# V-SEAM: Visual Semantic Editing and Attention Modulating for Causal Interpretability of Vision-Language Models

**Qidong Wang**[1], **Junjie Hu**[2], **Ming Jiang**[2*]
[1]Tongji University, [2]University of Wisconsin-Madison
wang_qidong@tongji.edu.cn, {junjie.hu, ming.jiang}@wisc.edu

## Abstract

Recent advances in causal interpretability have extended from language models to vision-language models (VLMs), seeking to reveal their internal mechanisms through input interventions. While textual interventions often target semantics, visual interventions typically rely on coarse pixel-level perturbations, limiting semantic insights on multimodal integration. In this study, we introduce V-SEAM, a novel framework that combines **V**isual **S**emantic **E**diting and **A**ttention **M**odulating for causal interpretation of VLMs. V-SEAM enables concept-level visual manipulations and identifies attention heads with positive or negative contributions to predictions across three semantic levels: objects, attributes, and relationships. We observe that positive heads are often shared within the same semantic level but vary across levels, while negative heads tend to generalize broadly. Finally, we introduce an automatic method to modulate key head embeddings, demonstrating enhanced performance for both LLAVA and InstructBLIP across three diverse VQA benchmarks. Our data and code are released at: https://github.com/petergit1/V-SEAM.

## 1 Introduction

Vision-language models (VLMs) have become a vital infrastructure for multimodal understanding and generation, powering a variety of downstream applications, such as visual question answering (VQA) (Liu et al., 2023; Dai et al., 2023; Liu et al., 2024), image captioning (Zhang et al., 2021; Wang et al., 2024), and navigation (Anderson et al., 2018; Zhou et al., 2024). Despite these models' impressive performance, VLMs' internal mechanisms remain underexplored, risking their trustworthiness in real deployments. To address this gap, researchers have increasingly focused on interpreting VLMs, with particular emphasis on causal intervention methods given their dual benefits: causally unraveling model behaviors and providing systematic pathways for model improvement, like model editing (Zhao et al., 2024; Lin et al., 2025).

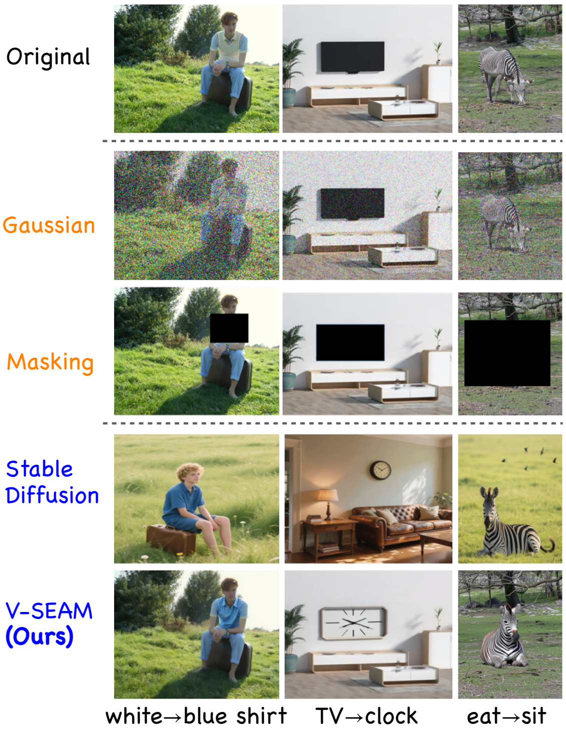


Figure 1: An example of visual intervention comparisons: Visual non-semantic vs. semantic interventions.

Existing work on causal interpretability primarily focuses on large language models (LLMs) (Vig et al., 2020; Meng et al., 2022; Geva et al., 2023; Zhao et al., 2024). Recently, some studies have adapted this strategy to VLMs by perturbing both visual and text inputs (Palit et al., 2023; Basu et al., 2024; Golovanevsky et al., 2025). However, unlike textual interventions (e.g., token replacement), which focus on fine-grained semantic changes (Basu et al., 2024; Zhang and Nanda, 2023), visual interventions, such as image-wide Gaussian noise (Palit et al., 2023) and random visual masking (Neo et al., 2024), typically em-

[*]corresponding author

*Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17396–17420
November 4-9, 2025 ©2025 Association for Computational Linguistics

phasize holistic disruptions at the pixel level (see Figure 1). Zhang and Nanda (2023) shows that this intervention gap leads to inconsistent interpretations of model mechanisms and can even produce deceptive outcomes, causing uncertainty with VLM analysis. Although the latest works propose semantic-based visual intervention by either retrieving similar image pairs or generating adversarial images via stable diffusion under the guidance of text instructions (Golovanevsky et al., 2025), this approach, as shown in Figure 1, may introduce unintended alterations to the original image, such as changes to the background or the main object. Consequently, it risks incorporating extraneous disturbances into model analysis.

In this study, we introduce V-SEAM, a fine-grained, semantics-based interpretability framework for VLMs. Motivated by the intuition that ideal visual interventions should alter only the image components relevant to a specific semantic property while preserving the rest, we propose ***V**isual **S**emantic **E**diting* to manipulate visual inputs at three semantic categories: (1) objects, (2) object attributes, and (3) object-object relationships. Building on this, we develop ***A**ttention **M**odulating*, a causal method that identifies key attention heads driving predictions at each semantic level and modulates their embeddings for model improvement.

Using V-SEAM on the visual question answering (VQA) task, we empirically investigate two popular VLMs, i.e., LLAVA and InstructBLIP, and identify four key findings: (1) VLMs generally emphasize object-level cross-modal alignment in early layers (e.g., layers 5–10), shifting to attributes and relationships in later layers (e.g., layers 9–15); (2) attention modules primarily capture salient semantic cues (e.g., color) that inform predictions, while MLP blocks play a key role in producing the final decisions; (3) for individual attention heads, we identify both positive heads that facilitate correct predictions and negative heads that introduce misleading signals. Positive heads are generally shareable within the same semantic categories but differ across categories, whereas negative heads tend to generalize across semantics; and (4) by rescaling key attention head embeddings, we achieve improved VQA accuracy for both LLAVA and InstructBLIP on three diverse benchmarks.

Overall, our main contributions are as follows:

- We introduce V-SEAM, a novel mechanistic interpretability framework that enables concept-level visual semantic editing and attention modulating to causally trace key components in VLMs for object-, attribute-, and relation-level visual understanding.
- We uncover a variety of novel insights into how VLMs process fine-grained semantic information from visual and textual inputs.
- We apply V-SEAM to improve model performance on VQA tasks by amplifying positive attention heads and suppressing negative ones through embedding rescaling. Experiments on three diverse VQA tasks demonstrate consistent improvements in VLM performance.

## 2 Related Work

**Vision-Language Models** Existing VLMs can be broadly categorized into two groups based on their strategies for integrating visual and textual information. One group employs cross-attention mechanisms to embed visual features into textual ones (Alayrac et al., 2022; Li et al., 2023a), while the other maps visual and textual features through a projection layer (Liu et al., 2023; Bai et al., 2023; Zhu et al., 2024; Chen et al., 2024; Li et al., 2024). Since projection-based VLMs typically achieve stronger performance, prior interpretability work has predominantly focused on this group (Bai et al., 2023; Liu et al., 2023). With the goal of offering a more comprehensive causal interpretation, we examine both groups of VLMs in this study.

**Multimodal Interpretability** Existing research on the interpretability of vision-language models (VLMs) can be broadly categorized into two complementary approaches: data-driven methods (Sood et al., 2023; Xing et al., 2025) and model structure–oriented analyses. Data-driven methods typically employ saliency-based techniques to associate specific input regions, e.g., pixels or visual tokens, with model outputs, thereby offering insights into which inputs most strongly influence predictions. Unlikely, structure-based analyses aim to interpret VLMs by examining their internal components and mechanisms, ranging from the study of hierarchical representations across layers (Palit et al., 2023; Zhang et al., 2024), the functional roles of distinct modules (Ben Melech Stan et al., 2024; Basu et al., 2024; Chen et al., 2025), to the contribution and specialization of attention heads (Golovanevsky et al., 2025), and the behavior of individual neurons or neuron groups (Gandelsman et al., 2023). Our work falls into this group, emphasizing both hierarchical representations and attention mechanisms within VLMs.

Within structure-based analyses, there are two major strands. One is correlation-based interpretations, diagnosing what kinds of semantic information are encoded across VLM layers and components. A common technique is probing, where classifiers are trained on hidden states to decode semantic properties (Alain and Bengio, 2016; Hendricks and Nematzadeh, 2021). One major issue with this method is its sensitivity to probe design and vulnerability to spurious correlations that obscure true model representations (Belinkov, 2022; Bilodeau et al., 2024). To address this, alternative methods have been developed. For example, TextSpan (Gandelsman et al., 2023) and Logit Lens (Neo et al., 2024) aim to align neurons or visual tokens with discrete, human-interpretable concepts, offering a more direct view into model internals. Neuron attribution methods such as MINER (Huang et al., 2024) and MMNeuron (Huo et al., 2024) identify neurons selectively activated by specific tasks or semantics. While these provide fine-grained insights into VLM organization, their interpretability is limited by neuronal polysemanticity, where single neurons may encode multiple, entangled concepts (Liu et al., 2025). In parallel, sparse autoencoder techniques show promise in disentangling representations in LLMs and ViTs (Cunningham et al., 2023), but face challenges in VLMs due to modality inconsistency and instability.

The second strand emphasizes causal tracing, investigating the causal effect of input interventions on model components (Vig et al., 2020; Geva et al., 2023). For example, Basu et al. (2024) found that early VLM layers encode factual knowledge, while Palit et al. (2023) highlighted the role of deeper layers. Golovanevsky et al. (2025) further analyzed modality-specific attention heads. However, existing interventions often rely on coarse perturbations (e.g., full-image replacement or noise), which can introduce artifacts and obscure semantic causality (Zhang and Nanda, 2023). To address this, we propose a vision-centric causal interpretability framework that supports fine-grained interventions. Our method combines semantic manipulations on image regions with activation patching on image and text tokens, enabling analysis of how fine-level semantic types (attributes, objects, relations) are processed and aligned across layers and modalities. We further perform attention-head-level interventions to identify "positive" and "negative" heads sensitive to distinct semantics, extending prior modality-specific attention analysis (Golovanevsky et al., 2025).

## 3 Preliminary

We first define the notations of data and models, then describe the *activation patching* (Meng et al., 2022) method (a.k.a. *causal tracing*) in the context of VLMs (Golovanevsky et al., 2025).

**Data and Models.** We consider a VQA dataset $\mathcal{D}$ consisting of $N$ question-image-answer triples. Each triple contains a question $x$ about an image $z$, and a single-token answer $y$. Each image $z$ is annotated with a list of objects $\mathcal{B}$, including their text labels and bounding box coordinates, and the object-object relation edges. We define a transformer-based VLM as $f_\theta$, where its parameters $\theta = \{\theta_{\text{in}}, \theta_{\text{out}}\} \cup \{\theta^l_{\text{att}}, \theta^l_{\text{mlp}}\}_{l=1}^L$ include the vision and text input encoding layer $\theta_{\text{in}}$, the output projection layer $\theta_{\text{out}}$ and $L$ transformer layers, each composed of self-attention $\theta^l_{\text{att}}$ and MLP $\theta^l_{\text{mlp}}$ modules. We denote the output logit of the correct answer $y$ to a given input $(x, z)$ as $\ell(x, z, y) \in \mathbb{R}$.

**Activation Patching in VLMs.** Given a VQA triple $(x, z, y) \in \mathcal{D}$, the method first modifies the original clean image $z$ to obtain a corrupted image $\tilde{z}$, denoted as $\tilde{z}$. It then feeds $(x, z)$ and $(x, \tilde{z})$ independently into the VLM, obtaining layerwise embeddings after self-attention and MLP modules denoted as $\boldsymbol{H}^l_\tau$, $\tilde{\boldsymbol{H}}^l_\tau \in \mathbb{R}^{T\times d}$ respectively, where $\tau \in \{\text{att}, \text{mlp}\}$, $T$ is the input sequence length, $d$ is the hidden dimension. The model's output logits for the correct answer $y$ are thus denoted as $\ell(x, z, y), \ell(x, \tilde{z}, y) \in \mathbb{R}$ respectively. We further identify the set of indices of the corrupted tokens in the combined input sequence $(x, \tilde{z})$ as $\mathcal{I}$.

