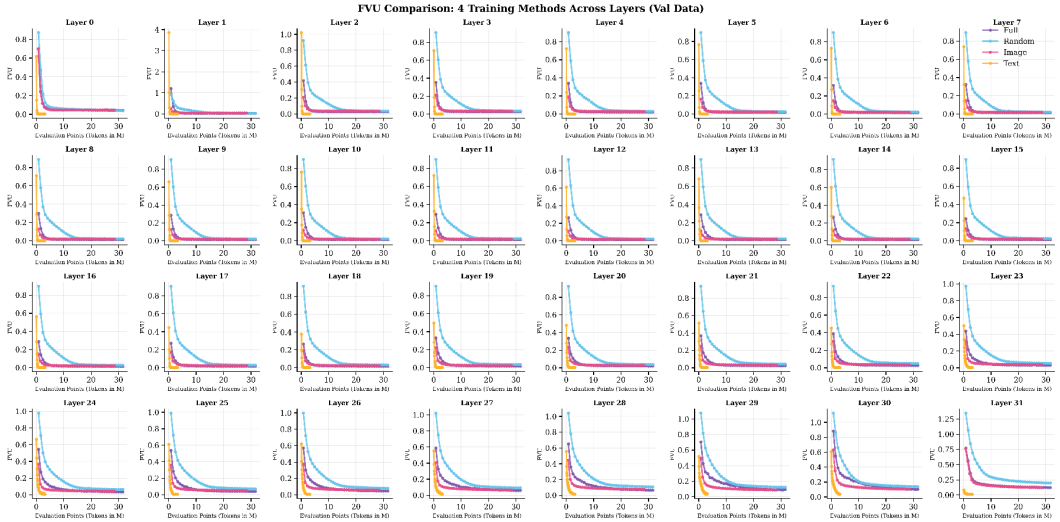
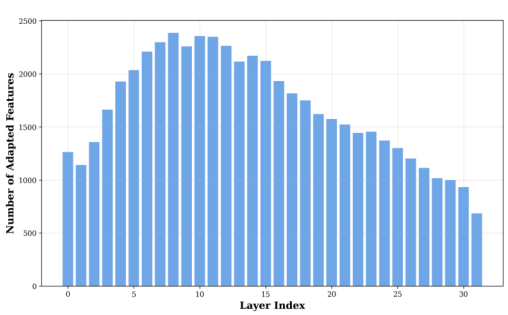


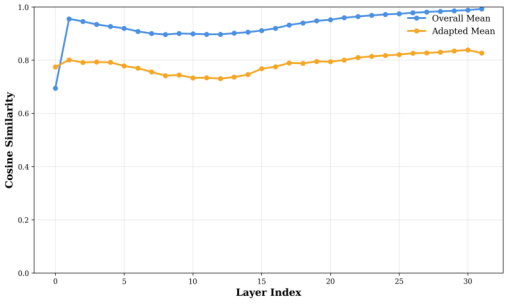
Figure 7: **Per-layer FVU across regimes.** Each panel shows the convergence of SAEs trained with different masking regimes for a specific layer. Text-only SAEs begin with slightly higher error in the shallowest layers but adapt almost immediately to near-zero reconstruction. Image-only and full-sequence 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.3 Per-Layer Statistics

Fig 8 shows that adapted features cluster in mid layers and taper in deeper blocks. Their decoder directions remain less aligned to the base dictionary than the overall pool, confirming stronger rotations under multimodal fine-tuning.



(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.

A.4 Threshold Sweep for Feature Selection

To ensure that our choice of thresholds is robust, we sweep over the cosine percentile cutoff ($p_{\cos}$) and visual energy threshold (ϵ). Fig. 9 reports three metrics: (i) total number of selected features, (ii) Jaccard overlap with the baseline adapted set, and (iii) per-layer count correlation. The results show a broad stable region around $\epsilon \approx 10^{-3}$ and $p_{\cos} \approx 25\%$, which yields a compact yet consistent set of adapted features. We adopt this operating point (white circle) for all downstream analyses.