An activation patching operation is then applied to modify $\tilde{\boldsymbol{H}}^l_\tau$, where $\tau \in \{\text{att}, \text{mlp}\}$. Specifically, the embeddings of the corrupted tokens in $\tilde{\boldsymbol{H}}^l_\tau$ are selectively replaced with their clean counterparts in $\boldsymbol{H}^l_\tau$, based on the indices $\mathcal{I}$ of the corrupted tokens.

$$\hat{\boldsymbol{H}}^l_\tau \leftarrow \text{Patch}(\boldsymbol{H}^l_\tau, \tilde{\boldsymbol{H}}^l_\tau, \mathcal{I}) \qquad (1)$$

The patched embedding $\hat{\boldsymbol{H}}^l_\tau$ is fed through the remaining VLM layers to obtain the patched logit for the correct answer $y$, denoted $\hat{\ell}^l_\tau(x, \tilde{z}, y) \in \mathbb{R}$. We then compute its difference from the corrupted logit to measure the effect of patching:

$$\Delta\ell^l_\tau(x, \tilde{z}, y) = \hat{\ell}^l_\tau(x, \tilde{z}, y) - \ell(x, \tilde{z}, y). \qquad (2)$$

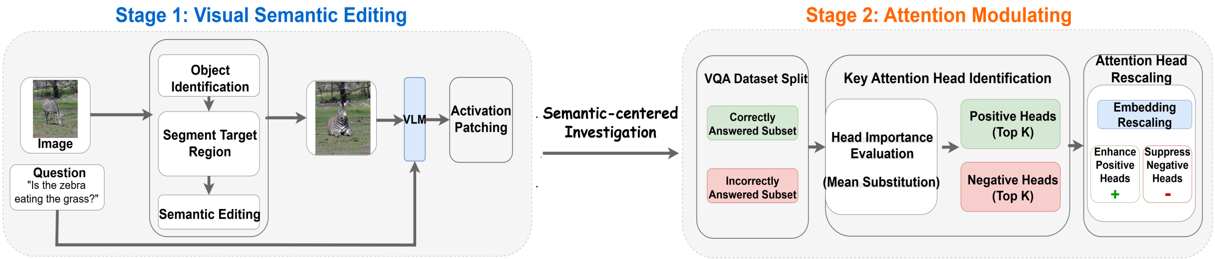


Figure 2: Our proposed semantic-level causal interpretability framework. The pipeline starts from semantic-guided perturbation of visual regions (attributes, objects, or relations), followed by vision-language model reasoning, and ends with causal analysis through activation patching and attention head ablation.

Intuitively, a large positive value of $\Delta\ell^l_\tau(x,\tilde{z},y)$ indicates that patching the corrupted embeddings after module $\tau$ at layer $l$ has a beneficial effect on predicting the correct answer for a given VQA triple. This allows us to identify modules that are critical for the model's performance. To improve estimation stability, we compute the final causal score for a VLM module as the averaged logit difference across all the $N$ triples in $\mathcal{D}$:

$$s(\tau,l)=\frac{1}{N}\sum_{(x,z,y)\in\mathcal{D}}\Delta\ell^l_\tau(x,\tilde{z},y). \qquad (3)$$

## 4 V-SEAM

Figure 2 provides an overview of our proposed two-stage framework. In the first stage, V-SEAM applies visual semantic editing on local image regions and layerwise activation patching to identify key modules and their functional roles in VLMs for visual semantic understanding (§4.1). Building on these insights, the second stage focuses on attention modulation, identifying critical attention heads that positively or negatively affect visual semantic understanding and prediction, and enabling targeted intervention via attention head rescaling to enhance VLM performance (§4.2).

### 4.1 Visual Semantic Editing

As shown in Figure 1, current visual intervention methods often rely on coarse perturbations that lack visual semantic precision, making it infeasible to isolate the causal contributions of specific visual semantic elements such as color, object identity, or spatial relations. Moreover, strong additive perturbations can push input data out of the model's training data distribution, leading to unreliable or misleading attributions (Zhang and Nanda, 2023). To address these limitations, we propose a *visual semantic editing* strategy that edits only the image regions corresponding to the question's semantics. Combined with activation patching, this allows us to assess whether the model's prediction can be causally restored, thereby tracing how semantic signals propagate across layers and modules.

**Semantic Editing Prompt Generation.** We construct counterfactual variants of the original question by altering attributes, objects, or relations. We manually design 10 reference examples and use GPT-4o to generate contextually plausible semantic substitutions (e.g., "red" to "green"). To ensure semantic clarity and factual accuracy, we manually verify the generated questions. Based on our verification, less than 1% cases were filtered out due to abstract descriptions (e.g., "bright" as a color), which are difficult to visualize. Each validated question then serves as a semantic editing prompt, which locates the corresponding image region and applies the intended semantic change. Detailed prompts can be found in Appendix §A.1.

**Semantic Region Editing.** For each question, to enable precise editing of a targeted semantic region while preserving the rest of the image, we require bounding box information for the region of interest, obtained either from the dataset annotations or an off-the-shelf object detection model. We then apply an image segmentation tool, i.e., SAM (Kirillov et al., 2023), to segment the target region and use an image editing tool, i.e., PowerPaint (Zhuang et al., 2024), to locally edit the image, guided by the generated counterfactual prompts, ensuring that only the intended semantics are modified. We manually inspected the visual edits, filtering out ∼10% due to unintended modifications outside the targeted region or unsuccessful edits.

**VQA Triple Selection.** To ensure the causal effectiveness of our interventions, we retain only VQA triples where the model correctly predicts the ground-truth answer $y$ on the clean image input

$(x, z)$ but fails on the corresponding perturbed input $(x, \tilde{z})$, with $z$ denoting the original image and $\tilde{z}$ the edited one. Specifically, the selected samples must satisfy the following condition:

$$f_\theta(x, z) = y, \quad f_\theta(x, \tilde{z}) \neq y, \tag{4}$$

where we compare the model predictions $f_\theta(x, z)$ and $f_\theta(x, \tilde{z})$ with the given correct answer $y$. This filtering criterion ensures that the observed causal effect is both significant and meaningful.

### 4.2 Attention Modulating

While activation patching is effective for identifying causal layers and token spans, it is limited in granularity and efficiency at the attention head level. First, token-wise patching for each head is computationally expensive. Second, activation-based methods are more suited to identifying helpful structures, but may potentially overlook harmful components that negatively impair visual understanding. To address these issues, we propose a head-level causal *attention modulating* strategy that effectively identifies attention heads critical to visual comprehension, enabling targeted rescaling to enhance model performance.

**Key Attention Head Identification.** The first step is identifying which attention heads are causally responsible for improving or distracting the model's prediction performance. Specifically, let $\boldsymbol{A}_h^l \in \mathbb{R}^{T \times d_h}$ denote the output embedding of the attention head $h \in [1, H]$ from layer $l \in [1, L]$, where $d_h$ is the attention head dimension. We mask an attention head by replacing its output embedding with the average output embedding of the remaining heads at the same layer $l$:

$$\widetilde{\boldsymbol{A}}_h^l = \frac{1}{H-1} \sum_{h'=1, h' \neq h}^{H} \boldsymbol{A}_{h'}^l \tag{5}$$

Intuitively, this operation preserves the overall structure of the layer while masking the semantic contribution of head $h$, enabling us to isolate its causal effect. To quantify this effect on a VQA triple $(x, z, y)$, we measure the change in the model's prediction probability of the correct answer token $y$ before and after masking this attention head. Specifically, we pass the masked embedding $\widetilde{\boldsymbol{A}}_h^l$ along with the other attention heads' embeddings $\boldsymbol{A}_{h'}^l, \forall h' \in [1, H] \wedge h' \neq h$ to the rest of the VLM. This yields a new prediction probability for the correct answer, denoted as $p_\theta(y|x, z; \widetilde{\boldsymbol{A}}_h^l)$. For comparison, we also compute the original prediction probability without attention head masking, denoted as $p_\theta(y|x, z)$. The causal effect of head $h$ at layer $l$ is then defined as the change in prediction probability after masking:

$$\Delta p_h^l(x, z, y) = p_\theta(y|x, z; \widetilde{\boldsymbol{A}}_h^l) - p_\theta(y|x, z) \tag{6}$$

To identify the positive and negative effects of a head over a dataset $\mathcal{D}$, we split the dataset into two subsets: the VQA triples predicted by the model correctly and incorrectly, i.e., $\mathcal{D}_{\text{correct}}$ and $\mathcal{D}_{\text{incorrect}}$. For head $h$ at layer $l$, we compute the average probability change over both subsets:

$$c_\kappa(h, l) = \frac{1}{|\mathcal{D}_\kappa|} \sum_{(x,z,y) \in \mathcal{D}_\kappa} \Delta p_h^l(x, z, y), \tag{7}$$

where $\kappa = \{\text{correct}, \text{incorrect}\}$. Ranking heads by these scores allows us to identify two sets of heads:

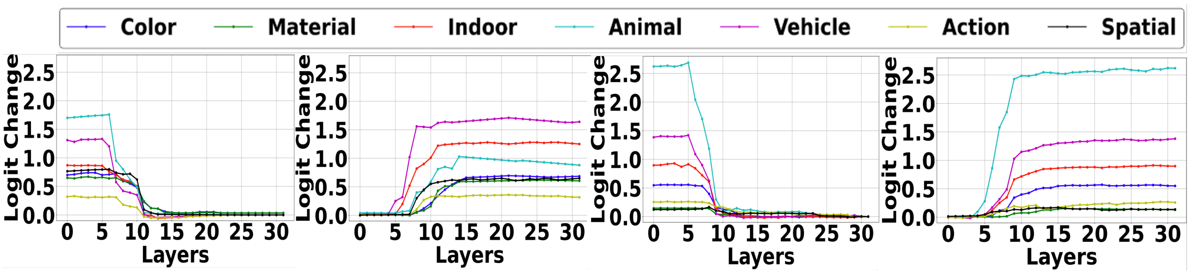
- **Positive Heads.** $\mathcal{H}_{\text{pos}}$ contains the top-$K$ heads with the highest values of $-c_{\text{correct}}(h, l)$. Masking these heads leads to the largest drop in prediction probability on $\mathcal{D}_{\text{correct}}$, indicating their positive causal effect on correct predictions.
- **Negative Heads.** $\mathcal{H}_{\text{neg}}$ contains the top-$K$ heads with the highest values of $c_{\text{incorrect}}(h, l)$. Masking these heads leads to the largest gain in prediction probability on $\mathcal{D}_{\text{incorrect}}$, indicating their negative causal effect on incorrect predictions.

To align with the three semantic levels, i.e., attributes, objects, and relations, we perform attention head identification separately for each semantic task type, using the corresponding VQA triples. This allows us to identify positive and negative heads that contribute either to general-purpose understanding or to specific semantic categories.

**Attention Head Rescaling.** To verify the causal effect of the heads in $\mathcal{H}_{\text{pos}}, \mathcal{H}_{\text{neg}}$, we propose an *attention head rescaling* strategy that modulates the influence of these heads to improve visual semantic understanding during inference.

First, we measure an importance score reflecting the absolute amount of probability changes for the positive and negative heads.

$$c(h, l) = \begin{cases} |c_{\text{correct}}(h, l)|, & \forall (h, l) \in \mathcal{H}_{\text{pos}} \\ |c_{\text{incorrect}}(h, l)|, & \forall (h, l) \in \mathcal{H}_{\text{neg}} \end{cases} \tag{8}$$



(a) LLaVA: visual-level (b) LLaVA: question-level (c) InstructBLIP: visual-level (d) InstructBLIP: question-level

Figure 3: Logit change analysis for image and query token patching in LLaVA and InstructBLIP.

Next, we normalize the score to a range of $[0, 1]$, converting them to a rescaling factor:

$$\lambda(h, l) = \frac{c(h, l) - c_{\min}}{c_{\max} - c_{\min}} \in [0, 1], \tag{9}$$

where $c_{\min}, c_{\max}$ are the minimum and maximum values of $c(h, l)$ computed separately for heads in $\mathcal{H}_{\text{pos}}$ and $\mathcal{H}_{\text{neg}}$. Finally, during inference, we rescale the embeddings $\boldsymbol{A}_h^l$ of head $h$ at layer $l$ based on their normalized weights.

$$\widetilde{\boldsymbol{A}}_h^l = \begin{cases} (1 + \lambda(h, l))\boldsymbol{A}_h^l, & (h, l) \in \mathcal{H}_{\text{pos}}, \\ (1 - \lambda(h, l))\boldsymbol{A}_h^l, & (h, l) \in \mathcal{H}_{\text{neg}}. \end{cases} \tag{10}$$

The key idea is to amplify attention signals from positive heads while suppressing those from negative heads. Importantly, we modify only the outputs of selected heads during the forward pass, leaving the model architecture and parameters unchanged.

## 5 VLM Interpretations via V-SEAM

### 5.1 Experimental Setting

In this study, we consider both VLM families, with a specific focus on two state-of-the-art VLMs: **LLaVA 1.5 7B** (Li et al., 2024) and **InstructBLIP 7B** (Dai et al., 2023). Regarding data, we apply V-SEAM using the GQA benchmark (Hudson and Manning, 2019), one of the largest VQA datasets (22M instances), featuring diverse and high-quality annotations of objects, attributes, and relationships across both visual and textual modalities. This makes GQA well-suited for our visual semantic editing to generate corrupted images. Considering this study's scope, we subsample the dataset based on two criteria: (1) we focus on instances involving concepts from high-frequency semantic categories to ensure representativeness; and (2) we prioritize binary, discriminative questions that target a specific semantic property and yield unambiguous answers, reducing data-centric noise in causal tracing analysis. We further balance the distribution of binary question types to ensure equal representation. Following the standard practice of prior work (Palit et al., 2023; Golovanevsky et al., 2025), we sample 12,647 questions in total. Table 1 displays the detailed data statistics. Example questions are shown in Appendix B.

| Concept-Level | VQA Category | Size |
|---|---|---|
| Attribute | Material | 1,300 |
| Attribute | Color | 1,500 |
| Object | Animal | 1,070 |
| Object | Vehicle | 1,740 |
| Object | Indoor | 2,092 |
| Relation | Spatial | 1,950 |
| Relation | Action | 2,995 |

Table 1: Dataset sizes for each VQA task categorized by Concept-Level, sorted by size within each Concept.

### 5.2 Key Findings

**Multimodal information transfer.** Figure 3 displays the results of information transfer across modalities. Overall, both VLMs start with understanding visual information in the early layers (0-5), then perform visual-text alignment in the middle layers (6–15), and finally shift focus toward the question text for answering in the later layers (>15). Comparatively, text processing requires more layers compared to visual understanding, indicating that VLMs use more parametric memory on the text side. Among semantic categories, both models align object-related information across modalities earlier than relational and attribute information. This pattern is similar to the progress in human visual-language comprehension (Ullman, 1987).

**Self-attention and MLP contributions.** Given the higher parametric memory demands of text processing compared to visual understanding to answer visual questions, we further analyze key modules (i.e., self-attention and MLP) in the language model within VLMs. Specifically, we apply activation patching (Meng et al., 2022) to replace

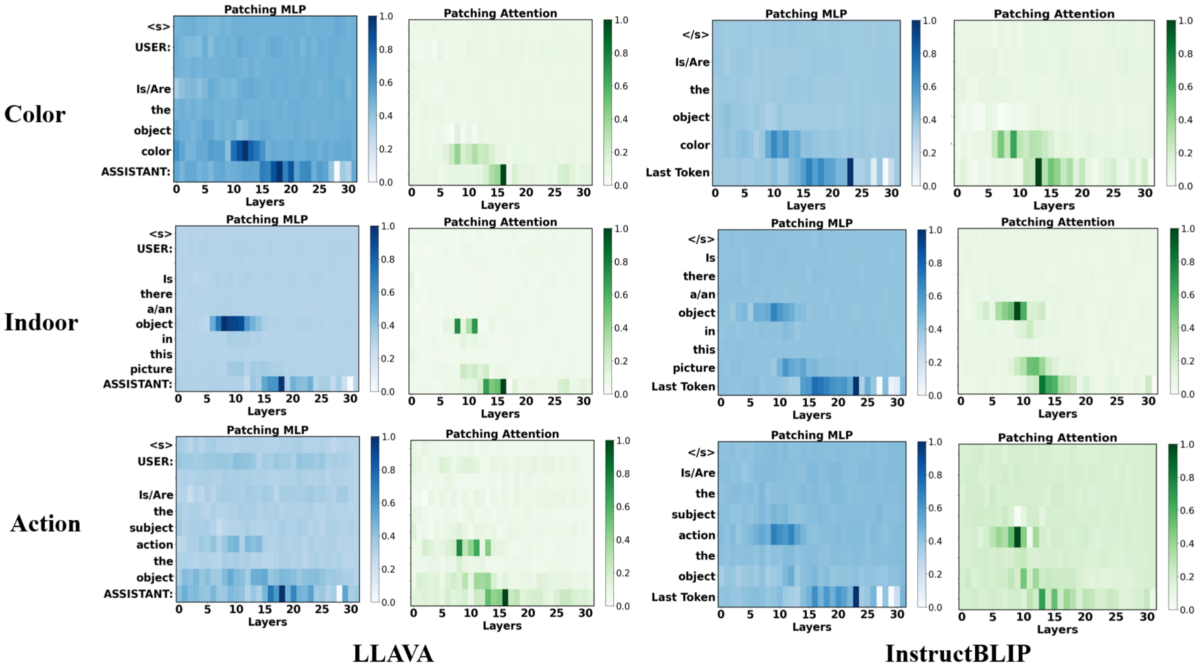


Figure 4: Layer-wise causal impact of MLP (blue) and self-attention (green) on cross-modal semantic understanding. Columns denote transformer layers, rows denote question tokens from a fixed prompt (e.g., "Is the object red?"), and color intensity indicates the $\Delta$logit gain of the correct answer after patching.

corrupted question token embeddings with their clean counterparts at different layers, and measure the logit difference of the correct answer. Figure 4 shows the results. We observe that both MLP and self-attention store the salient information of target aspect-related tokens (e.g., "object") in the middle layers (6–12). Looking into the last token, we find that the most influential self-attention layer for answering the question appears earlier (e.g., Layer 16 in LLaVA and Layer 13 in InstructBLIP) compared to the MLP module (e.g., Layer 18 in LLaVA and Layer 23 in InstructBLIP). This suggests that attention contributes more to semantic retrieval at mid-depth, whereas MLPs carry a stronger influence on final decision-making in deeper layers.

To gain deeper insight into the semantic information captured by each module, we project layer-wise self-attention and MLP activations into the vocabulary space using LogitLens (Neo et al., 2024).

As shown in Figure 5, critical self-attention layers primarily capture answer cues (e.g., "hold", "woman", and "ski"), while MLP layers directly focus on the final answer (e.g., "yes" or "no"). Additional examples are shown in Appendix D.

**Key attention heads.** Given the role of self-attention layers in identifying answer-relevant evidence, we further identify the key attention heads responsible for capturing this information. Table 2 lists the most salient heads across semantic levels for both VLMs. Notably, we identify both *positive heads* with stronger causal effects on correct predictions and *negative heads* that introduce more misleading signals compared to randomly selected attention heads. Overall, both types of heads are mutually exclusive, suggesting their distinct roles in VLMs. Interestingly, unlike negative heads that generalize across semantic levels, positive heads are typically shareable within each level but differ across levels. To validate our findings, we measure the spatial alignment between identified heads and object regions. As detailed in Appendix F, positive heads consistently attend to target-relevant areas, while negative heads often focus on distracting background regions.

## 6 Attention Head Modulating Assessment

**Effectiveness** We systematically evaluate the impact of editing each identified key attention head on the sampled GQA data, accounting for semantic levels and functional polarity (i.e., positive vs. negative contribution). As shown in Table 3, both VLMs achieve notable performance gains ($\sim$5% on average) after editing, with the highest improvements on relation-level questions and the lowest on object-level ones. To measure the significance of performance differences between our editing

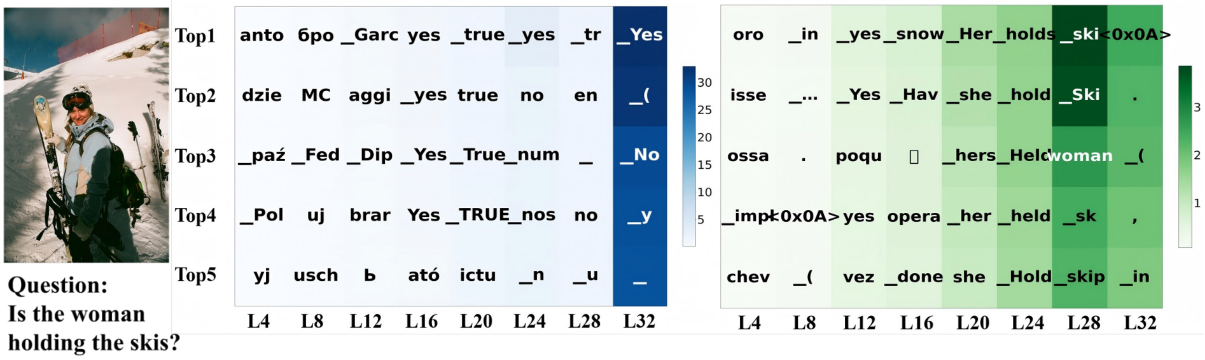


Figure 5: Case study of the MLP (blue) and self-attention (green) logit lens in LLaVA on the Action VQA task. Each row denotes a top-5 predicted token, and each column corresponds to a transformer layer. Color intensity reflects the logit value.

| **Semantic Type** | **LLaVA** | | **InstructBLIP** | |
|---|---|---|---|---|
| | **Positive Heads** | **Negative Heads** | **Positive Heads** | **Negative Heads** |
| **Attribute** | L16.H1, L15.H7, L22.H11 | L31.H11, L30.H11, L29.H11 | L11.H27, L6.H30 | L0.H30, L10.H23, L6.H7, L6.H10, L7.H2, L30.H30 |
| **Object** | L0.H11, L26.H11, L11.H7 | L31.H11, L30.H11, L29.H11, L12.H2, L7.H2, L9.H27 | L11.H28, L7.H20, L23.H30, L12.H31 | L0.H30, L6.H10, L10.H9, L30.H30, L12.H25 |
| **Relation** | L13.H8, L14.H15 | L31.H11, L30.H11, L29.H11, L0.H11 | L23.H16, L15.H22, L31.H19, L11.H5, L9.H28, L7.H19, L8.H28 | L12.H25, L10.H12, L6.H27, L30.H30 |

Table 2: Shared positive and negative attention heads across semantic types for LLaVA and InstructBLIP. Attention heads are formatted as Layer.Head (e.g., L26.H11).

| **Model** | **Ablation** | **Attribute** | **Object** | **Relation** | **Avg** |
|---|---|---|---|---|---|
| LLaVA | original | 77.49 | 93.37 | 83.29 | 84.72 |
| | w/o negative | 81.75 | 94.86 | 88.93 | 88.51 |
| | w/o positive | 75.01 | 92.27 | 79.75 | 82.34 |
| | random remove | 77.46 | 93.24 | 83.00 | 84.57 |
| | rescaling | **82.79** | **94.99** | **90.66** | **89.48** |
| InstructBLIP | original | 79.46 | 93.12 | 88.55 | 87.04 |
| | w/o negative | 82.78 | 95.24 | 94.52 | 90.85 |
| | w/o positive | 77.85 | 91.08 | 77.84 | 82.26 |
| | random remove | 79.34 | 93.04 | 89.50 | 87.29 |
| | rescaling | **84.06** | **95.67** | **95.21** | **91.65** |

Table 3: Accuracy (%) of VLMs on VQA under different attention head editing strategies."original" uses all attention heads. "w/o negative" and "w/o positive" remove the top-10 negative or positive heads, respectively. "random remove" randomly drops 10 heads. "rescaling" denotes head embedding rescaling.

and each baseline, we conduct paired $t$-tests on 1,000 bootstrap-sampled folds per semantic category. As shown in Appendix H, our rescaling strategy achieves statistically significant improvements ($p < 0.001$) over all four baselines across three semantic levels.