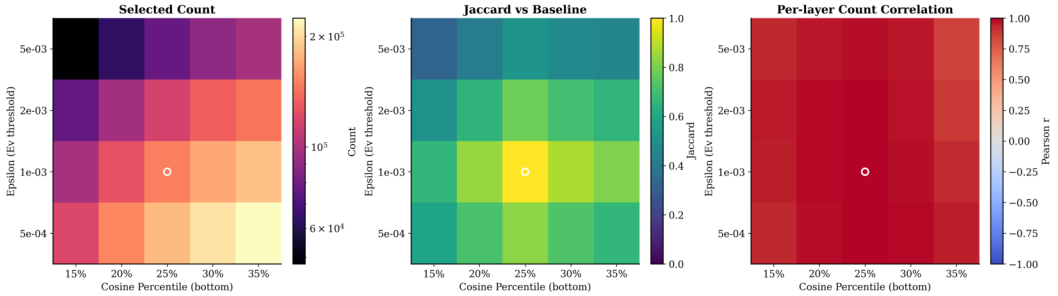
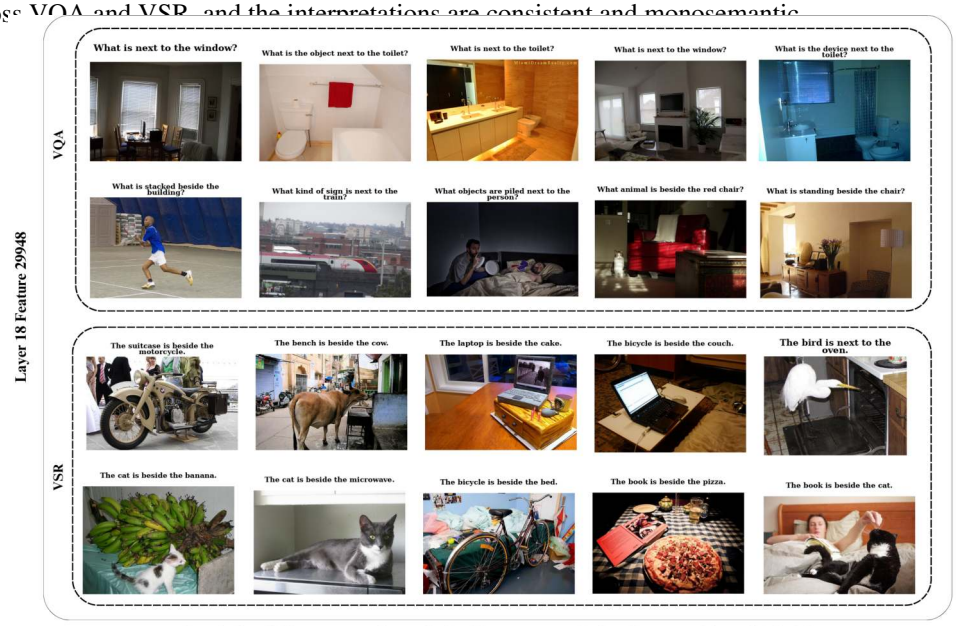


Figure 9: **Threshold sweep for feature selection.** Left: feature counts increase smoothly with more lenient thresholds. Middle: Jaccard overlap with the baseline peaks near the chosen point. Right: per-layer counts remain highly correlated across thresholds. The white circle marks the adopted operating point.

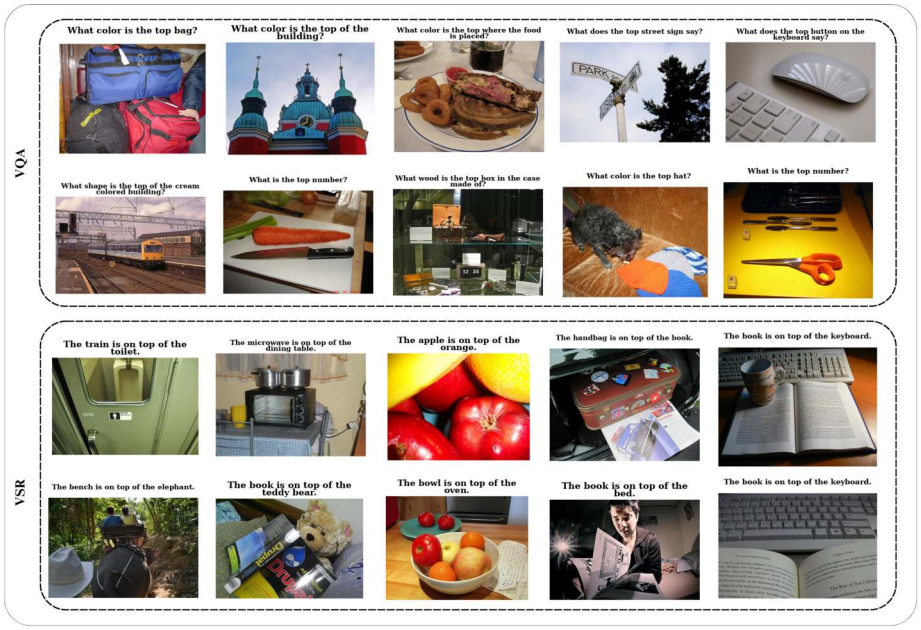
The visual-energy statistic E_v is computed under a text-only mask, since our SAEs are text-only. As a result, most features have $E_v = 0$, so requiring $\epsilon > 0$ acts as a strong filter. When cross-checking with downstream spatial tasks, we find that features with very low E_v rarely contribute meaningfully: they tend to cluster in shallow layers, show low spatial hit rates, and often appear polysemantic on inspection. In contrast, those that pass the ϵ cutoff carry a cleaner visual signal and align more consistently with spatially selective units in downstream evaluations, suggesting that the thresholded set captures genuinely vision-grounded features.

B Additional Auto-Interp Examples

In the main text (Sec. 5.1), we showed examples of adapted features using our automated interpretation pipeline. We include two further examples here. In both cases, the top-activating samples agree across VQA and VSR, and the interpretations are consistent and monosemantic.



description: "this neuron activates for beside or next to relations between objects.", F1: 0.8



description: "this neuron activates for top of relations.", F1: 0.7

Figure 10: **Additional Auto-Interp examples.** Top-activating VQA and VSR samples for two adapted features, showing consistent spatial relations.

B.1 Scoring and Implementation Details

For completeness, we outline the scoring pipeline. For each feature, we collect its top five activating samples from VQA and VSR, since the `gpt-4o-mini` API only supports up to five examples at a

time. These samples are passed to the model, which generates a one-sentence description of the feature. To assess reliability, we then draw additional positives (other high-activation samples of the same feature) and negatives (unrelated samples), and ask the model to classify them using the generated description. We report F1 on this task as a lightweight proxy for interpretability confidence.

C Attribution Patching Additional Experiments Results

C.1 Aggregated Attribution Results

To complement the per-feature attribution patching results in the main text, we provide aggregated views over layers and heads on the top 150 features ranked by odds ratio and ablation results. These plots confirm that both attribution patching methods yield consistent trends, with middle layers showing the strongest contributions and only a small subset of attention heads emerging as dominant.

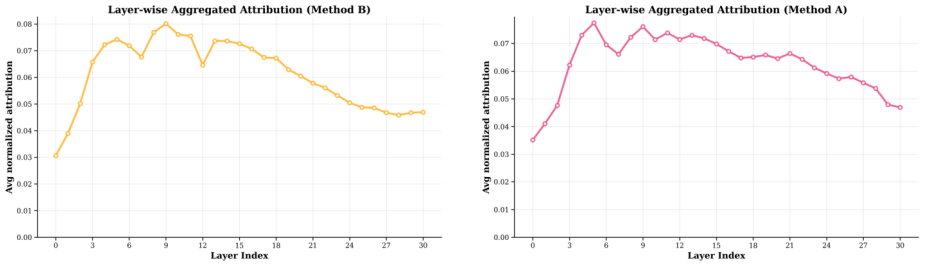


Figure 11: Layer-wise aggregated attribution curves for Method B (left) and Method A (right). Both peak in around middle layers, consistent with the emergence of spatial features.