**Data Dependence** To assess the method's data dependence, we randomly sample 10–50% of the experimental data for key head identification and editing, then test on the full set. Each setting is repeated 10 times to ensure stability, and we report the average results. As shown in Table 4, our method achieves comparable performance even with just 10% of the data, highlighting its applicability in low-resource scenarios.

| **Category** | **Initial** | **10%** | **20%** | **30%** | **50%** | **Full** |
|---|---|---|---|---|---|---|
| Attribute | 77.49 | 79.48 | 81.17 | 82.20 | 82.50 | **82.79** |
| Object | 90.18 | 92.42 | 93.20 | 93.24 | **93.26** | 93.25 |
| Relation | 83.03 | 87.01 | 90.96 | 91.02 | 91.07 | **91.11** |
| Average | 83.57 | 86.30 | 88.44 | 88.82 | 88.94 | **89.05** |

Table 4: Accuracy (%) of **LLaVA** under different sample proportions for attention head embedding rescaling, grouped by task categories. Bold values indicate best performance for each category.

**Generalizability** We further evaluate the generalizability of attention editing on two out-of-distributional (OOD) benchmarks. One is POPE (Li et al., 2023b), which similarly involves binary questions covering three subsets: popular, adversarial, and random. Table 5 compares the unedited LLaVA with models edited using in-distribution POPE data (20%) and GQA data. Both edited versions outperform the unedited model, with the GQA-edited variant even slightly surpassing the one trained on in-distribution data. The

| Task | Metric | Initial | Edit_POPE | Edit_GQA |
|---|---|---|---|---|
| Popular | Acc | 0.858 | 0.862 | **0.866** |
|  | Recall | 0.767 | 0.785 | **0.798** |
|  | F1 | 0.844 | 0.857 | **0.859** |
| Adversarial | Acc | 0.835 | 0.838 | **0.843** |
|  | Recall | 0.767 | 0.792 | **0.795** |
|  | F1 | 0.823 | **0.832** | 0.831 |
| Random | Acc | 0.867 | 0.880 | **0.885** |
|  | Recall | 0.767 | 0.795 | **0.800** |
|  | F1 | 0.856 | 0.875 | **0.878** |

Table 5: Performance of LLaVA on POPE under different rescaling strategies. "Initial" refers to the performance without any intervention. "Edit_POPE" refers to applying head rescaling based on POPE. "Edit_GQA" denotes using heads and scores discovered on GQA.

other one is COCOQA (Lu et al., 2016), which contains open-ended questions spanning color, object, and location categories. We directly evaluate the GQA-edited model against the unedited baseline, and as shown in Appendix H, both VLMs edited with GQA data continue to outperform the unedited version.

## 7 Conclusion

We present V-SEAM, a novel semantics-based causal interpretability framework for VLMs. We introduce visual semantic editing for concept-level visual interventions and propose attention modulation to identify and edit key attention heads. Through a systematic analysis of LLaVA and InstructBLIP, we find that attention primarily captures salient semantic cues (e.g., color) that guide predictions, while MLP layers are crucial for generating final outputs. We identify top attention heads playing either positive or negative roles. Our attention editing method not only improves model performance but also remains effective in low-resource settings and generalizes well to OOD cases.

## Limitations

Despite the insights presented in this study, several limitations remain:

1. To ensure the controllability of causal interventions, we focus primarily on binary discriminative questions with uniform formats (e.g., “Is the object red?”). While this design facilitates precise patching and attribution, it limits our ability to interpret model behavior in more complex reasoning tasks such as counting, sorting, or multi-hop inference. We leave these aspects for future exploration.
2. Activation patching requires multiple forward passes, especially when intervening across many layers, tokens, or attention heads, which incurs significant computational cost. This limits the scalability of our method to larger models and datasets. Future work may explore more efficient patching strategies to reduce inference time.
3. While we identify reusable “positive heads” and “negative heads” across various semantic tasks, the mechanisms underlying attention heads in VLMs remain underexplored. For example, it is still unclear whether attention heads interact in cooperative or competitive ways, or whether cross-modal heads exhibit domain-specific dynamic routing. Further analysis is required to systematically understand these behaviors.

## References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, and Iain Barr. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683.

Yuxuan Bai, Hao Cheng, Yuwei Gu, and 1 others. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2310.07904*.

Samyadeep Basu, Martin Grayson, Cecily Morrison, Besmira Nushi, Soheil Feizi, and Daniela Massiceti. 2024. Understanding information storage and transfer in multi-modal large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Gabriela Ben Melech Stan, Estelle Aflalo, Raanan Yehezkel Rohekar, Anahita Bhiwandiwalla, Shao-Yen Tseng, Matthew Lyle Olson, Yaniv

Gurwicz, Chenfei Wu, Nan Duan, and Vasudev Lal. 2024. Lvlm-interpret: An interpretability tool for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8182–8187.

Blair Bilodeau, Natasha Jaques, Pang Wei Koh, and Been Kim. 2024. Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences*, 121(2):e2304406120.

Shiqi Chen, Tongyao Zhu, Ruochen Zhou, Jinghan Zhang, Siyang Gao, Juan Carlos Niebles, Mor Geva, Junxian He, Jiajun Wu, and Manling Li. 2025. Why is spatial reasoning hard for vlms? an attention mechanism perspective on focus areas. *arXiv preprint arXiv:2503.01773*.

Weijie Chen, Yizhe Zhang, Qian Wu, and 1 others. 2024. Internvl: Scaling up vision-language pretraining with multimodal reinforcement learning. *arXiv preprint arXiv:2402.00028*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS) 36*.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. 2023. Interpreting clip's image representation via text-based decomposition. *arXiv preprint arXiv:2310.05916*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*.

Michal Golovanevsky, William Rudman, Vedant Palit, Carsten Eickhoff, and Ritambhara Singh. 2025. What do vlms notice? a mechanistic interpretability pipeline for gaussian-noise-free text-image corruption and evaluation. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11462–11482, Albuquerque, New Mexico. Association for Computational Linguistics.

Lisa Anne Hendricks and Aida Nematzadeh. 2021. Probing image-language transformers for verb understanding. *arXiv preprint arXiv:2106.09141*.

Kaichen Huang, Jiahao Huo, Yibo Yan, Kun Wang, Yutao Yue, and Xuming Hu. 2024. Miner: Mining the underlying pattern of modality-specific neurons in multimodal large language models. *arXiv preprint arXiv:2410.04819*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Jiahao Huo, Yibo Yan, Boren Hu, Yutao Yue, and Xuming Hu. 2024. Mmneuron: Discovering neuron-level domain-specific interpretation in multimodal large language model. *arXiv preprint arXiv:2406.11193*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, and 1 others. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Junnan Li, Dongxu Li, Yixuan Xie, Yixuan Guo, Xiyang Dai, Jianfeng Gao, Jianwei Sang, and Lijuan Wang. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Zihao Lin, Samyadeep Basu, Mohammad Beigi, Varun Manjunatha, Ryan A Rossi, Zichao Wang, Yufan Zhou, Sriram Balasubramanian, Arman Zarei, Keivan Rezaei, and 1 others. 2025. A survey on mechanistic interpretability for multi-modal foundation models. *arXiv preprint arXiv:2502.17516*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Y Liu, Y Zhang, and S Yeung-Levy. 2025. Mechanistic interpretability meets vision language models: Insights and limitations. In *The Fourth Blogpost Track at ICLR 2025*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. *Advances in neural information processing systems*, 29.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. 2024. Towards interpreting visual information processing in vision-language models. *arXiv preprint arXiv:2410.07149*.

Vedant Palit, Rohan Pandey, Aryaman Arora, and Paul Pu Liang. 2023. Towards vision-language mechanistic interpretability: A causal tracing tool for blip. In *Proceedings of the ICCV Workshop on Computational Linguistics for Vision and Language (CLVL)*.

Ekta Sood, Fabian Kögel, Philipp Müller, Dominike Thomas, Mihai Bâce, and Andreas Bulling. 2023. Multimodal integration of human-like attention in visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2648–2658.

Student. 1908. The probable error of a mean. *Biometrika*, pages 1–25.

Shimon Ullman. 1987. Visual routines. In *Readings in computer vision*, pages 298–328. Elsevier.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, and 1 others. 2024. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499.

Xiaoying Xing, Chia-Wen Kuo, Li Fuxin, Yulei Niu, Fan Chen, Ming Li, Ying Wu, Longyin Wen, and Sijie Zhu. 2025. Where do large vision-language models look at when answering questions? *arXiv preprint arXiv:2503.13891*.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588.

Xiaofeng Zhang, Chen Shen, Xiaosong Yuan, Shaotian Yan, Liang Xie, Wenxiao Wang, Chaochen Gu, Hao Tang, and Jieping Ye. 2024. From redundancy to relevance: Enhancing explainability in multimodal large language models. *arXiv e-prints*, pages arXiv–2406.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. 2024. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pages 260–278. Springer.

Deyao Zhu, Xiangxin Zhou, Xiang Wang, Xiubo Geng, Fan Liu, and Jiashen Zhu. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. 2024. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pages 195–211. Springer.

# A Prompt Template

## A.1 Semantic Perturbation Prompts

To support causal editing in the visual domain, we design semantic perturbation prompts that precisely identify which part of the question's meaning should change. By isolating a single semantic unit—such as an attribute, object, or relation—this strategy provides a grounded and logically consistent basis for local image edits via PowerPaint (Zhuang et al., 2024). It also ensures that the intended visual intervention aligns with a meaningful linguistic transformation, enabling controlled and interpretable reasoning analysis.

To automate and accurately generate semantically valid counterfactual questions, we leverage GPT-4o as a language engine. To guide it in producing controlled perturbations, we adopt a few-shot prompting strategy. Specifically, we manually construct 10 reference examples covering three semantic types—attributes, objects, and relations. These examples are used as in-context demonstrations to instruct the model on how to minimally alter a question's semantic unit while preserving its grammaticality and contextual plausibility.

The prompt consists of three parts: (1) a task-specific instruction, (2) a few in-context examples

| Semantic Type | Original Question | Answer | Full Answer | Counterfactual Question |
|---|---|---|---|---|
| Attribute | Is the shirt **blue**? | No | The shirt is black. | Is the shirt **black**? |
| Attribute | Is the chair made of **wood**? | Yes | The chair is made of wood. | Is the chair made of **metal**? |
| Attribute | Is the car **black**? | No | The car is white. | Is the car **white**? |
| Object | Is there a **dog** in this picture? | No | There is a cat in the image. | Is there a **cat** in this picture? |
| Object | Is there a **chair** in this picture? | Yes | There is a chair in the image. | Is there a **table** in this picture? |
| Object | Is there a **bus** in this picture? | No | A bicycle is in the image. | Is there a **bicycle** in this picture? |
| Relation | Is the person **riding** the horse? | Yes | The person is riding the horse. | Is the person **feeding** the horse? |
| Relation | Is the person **holding** a tennis racket? | Yes | The person is holding a tennis racket. | Is the person **throwing** a tennis racket? |
| Relation | Is the cat to the **left** of the sofa? | Yes | The cat is positioned to the left of the sofa. | Is the cat to the **right** of the sofa? |
| Relation | Is the ball **under** the table? | No | The ball is on top of the table. | Is the ball **on top of** the table? |

Table 6: Representative examples used for semantic perturbation. Each row shows the original binary question with its answer and full natural-language explanation, followed by a counterfactual version with a modified semantic unit (in bold).

for the corresponding semantic type, and (3) a new question to be rewritten. In Table 6, we show the complete set of manually created reference examples. These examples serve as the foundation for constructing few-shot prompts, enabling GPT-4o to learn how to generate semantically valid counterfactual questions. Note that in Table 6, the `Answer` field corresponds to the original binary label provided by the dataset, while the `Full Answer` is a natural-language explanation included in the GQA annotations to justify the label. We also incorporate them as contextual guidance when generating counterfactual questions, helping to ensure semantic plausibility.