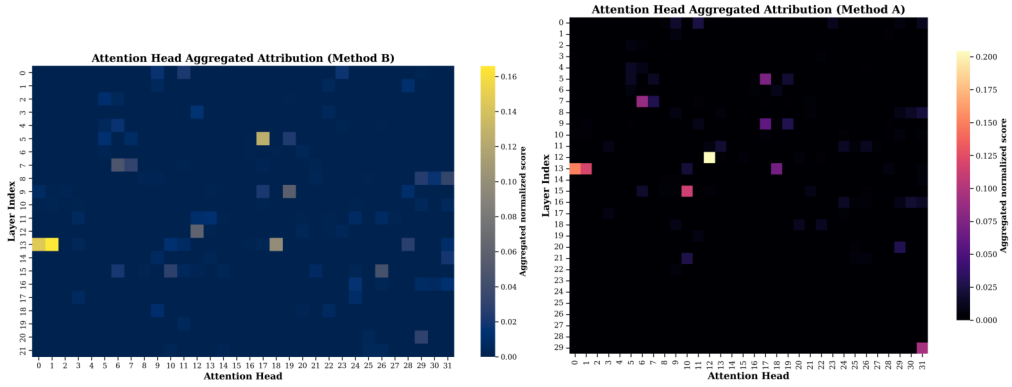
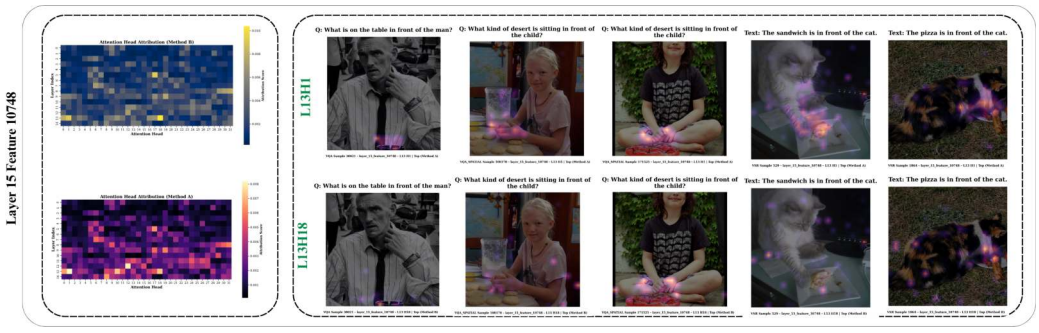


Figure 12: Attention head aggregated attribution maps for Method B (left) and Method A (right). Both highlight a similar set of specialized heads with high attribution scores.

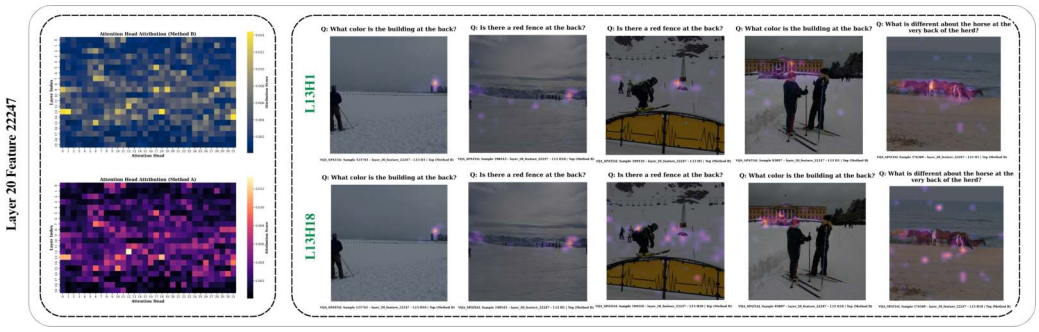
C.2 Per-Feature Panels with Top Heads

For individual spatial features, we show (i) per-layer/head attribution maps (Methods A and B) and (ii) attention overlays from the strongest heads on the feature's top-activating samples across both VSR and VQA datasets.

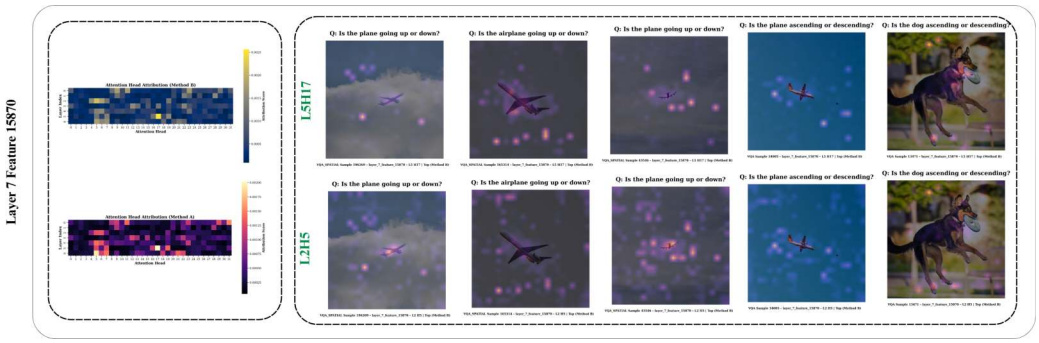
Across these examples, we see consistent overlap between the heads surfaced by the two attribution methods, with certain heads repeatedly appearing across different spatial relations. This convergence supports the idea that a group of heads concentrates much of the spatial signal. At the same time, Method B tends to yield clearer head rankings and more focused overlays, suggesting it provides a stronger and more reliable signal for pinpointing the causal drivers of spatial features.



(a) **Layer 15, Feature 10748.** VSR Relation: "in front of." Top heads (Method A): `L13H1`, `L12H12`, `L13H18`. Top heads (Method B): `L13H18`, `L5H17`, `L13H1`. *Overlap*: `L13H1`, `L13H18`. Attention overlays are shown on the top-activating samples across VSR and VQA.



(b) **Layer 20, Feature 22247.** VSR Relation: "at the back of." Top heads (Method A): `L12H12`, `L13H18`, `L13H1`. Top heads (Method B): `L13H1`, `L13H18`, `L14H31`. *Overlap*: `L13H1`, `L13H18`. Attention overlays are shown on the top-activating samples across VSR and VQA.



(c) **Layer 7, Feature 15870.** VSR Relation: "above." Top heads (Method A): `L5H17`, `L6H5`, `L0H31`. Top heads (Method B): `L5H17`, `L2H5`, `L2H6`. *Overlap*: `L5H17`. Attention overlays are shown on the top-activating samples across VSR and VQA.

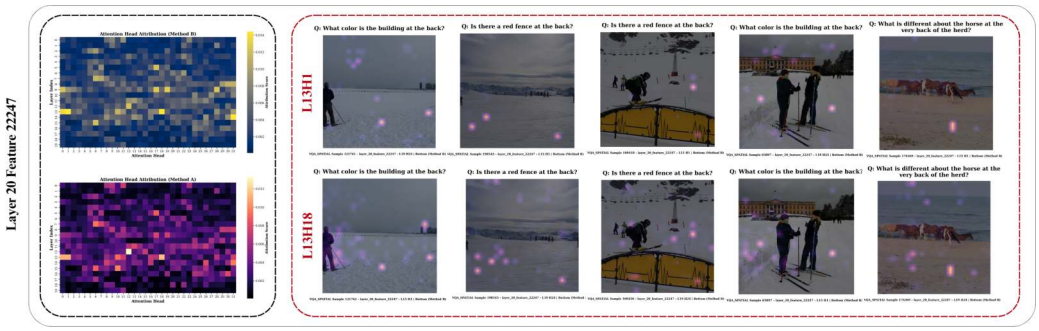
Figure 13: **Attribution patching on individual spatial features.** Each subfigure displays aggregated head/layer attribution maps (left) and attention overlays (right) using the strongest heads on the feature's top-activating samples across both VSR and VQA.

C.3 Bottom-Ranked Heads as a Control

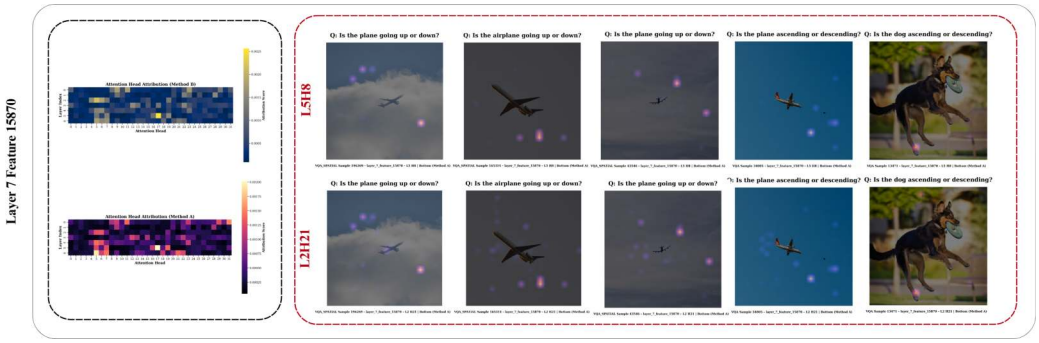
As a control, we visualize overlays from the *bottom-ranked* heads (per method, per feature). Across VSR and VQA top-activating samples, these heads generally fail to localize semantically relevant regions.



(a) **Layer 15, Feature 10748.** VSR Relation: "in front of."



(b) **Layer 20, Feature 22247.** VSR Relation: "at the back of."



(c) **Layer 7, Feature 15870.** VSR Relation: "above."

Figure 14: **Bottom-ranked heads yield weak localization.** For each feature, we show overlays from the lowest-scoring heads under Methods A and B on the feature's top-activating samples across VSR and VQA. In contrast to Appx. Fig. 13, these heads produce diffuse or irrelevant attention.