Table 7 provides representative few-shot prompt templates used for each semantic type. The prompts are formatted to clearly separate the original question, the full answer, and the desired counterfactual question. The final line always instructs the model to generate only the rewritten question.

We also applied human filtering to ensure that the model-generated counterfactuals are semantically plausible, grammatically correct, and minimally modified from the original input. All generated questions were manually verified, and less than 1% were filtered out due to abstract or ambiguous expressions (e.g., “bright” as a color) that cannot be reliably grounded in visual content.

### A.2 Prompt Templates for Evaluation

To ensure consistent and interpretable input formatting across evaluation settings, we design task-specific prompt templates for both LLaVA and InstructBLIP. Each template is tailored to the corresponding experimental step, accounting for differences in task format and model instruction style.

For binary judgment tasks—such as activation patching and other classification-based interventions—we adopt a unified yes/no prompt format that explicitly instructs the model to answer concisely. For open-ended tasks like COCOQA, which require free-form answers, we apply lightweight role descriptions and prompt the model to generate concise responses (e.g., single words or phrases). We also align the prompt style with each model’s native instruction tuning paradigm—for instance, LLaVA follows a chat-style interaction between USER and ASSISTANT, while InstructBLIP uses standalone declarative prompts.

Table 8 summarizes the full set of prompt templates used in our evaluation pipeline, organized by task type and model.

## B Semantic Benchmark Details

To ensure the feasibility of controlled causal intervention and minimize semantic ambiguity, we construct fine-grained subsets from the GQA dataset based on the following principles:

- **Question Type Filtering:** We retain only binary, fact-based questions with a uniform format and clear Yes/No answers. These typically involve single entities and single predicates, such as “Is the object red?” or “Is there a dog in this picture?”.
- **High-Frequency Semantics:** Within each semantic category—attributes, objects, and relations—we select commonly occurring subtypes to ensure sufficient coverage, reduce sparsity, and support statistically robust evaluation.
- **Structural Consistency:** We retain only syntactically simple and semantically clear

| **Strategy** | **Examples** |
|---|---|
| Attribute Perturbation<br>Few-shot Prompt | **You are given a binary Yes/No question about an object's attribute, along with its ground-truth answer and a full explanation.** Rewrite the question by changing the attribute (e.g., color or material) while preserving the original structure and context.<br>**Example 1:**<br>`Original Question: Is the shirt blue? Answer: No. Full Answer: The shirt is black. Counterfactual Question: Is the shirt black?`<br>**Example 2:** `Original Question: Is the chair made of wood? Answer: Yes. Full Answer: The chair is made of wood. Counterfactual Question: Is the chair made of metal?`<br>**Example 3:** `Original Question: Is the car black? Answer: No. Full Answer: The car is white. Counterfactual Question: Is the car white?`<br>**Now please complete the following. Only output the rewritten counterfactual question, without explanation:** `Original Question: Is the sofa red? Answer: Yes. Full Answer: The sofa is red. Counterfactual Question:` |
| Object Perturbation<br>Few-shot Prompt | **You are given a binary Yes/No question about the presence of an object in an image, along with its answer and a full explanation.** Rewrite the question by changing the queried object to another semantically plausible one.<br>**Example 1:** `Original Question: Is there a dog in this picture? Answer: No. Full Answer: There is a cat in the image. Counterfactual Question: Is there a cat in this picture?`<br>**Example 2:** `Original Question: Is there a chair in this picture? Answer: Yes. Full Answer: There is a chair in the image. Counterfactual Question: Is there a table in this picture?`<br>**Example 3:** `Original Question: Is there a bus in this picture? Answer: No. Full Answer: A bicycle is in the image. Counterfactual Question: Is there a bicycle in this picture?`<br>**Now please complete the following. Only output the rewritten counterfactual question, without explanation:** `Original Question: Is there a lamp in this picture? Answer: No. Full Answer: There is a shelf in the image. Counterfactual Question:` |
| Relation Perturbation<br>Few-shot Prompt | **You are given a binary Yes/No question about a spatial or action relationship between objects, along with its ground-truth answer and full explanation.** Rewrite the question by changing the relation (e.g., left to right, under to above, riding to feeding).<br>**Example 1:** `Original Question: Is the person riding the horse? Answer: Yes. Full Answer: The person is riding the horse. Counterfactual Question: Is the person feeding the horse?`<br>**Example 2:** `Original Question: Is the person holding a tennis racket? Answer: Yes. Full Answer: The person is holding a tennis racket. Counterfactual Question: Is the person throwing a tennis racket?`<br>**Example 3:** `Original Question: Is the cat to the left of the sofa? Answer: Yes. Full Answer: The cat is positioned to the left of the sofa. Counterfactual Question: Is the cat to the right of the sofa?`<br>**Example 4:** `Original Question: Is the ball under the table? Answer: No. Full Answer: The ball is on top of the table. Counterfactual Question: Is the ball above the table?`<br>**Now please complete the following. Only output the rewritten counterfactual question, without explanation:** `Original Question: Is the cup next to the book? Answer: No. Full Answer: The cup is on top of the book. Counterfactual Question:` |

Table 7: Few-shot prompt strategies for semantic perturbation. Each cell lists in-context examples guiding GPT-4o to produce counterfactual binary questions by altering one semantic unit.

| Step | Model | Prompt Template |
|---|---|---|
| Activation Patching (Binary Judgment) | LLaVA | **USER:**<br><Image><br>{Role Description}*<br>Question: {Question}<br>Please answer the question using Yes or No.<br>**ASSISTANT:** |
| | InstructBLIP | <Image><br>{Role Description}*<br>Question: {Question}<br>Please answer the question using yes or no. |
| Closed-form Binary Judgment Tasks (e.g., Color, Material, Indoor, POPE) | LLaVA | **USER:**<br><Image><br>{Role Description}*<br>Question: {Question}<br>Please answer the question using Yes or No.<br>**ASSISTANT:** |
| | InstructBLIP | <Image><br>{Role Description}*<br>Question: {Question}<br>Please answer the question using yes or no. |
| Open-ended Tasks (e.g., COCOQA) | LLaVA | A chat between a curious user and an artificial intelligence assistant.<br>**USER:**<br><Image><br>{Role Description}*<br>Question: {Question}<br>Context: N/A<br>Answer the question using a single word or phrase.<br>**ASSISTANT:** |
| | InstructBLIP | <Image><br>{Role Description}*<br>Question: {Question}<br>Short Answer: |

Table 8: Prompt templates used in different experimental steps for both LLaVA and InstructBLIP. Binary judgment tasks (e.g., activation patching and classification subtasks) adopt yes/no format, while open-ended tasks like COCOQA require concise short-form answers.

questions, such as those with a single subject–verb–object structure. Questions involving three or more entities or compound constructions are excluded to ensure consistency in question type and support reliable semantic interventions.

- **Answer Balance:** Each semantic subset is curated to maintain a roughly equal distribution of Yes and No answers (approximately 50% each), thereby reducing label bias and enabling fairer evaluation of model behavior under semantic perturbation.
- **Benchmark Representativeness:** Our subset selection process is informed by prior interpretability work (Palit et al., 2023; Golovanevsky et al., 2025). Moreover, the sample size of each subset is comparable to or larger than those in related studies.

Based on the above selection principles, we organize the curated VQA questions into a three-level semantic hierarchy—attributes, objects, and relations—each capturing commonly encountered visual concepts in real-world images.

To enhance semantic coverage and ensure representativeness, we select seven high-frequency subtypes across the three semantic levels: for attributes, we include color and material; for objects, we include indoor items, animals, and vehicles; and for relations, we cover spatial and action relations. These subtypes span a wide range of visual semantics and support diverse yet interpretable evaluation.

Table 9 provides an overview of these subtypes along with representative question examples.

## C Comparison of Perturbation Methods

To validate the effectiveness of our proposed semantic perturbation method in preserving the global semantic consistency of the input image, we compare it against several commonly used perturbation strategies, including Gaussian noise, salt-and-pepper noise, masking, and diffusion-based image editing. Traditional methods typically apply random noise or local occlusion to the image, which often disrupts low-level features and structural coherence, thereby distorting the distribution of visual features extracted by the encoder. In con-

| Semantic Type | Example Question | Samples | Image Example |
|---|---|---|---|
| **Attribute** | | | |
| Material | Is the chair made of Wood? | 1,300 | |
| Color | Is the shirt blue? | 1,500 | |
| **Object** | | | |
| Animal | Is there a dog in this picture? | 1,070 | |
| Vehicle | Is there a bike in this picture? | 1,740 | |
| Indoor | Is there a mirror in this picture? | 2,092 | |
| **Relation** | | | |
| Spatial | Is the man right of the horse? | 1,950 | |
| Action | Is the woman holding the skis? | 2,995 | |

Table 9: Statistics and example images for each VQA task in our semantic benchmark. Semantic subtypes are grouped under attribute, object, and relation categories.

trast, our method leverages scene graphs and the SAM model, combined with state-of-the-art image editing techniques, to precisely modify only the targeted semantic region while leaving the rest of the image intact. As illustrated in Figures 6, 7, and 8, our semantic perturbation approach maintains global visual coherence while achieving fine-grained, interpretable edits across attribute-level, object-level, and relation-level tasks, respectively.

To quantitatively assess the distributional shift introduced by different perturbation methods, we extract global image features from both the original and perturbed images using the visual encoder employed by LLaVA (CLIP-ViT-L), and compute the cosine similarity between them. Theoretically, if a perturbation only modifies a localized semantic element, the global feature representation should remain highly similar to the original. In contrast, traditional perturbations such as noise or masking often cause significant degradation in this similarity. Tables 10, 11, and 12 report the average cosine similarity across three representative semantic tasks. Our method consistently yields the highest similarity, indicating minimal disturbance to global representations.

Experimental results demonstrate that our method consistently yields higher cosine similarity across all semantic tasks, indicating that it achieves effective semantic intervention with minimal disturbance to the global image distribution. In contrast, Gaussian noise, salt-and-pepper noise, masking, and full-image redrawing via diffusion models introduce substantial low-level feature distortions, leading to more pronounced shifts in the visual representation space.

## D Logit Lens Analysis of MLP and Attention Projections

To better understand the reasoning process within vision-language models, we apply the Logit Lens method to analyze the token prediction distributions across layers in the language models of LLAVA and InstructBLIP. Following the standard procedure described by (Neo et al., 2024), we compare the output logits projected from the MLP pathway and the attention output pathway at each transformer layer.

In VQA tasks (e.g., “Is the chair made of

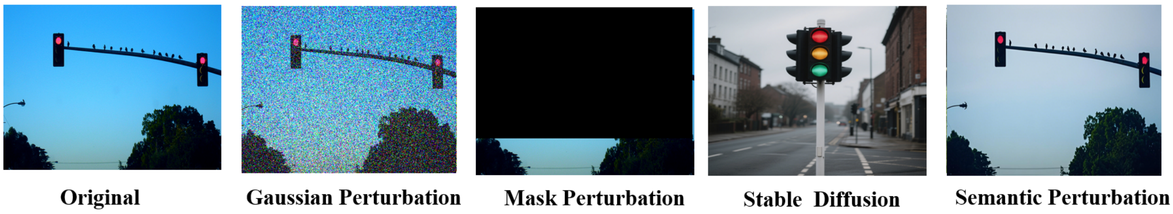


Figure 6: Visual comparison of different perturbation strategies for the object-level task ("Is the sky blue?"). Our semantic perturbation method (right) maintains global visual coherence while precisely intervening on the targeted semantic region (the sky). Traditional methods, including Gaussian noise, regional masking, and full-image redrawing with Stable Diffusion, introduce larger visual distortions and unintended semantic changes.