D Full Ablation Results

Table 2 reports a more detailed version of ablation results for the top SAE features. For each feature, we show average accuracy and probability drops on VSR across seeds, together with the number of evaluation samples. We also report accuracy drops on VQA, random-feature control drops (ΔCtrl), odds ratios (VSR OR), and relation-specific subsets of VSR derived from top-activating samples. Large negative ΔVSR Acc with small ΔVQA Acc indicates spatial specificity, near-zero ΔCtrl supports robustness, and high odds ratios reflect selective recruitment under spatial prompts.

E

Layer	Feature	ΔVSR Acc	ΔVSR Prob	ΔVQA Acc	ΔCtrl	VSR OR	#Samples	VSR Relations
11	27061	-13.30	-0.09	-0.40	0.00	8.03	94	across from
12	23874	-10.24	-0.10	-0.40	-0.95	9.10	210	left of
18	29948	-7.98	-0.09	-0.30	0.00	8.36	188	beside
23	4060	-5.85	-0.00	-0.70	-1.06	7.47	94	at the back of
14	17873	-10.00	-0.07	-0.30	-2.71	7.17	480	at the right side of
9	15404	-11.19	-0.07	-0.80	1.08	5.60	277	below
7	6986	-10.87	-0.03	-0.50	0.34	4.74	589	under
7	15870	-15.54	-0.09	-0.10	-0.88	4.32	341	above
10	5121	-7.92	-0.06	-0.10	0.12	4.22	846	above, on top of
11	24089	-7.68	-0.05	-0.60	-0.12	4.18	846	above, on top of
12	13305	-6.38	-0.05	-0.70	0.24	4.17	846	above, on top of

Table 2: Full ablation results for top SAE features, averaged over seeds. The number of VSR samples evaluated is shown alongside accuracy/probability drops and odds ratios.

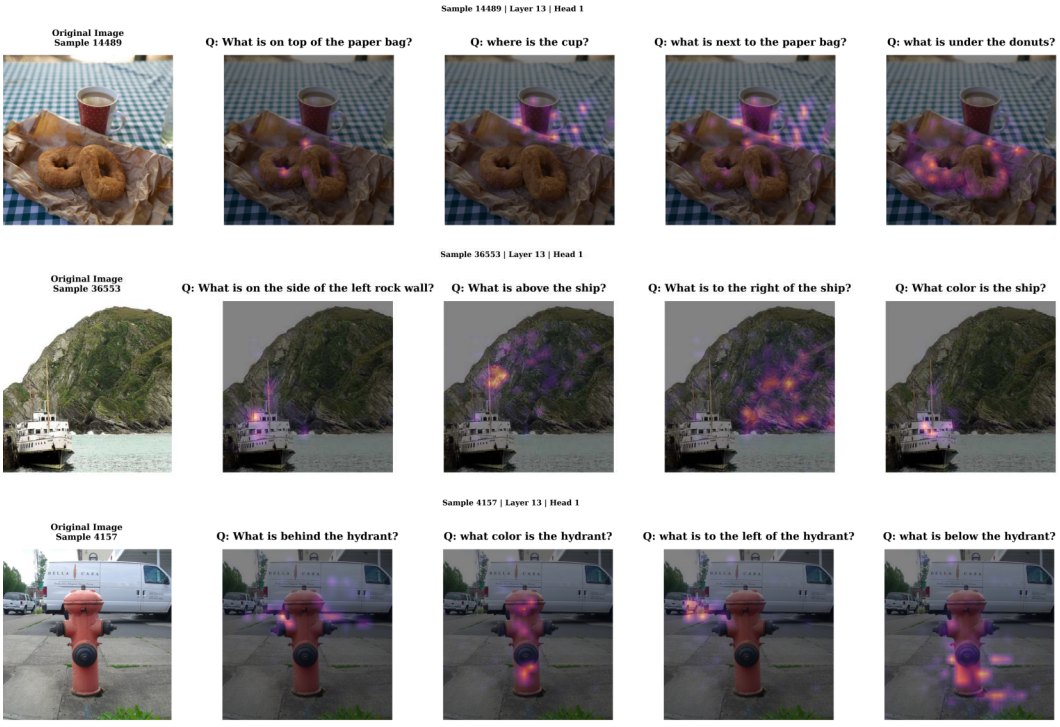


Figure 15: **Attention head visualizations across queries.** Each row shows one image with attention overlays from a single high-attribution head across multiple spatial and non-spatial custom queries. The same heads consistently focus on semantically relevant regions.