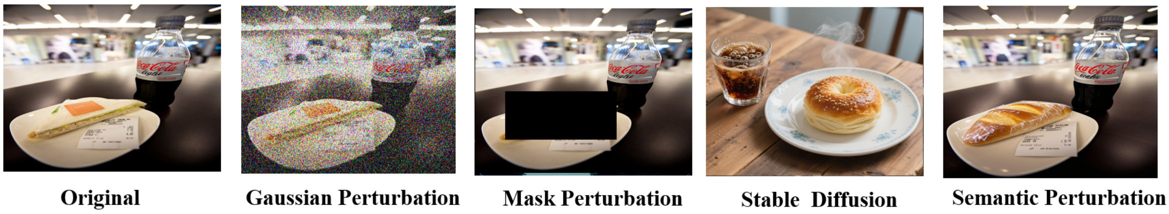


Figure 7: Visual comparison of different perturbation strategies for the object-level task ("Is there a sandwich in this picture?"). Our semantic perturbation method (right) maintains global visual coherence while precisely modifying the presence of the target object. Traditional methods, including Gaussian noise, regional masking, and full-image redrawing with Stable Diffusion, introduce significant visual distortion and lack fine-grained semantic control.

| Perturbation Method | Cosine Similarity |
|---|---|
| Gaussian Noise – Intensity 10 | 0.8766 |
| Gaussian Noise – Intensity 30 | 0.6333 |
| Gaussian Noise – Intensity 50 | 0.5025 |
| Gaussian Noise – Intensity 80 | 0.3446 |
| Salt-and-Pepper Noise – Intensity 10 | 0.7088 |
| Salt-and-Pepper Noise – Intensity 30 | 0.5554 |
| Salt-and-Pepper Noise – Intensity 50 | 0.4110 |
| Salt-and-Pepper Noise – Intensity 80 | 0.3142 |
| Masking | 0.6005 |
| Diffusion-Based Redrawing | 0.6891 |
| **Semantic Perturbation (Ours)** | **0.9168** |

Table 10: Cosine similarity between original and perturbed image features under different perturbation strategies. Feature representations are extracted from the vision encoder of the LLaVA model (CLIP-ViT-L) on the Color VQA dataset. Higher similarity indicates better preservation of global visual semantics, which is crucial for stable reasoning in real-world environments.

| Perturbation Method | Cosine Similarity |
|---|---|
| Gaussian Noise – Intensity 10 | 0.8628 |
| Gaussian Noise – Intensity 30 | 0.8148 |
| Gaussian Noise – Intensity 50 | 0.7206 |
| Gaussian Noise – Intensity 80 | 0.5988 |
| Salt-and-Pepper Noise – Intensity 10 | 0.7427 |
| Salt-and-Pepper Noise – Intensity 30 | 0.6729 |
| Salt-and-Pepper Noise – Intensity 50 | 0.6159 |
| Salt-and-Pepper Noise – Intensity 80 | 0.5840 |
| Masking | 0.6821 |
| Diffusion-Based Redrawing | 0.6205 |
| **Semantic Perturbation (Ours)** | **0.8970** |

Table 11: Cosine similarity between original and perturbed image features under different perturbation strategies. Feature representations are extracted from the vision encoder of the LLaVA model (CLIP-ViT-L) on the Indoor VQA dataset. Higher similarity indicates better preservation of global visual semantics, which is crucial for stable reasoning in real-world environments.

wood?”), we observe that in the early layers (Layers 2–7), both MLP and attention projections primarily output low-level or semantically ambiguous tokens, such as punctuation, garbled text, or subword fragments, indicating that meaningful semantic representations have not yet formed. Around Layer 9, the attention projection begins to produce answer-relevant tokens (e.g., `yes`), marking the onset of vision-language alignment. In contrast, the MLP projection only begins to consistently generate high-confidence answer tokens (e.g., `yes`, `true`, `number`) after Layer 15.

Between Layers 16–25, the attention projections increasingly focus on image-grounded semantic concepts (e.g., `wood`, `material`, `color`), exhibiting strong semantic sensitivity. Meanwhile, the MLP projections focus more on decision-related tokens, with steadily increasing confidence in out-

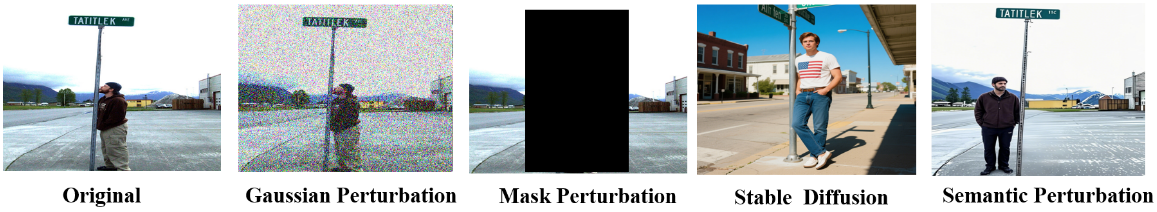


Figure 8: Visual comparison of different perturbation strategies for the relation-level task ("Is the man standing to the left of the signpost?"). Our semantic perturbation method (right) maintains global visual coherence while precisely modifying the spatial relationship between the target entities (man and signpost). In contrast, traditional approaches—such as Gaussian noise, regional masking, and full-image redrawing via Stable Diffusion—introduce substantial visual artifacts and fail to offer controlled, semantics-aware editing.

| Perturbation Method | Cosine Similarity |
|---|---|
| Gaussian Noise – Intensity 10 | 0.7921 |
| Gaussian Noise – Intensity 30 | 0.7430 |
| Gaussian Noise – Intensity 50 | 0.6204 |
| Gaussian Noise – Intensity 80 | 0.4877 |
| Salt-and-Pepper Noise – Intensity 10 | 0.7033 |
| Salt-and-Pepper Noise – Intensity 30 | 0.6182 |
| Salt-and-Pepper Noise – Intensity 50 | 0.5820 |
| Salt-and-Pepper Noise – Intensity 80 | 0.5194 |
| Masking | 0.4213 |
| Diffusion-Based Redrawing | 0.5586 |
| **Semantic Perturbation (Ours)** | **0.8542** |

Table 12: Cosine similarity between original and perturbed image features under different perturbation strategies. Feature representations are extracted from the vision encoder of the LLaVA model (CLIP-ViT-L) on the Action VQA dataset. Results show lower similarity compared to object-level tasks, reflecting higher sensitivity of action semantics to visual changes.

put logits. In the final layer, the MLP projection directly outputs the final answer token with high certainty, while the attention projection tends to focus on formatting or structural tokens (e.g., `<0x0A>`, `.`), reflecting its role in output structuring.

These findings support a stage-wise interpretation of the hierarchical reasoning process in VQA. Early layers are mainly involved in visual signal processing, the middle layers (Layers 8–16) are responsible for cross-modal alignment and semantic grounding, and the deeper layers (Layers 16–32) integrate language reasoning and answer generation. Our Logit Lens heatmap analysis (Figure 9 and 10) clearly illustrates this semantic progression: predicted tokens evolve from noisy or ambiguous outputs to aligned concepts and finally converge on task-specific, high-confidence answers. Moreover, the analysis reveals a clear division of labor across layers: attention mechanisms are primarily responsible for multimodal alignment and semantic construction, while MLP projections focus on decision-making.

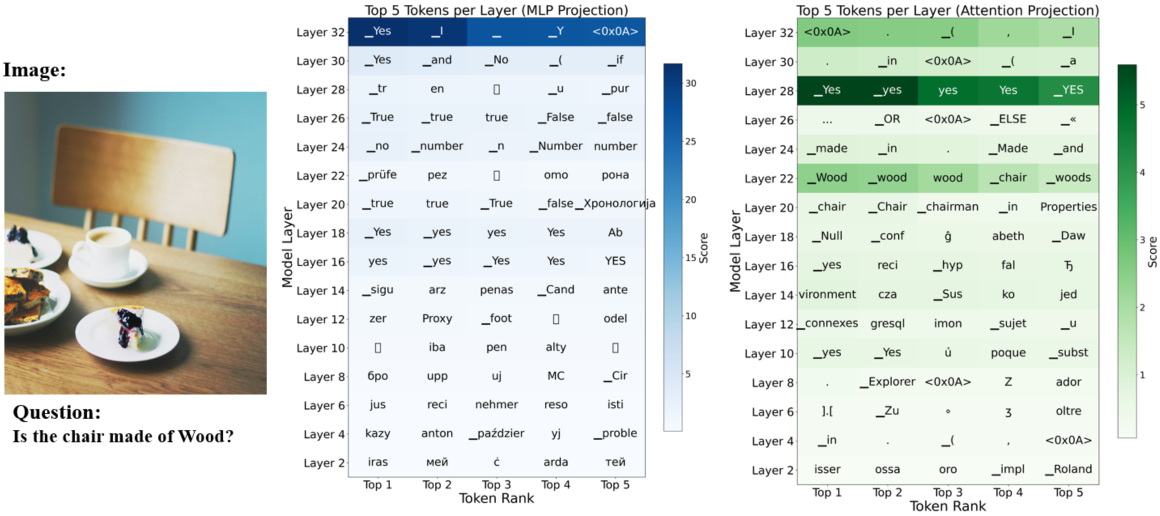
## E Activation Patching Supplementary Results

In this section, we supplement our analysis with activation patching experiments on four VQA sub-tasks—**Material**, **Vehicle**, **Animal**, and **Spatial**—to systematically examine the causal roles of the language model's **MLP** and **Attention** layers in the information flow. As shown in Figures 11, 12, 13, and 14. These experiments are conducted on two representative multimodal models, **LLaVA** and **InstructBLIP**, both of which exhibit highly consistent causal response patterns and functional specialization across different semantic tasks. These findings further support our main analysis regarding the separation of reasoning phases and the division of roles between MLP and Attention layers in multimodal reasoning.

## F Semantic Attention Head Supplementary Results

**Comprehensive Attention Head Attribution Results by Semantic Category.** To complement the main paper's analysis of shared attention heads across semantic levels, we provide the complete attribution results for each VQA task in Table 13. For every semantic subtype (e.g., color, material, spatial relation), we identify the Top-10 *reasoning heads* and Top-10 *distraction heads* based on their estimated causal contributions to the model's output, as measured by attention-head ablation and activation patching methods described in Section 4.

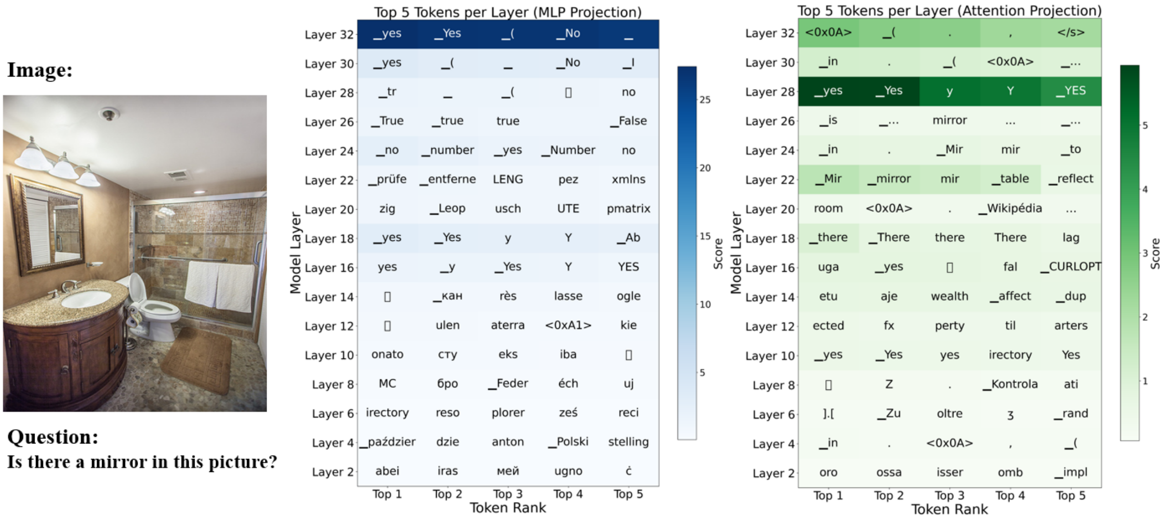
We list the most influential heads in both LLaVA and InstructBLIP, denoted as `L# H#`, where `L` refers to the transformer layer index and `H` to the head index within the layer.



| | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|
| Layer 32 | _Yes | _I | _ | _Y | <0x0A> |
| Layer 30 | _Yes | _and | _No | _( | _if |
| Layer 28 | _tr | en | □ | _u | _pur |
| Layer 26 | _True | _true | true | _False | _false |
| Layer 24 | _no | _number | _n | _Number | number |
| Layer 22 | _prüfe | pez | □ | omo | рона |
| Layer 20 | _true | true | _True | _false | _Хронологија |
| Layer 18 | _Yes | _yes | yes | Yes | Ab |
| Layer 16 | yes | _yes | _Yes | Yes | YES |
| Layer 14 | _sigu | arz | penas | _Cand | ante |
| Layer 12 | zer | Proxy | _foot | □ | odel |
| Layer 10 | □ | iba | pen | alty | □ |
| Layer 8 | 6po | upp | uj | MC | _Cir |
| Layer 6 | jus | reci | nehmer | reso | isti |
| Layer 4 | kazy | anton | _październ | yj | _proble |
| Layer 2 | iras | мей | ć | arda | тей |

| | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|
| Layer 32 | <0x0A> | . | _( | , | _I |
| Layer 30 | . | _in | <0x0A> | _( | _a |
| Layer 28 | _Yes | _yes | yes | Yes | _YES |
| Layer 26 | ... | _OR | <0x0A> | _ELSE | _« |
| Layer 24 | _made | _in | . | _Made | _and |
| Layer 22 | _Wood | _wood | wood | _chair | _woods |
| Layer 20 | _chair | _Chair | _chairman | _in | Properties |
| Layer 18 | _Null | _conf | ğ | abeth | _Daw |
| Layer 16 | _yes | reci | _hyp | fal | Ђ |
| Layer 14 | vironment | cza | _Sus | ko | jed |
| Layer 12 | _connexes | gresql | imon | _sujet | _u |
| Layer 10 | _yes | _Yes | ù | poque | _subst |
| Layer 8 | . | _Explorer | <0x0A> | Z | ador |
| Layer 6 | ].[ | _Zu | ∘ | ȝ | oltre |
| Layer 4 | _in | . | _( | , | <0x0A> |
| Layer 2 | isser | ossa | oro | _impl | _Roland |

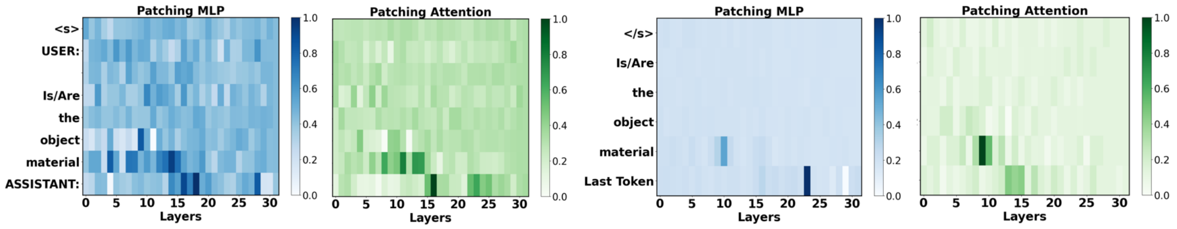
Figure 9: Case of MLP (blue) and self-attention (green) Logit Lens in LLAVA on Material VQA.



| | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|
| Layer 32 | _yes | _Yes | _( | _No | _ |
| Layer 30 | _yes | _( | _ | _No | _I |
| Layer 28 | _tr | _ | _( | □ | no |
| Layer 26 | _True | _true | true | | _False |
| Layer 24 | _no | _number | _yes | _Number | no |
| Layer 22 | _prüfe | _entferne | LENG | pez | xmlns |
| Layer 20 | zig | _Leop | usch | UTE | pmatrix |
| Layer 18 | _yes | _Yes | y | Y | _Ab |
| Layer 16 | yes | _y | _Yes | Y | YES |
| Layer 14 | □ | _кан | rès | lasse | ogle |
| Layer 12 | □ | ulen | aterra | <0xA1> | kie |
| Layer 10 | onato | сту | eks | iba | □ |
| Layer 8 | MC | 6po | _Feder | éch | uj |
| Layer 6 | irectory | reso | plorer | ześ | reci |
| Layer 4 | _październ | dzie | anton | _Polski | stelling |
| Layer 2 | abei | iras | мей | ugno | ć |

| | Top 1 | Top 2 | Top 3 | Top 4 | Top 5 |
|---|---|---|---|---|---|
| Layer 32 | <0x0A> | _( | . | , | </s> |
| Layer 30 | _in | . | _( | <0x0A> | _... |
| Layer 28 | _yes | _Yes | y | Y | _YES |
| Layer 26 | _is | _... | mirror | ... | _... |
| Layer 24 | _in | . | _Mir | mir | _to |
| Layer 22 | _Mir | _mirror | mir | _table | _reflect |
| Layer 20 | room | <0x0A> | . | _Wikipédia | ... |
| Layer 18 | _there | _There | there | There | lag |
| Layer 16 | uga | _yes | □ | fal | _CURLOPT |
| Layer 14 | etu | aje | wealth | _affect | _dup |
| Layer 12 | ected | fx | perty | til | arters |
| Layer 10 | _yes | _Yes | yes | irectory | Yes |
| Layer 8 | □ | Z | . | _Kontrola | ati |
| Layer 6 | ].[ | _Zu | oltre | ȝ | _rand |
| Layer 4 | _in | . | <0x0A> | , | _( |
| Layer 2 | oro | ossa | isser | omb | _impl |

Figure 10: Case of MLP (blue) and self-attention (green) Logit Lens in InstructBLIP on Indoor VQA.



**LLAVA** **InstructBLIP**

Figure 11: Layer-wise causal impact of MLP and self-attention on the Material VQA task

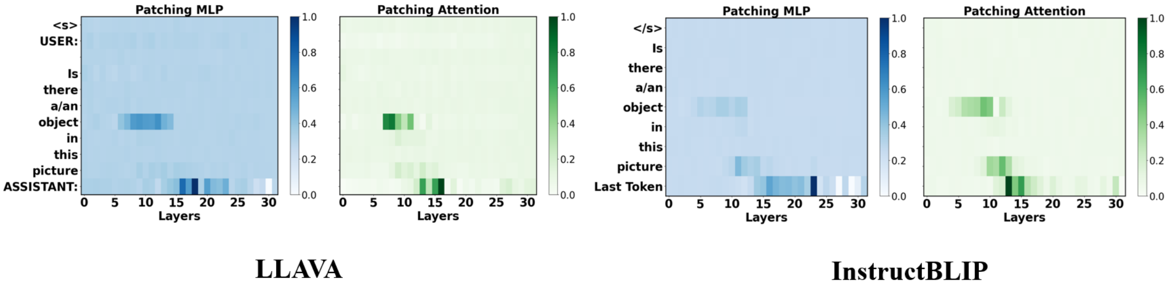


Figure 12: Layer-wise causal impact of attention and MLP on the Vehicle VQA task

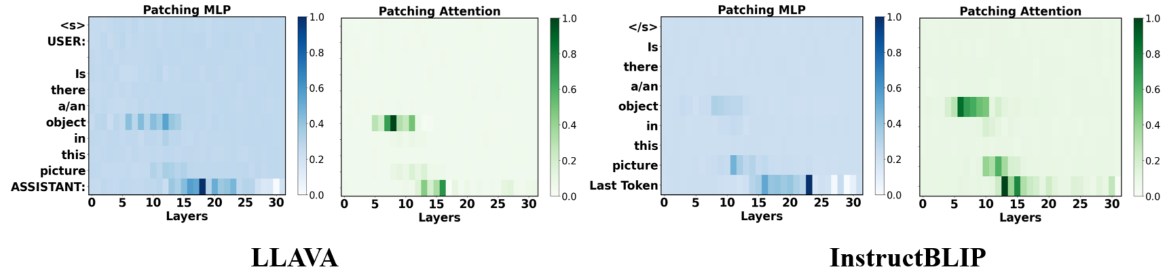


Figure 13: Layer-wise causal impact of MLP and self-attention on the Animal VQA task

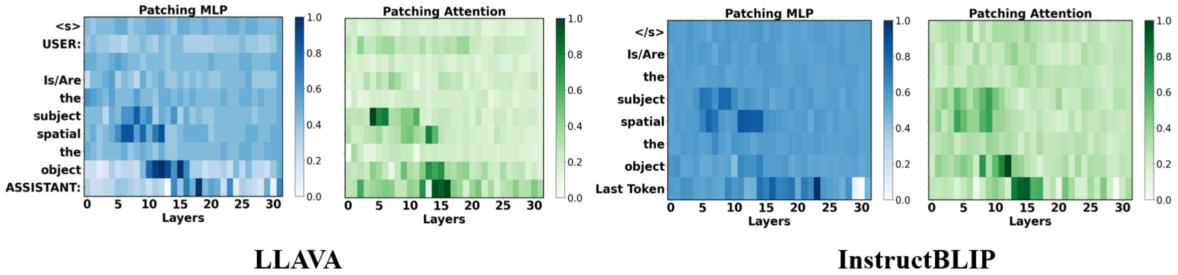


Figure 14: Layer-wise causal impact of MLP and self-attention on the Spatial VQA task

| VQA Task (Dataset) | LLAVA (Positive/Negative Heads) | InstructBLIP (Positive/Negative Heads) |
|---|---|---|
| Color Attribute Positive | L6 H19, L26 H15, L16 H4, L11 H29, L15 H7, L16 H1, L0 H11, L22 H11, L7 H23, L24 H11 | L11 H26, L10 H7, L6 H30, L9 H17, L10 H9, L6 H10, L10 H23, L6 H27, L11 H27, L11 H11 |
| Color Attribute Negative | L31 H11, L30 H11, L29 H11, L9 H12, L26 H11, L7 H27, L15 H26, L10 H18, L10 H1, L15 H3 | L0 H30, L10 H23, L6 H7, L29 H30, L11 H26, L6 H10, L10 H7, L7 H2, L30 H30, L9 H18 |
| Material Attribute Positive | L12 H23, L16 H1, L3 H14, L13 H8, L6 H17, L9 H22, L14 H18, L22 H11, L8 H15, L15 H7 | L23 H30, L7 H5, L6 H30, L11 H5, L8 H21, L7 H19, L11 H27, L5 H28, L11 H20, L12 H6 |
| Material Attribute Negative | L30 H11, L31 H11, L29 H11, L11 H29, L3 H11, L2 H11, L4 H11, L9 H24, L10 H16, L7 H12 | L0 H30, L10 H23, L6 H10, L7 H2, L30 H30, L10 H12, L13 H19, L11 H27, L6 H7, L15 H5 |
| Indoor Object Positive | L0 H11, L0 H23, L26 H11, L7 H20, L2 H16, L1 H17, L14 H18, L6 H26, L22 H11, L11 H7 | L23 H30, L11 H23, L7 H20, L11 H28, L23 H30, L9 H0, L12 H31, L6 H29, L11 H6, L7 H19 |
| Indoor Object Negative | L31 H11, L30 H11, L29 H11, L12 H2, L7 H2, L2 H11, L10 H1, L9 H27, L12 H29, L12 H13 | L0 H30, L8 H27, L6 H10, L30 H30, L12 H25, L10 H9, L13 H19, L14 H25, L30 H4, L10 H8 |
| Animal Object Positive | L0 H11, L26 H11, L1 H2, L11 H3, L11 H7, L2 H2, L6 H8, L15 H11, L10 H26, L11 H1 | L23 H30, L5 H28, L23 H30, L9 H0, L11 H28, L15 H22, L7 H20, L7 H13, L7 H11, L12 H31 |
| Animal Object Negative | L31 H11, L30 H11, L12 H2, L29 H11, L2 H31, L7 H2, L9 H27, L7 H9, L5 H0, L7 H1 | L0 H30, L6 H10, L10 H9, L0 H28, L12 H25, L13 H28, L10 H12, L30 H30, L6 H7, L5 H16 |
| Vehicle Object Positive | L0 H11, L26 H11, L8 H11, L2 H2, L1 H2, L11 H7, L10 H15, L11 H17, L13 H4, L11 H1 | L23 H30, L7 H20, L11 H28, L12 H22, L11 H5, L12 H31, L10 H10, L8 H13, L9 H28, L31 H19 |
| Vehicle Object Negative | L31 H11, L30 H11, L29 H11, L12 H2, L9 H30, L11 H30, L9 H27, L12 H29, L7 H2, L12 H13 | L0 H30, L6 H10, L10 H9, L30 H30, L12 H25, L7 H4, L6 H7, L6 H27, L8 H27, L10 H12 |
| Action Relation Positive | L16 H1, L26 H11, L1 H16, L13 H8, L8 H13, L8 H10, L12 H16, L14 H15, L5 H16, L13 H24 | L23 H16, L3 H30, L8 H28, L15 H22, L10 H11, L11 H5, L9 H28, L7 H19, L10 H6, L31 H19 |
| Action Relation Negative | L31 H11, L30 H11, L29 H11, L12 H14, L10 H1, L1 H4, L12 H2, L5 H19, L0 H11, L2 H11 | L6 H27, L12 H25, L10 H9, L7 H3, L6 H10, L10 H0, L10 H12, L30 H30, L30 H4, L7 H27 |
| Spatial Relation Positive | L10 H17, L14 H15, L15 H18, L18 H17, L10 H14, L8 H27, L9 H28, L2 H23, L13 H8, L8 H20 | L23 H16, L15 H22, L31 H19, L11 H5, L11 H28, L10 H1, L7 H19, L9 H28, L5 H30, L8 H28 |
| Spatial Relation Negative | L31 H11, L30 H11, L29 H11, L0 H11, L15 H23, L15 H10, L15 H4, L3 H11, L6 H14, L1 H16 | L10 H10, L12 H25, L6 H27, L13 H19, L0 H30, L12 H27, L5 H23, L30 H30, L10 H12 |

Table 13: Identified Top 10 positive and negative attention heads for different semantic VQA tasks in LLAVA and InstructBLIP.

| Dataset | Head Group | LLaVA | InstructBLIP |
|---|---|---|---|
| Animal | Positive | 83.8 | 79.2 |
| | Negative | 7.1 | 8.5 |
| Action | Positive | 87.0 | 85.4 |
| | Negative | 15.6 | 13.9 |

Table 14: Average Overlap (%) of Attention Inside Object Bounding Boxes. Higher values indicate stronger alignment with target objects. Results are computed over the top-10 positive/negative heads per model and averaged across samples.

Notably, several trends are consistent across models and semantic categories:

- In LLAVA, Positive heads tend to cluster in the mid-to-late layers (e.g., Layers 10–23), aligning with the model's transition from perceptual alignment to abstract reasoning.
- In LLAVA, Negative heads frequently concentrate near the final layers (e.g., Layers 29–31), suggesting late-stage attention drift or feature overfitting.
- Certain heads (e.g., `L11 H28`, `L23 H30`) appear recurrently across multiple semantic tasks in InstructBLIP, potentially functioning as universal positive heads.

These results further substantiate our claim that LVLMs develop semantically differentiated reasoning pathways, and offer concrete targets for causal editing interventions such as Attention rescaling.

**Behavioral Divergence of Positive and Negative Heads.** To better quantify the behavioral differences between positive and negative heads, we compute the average spatial alignment of their attention distributions with ground-truth object bounding boxes. Specifically, for each head, we calculate the attention overlap ratio—defined as the sum of attention weights inside the object bounding box divided by the total attention mass. Results are averaged across the top-10 positive and negative heads per model and per dataset. As shown in Table 14, positive heads show strong alignment (e.g., 83.8% in LLaVA-Animal), while negative heads often misalign (as low as 7.1%), typically attending to irrelevant background regions such as grass or walls. These findings quantitatively support the causal distinction between the two groups.

## G Dataset and Task Details for Causal Attention Head Evaluation

### G.1 POPE: A Benchmark for Object Hallucination Evaluation

The POPE (Polling-based Object Probing Evaluation) dataset (Li et al., 2023b) is a diagnostic benchmark specifically designed to evaluate large

| Subset | Number of Samples | Example Question |
|---|---|---|
| Popular | 3,000 | Is there a chair in the image? |
| Adversarial | 3,000 | Is there a man in the image? |
| Random | 2,910 | Is there a bottle in the image? |

Table 15: Overview of POPE subsets. Each subset contains binary object presence questions designed for different evaluation purposes.

| Question Type | Number of QA Pairs | Percentage |
|---|---|---|
| Object | 69,753 | 59.3% |
| Number | 16,594 | 14.1% |
| Color | 16,201 | 13.8% |
| Location | 15,136 | 12.8% |
| **Total** | **117,684** | **100%** |

Table 16: Distribution of question types in COCO-QA. The dataset is dominated by object-related questions.

vision-language models (LVLMs) on object hallucination in recognition tasks. Built on images from the MS-COCO dataset, POPE focuses on probing model robustness and bias under distribution shifts via object-level binary questions.

The POPE dataset consists of three subsets, each targeting a specific evaluation goal:

- **Popular**: Contains frequently occurring objects in pretraining corpora. This subset is used to test the model's reliance on common object priors.
- **Adversarial**: Introduces misleading distractors or co-occurrence biases to evaluate model robustness in challenging scenarios.
- **Random**: Includes randomly sampled object-question pairs, serving as a control group for unbiased evaluation.

Each image is paired with multiple binary questions, typically in the format: "Is there a `<object>` in the image?", where `<object>` is a specific noun.

In our experiments, we directly transfer attention heads identified from the GQA dataset to POPE, and compare their performance with those re-identified from 10% and 20% of POPE's training samples.

The statistics and examples of each subset are shown in Table 15.

### G.2 COCO-QA: A Dataset for Open-Ended Visual Question Answering

The COCO-QA (COCO Question Answering) dataset (Lu et al., 2016) is an automatically generated visual question answering (VQA) dataset based on images from the MS-COCO Captions dataset. It is designed to evaluate image understanding and language grounding in an open-ended QA setting.

**Dataset Structure.** COCO-QA consists of:

- **Images:** Derived from the MS-COCO dataset, with approximately 123,287 unique images.
- **Question-Answer Pairs:** A total of 117,684 QA pairs, each linked to a COCO image. Questions are automatically generated from image captions using syntactic and semantic transformations. Answers are typically single words or short phrases.

**Question Types.** COCO-QA questions are categorized into four semantic types:

- **Object:** Identifying objects in the image (e.g., "What is on the table?").
- **Number:** Counting entities (e.g., "How many people are there?").
- **Color:** Recognizing object attributes (e.g., "What color is the car?").
- **Location:** Understanding spatial relationships (e.g., "Where is the cat?").

As shown in Table 16, object-related questions constitute the majority of the dataset, followed by number, color, and location types.

**Subset Selection.** In our experiments, we select the *Object*, *Color*, and *Location* subsets for evaluation. These three categories exhibit clear semantic meanings and align closely with our defined semantic reasoning types: object recognition, attribute identification, and spatial relation understanding. We exclude the *Number* subset due to its relatively ambiguous semantics and lack of direct correspondence to our reasoning framework.

## H Additional Results on Semantic Attention Editing

To complement our main results on LLaVA, we report additional results of Attention head rescaling applied to the InstructBLIP model across

| Model | Ablation | Attribute | | Object | | | Relation | |
|---|---|---|---|---|---|---|---|---|
| | | Color | Material | Animal | Vehicle | Indoor | Action | Spatial |
| LLaVA | original | 76.13 | 78.84 | 96.45 | 90.29 | 83.80 | 88.21 | 77.85 |
| | w/o negative | 79.73 | 83.77 | 97.01 | 92.70 | 88.58 | 92.89 | 85.33 |
| | w/o positive | 74.40 | 75.62 | 95.33 | 89.20 | 79.64 | 86.44 | 73.18 |
| | random remove | 76.00 | 78.92 | 96.36 | 90.11 | 83.13 | 88.08 | 77.79 |
| | rescaling | **80.27** | **85.31** | **97.10** | **92.87** | **89.77** | **93.29** | **88.92** |
| InstructBLIP | original | 77.53 | 81.38 | 96.17 | 90.06 | 87.14 | 90.88 | 87.64 |
| | w/o negative | 80.47 | 85.08 | 97.38 | 93.10 | 92.78 | 95.19 | 95.59 |
| | w/o positive | 75.47 | 80.23 | 94.86 | 87.30 | 80.45 | 81.17 | 71.90 |
| | random remove | 77.53 | 81.15 | 96.07 | 90.00 | 87.14 | 91.86 | 86.51 |
| | rescaling | **81.27** | **86.84** | **97.48** | **93.85** | **92.83** | **96.03** | **96.77** |

Table 17: Accuracy (%) of LLaVA and InstructBLIP on VQA tasks categorized by task type: Attribute (Color, Material), Object (Animal, Vehicle, Indoor), and Relation (Action, Spatial). The table reports performance under different attention head editing strategies.

both GQA-style semantic sub-tasks and the POPE dataset. These results demonstrate that head rescaling is also effective when applied to more instruction-tuned multimodal models, such as InstructBLIP.

**Effectiveness of Attention Head Editing** Table 17 presents the results of systematically editing the identified key attention heads. We observe that for both LLaVA and InstructBLIP, attention head editing brings significant performance improvements across all semantic levels. These results confirming that our method effectively enhances the semantic understanding capabilities of vision-language models.

**Statistical Significance Analysis** To assess whether the performance gains are statistically significant, we applied bootstrap resampling (Koehn, 2004) by generating 1,000 folds, each consisting of 100 test cases randomly sampled from the original test sets, with each set emphasizing a specific type of semantic change (i.e., object, relationship, or attribute). For each fold, we calculated the performance difference ($\Delta$(pp)) between our attention head rescaling method and each of the four baselines, defined as $\Delta$(pp) = rescaling - baseline.

For each semantic property, we obtained 1,000 performance difference values per method comparison and conducted paired t-tests (Student, 1908) to assess the statistical significance between each pair of methods. Table 18 shows the results based on the LLaVA model. Note that the "Overall" results are calculated based on all test folds across the three semantic properties for each method comparison.

As shown in the Table 18, our rescaling strategy achieves statistically significant improvements over all four baselines across every category of the LLaVA model. Paired t-tests further show the robustness of these results, with all p-values being extremely small ($p < 0.001$), indicating a high level of statistical significance.

**Data Dependence Analysis** To assess the data dependence of our method, we randomly sample 10%–50% of the experimental data for key head identification and editing, then test on the full set. Tables 19, 20, and 21 show the performance of InstructBLIP and LLaVA under different data proportions.

For InstructBLIP, even with only 10% of the training data, our method achieves performance close to full-data supervision across all task categories. Table 20 further provides detailed performance of InstructBLIP on seven specific semantic tasks, demonstrating the data efficiency and general applicability of our method.

For LLaVA, Table 21 shows that even in low-resource scenarios (using only 10% of the data), our method can achieve comparable performance, highlighting the advantages of attention head rescaling in practical applications.

**Generalizability Evaluation** We evaluate the generalizability of our editing method on two out-of-distribution (OOD) benchmarks: POPE and COCOQA.

**Performance on POPE Dataset** Table 22 demonstrates the performance of InstructBLIP on

| Category | Baseline | Mean $\Delta$(pp)(%) | SD(%) | t | p-value |
|---|---|---|---|---|---|
| Attribute | original | 7.92 | 2.96 | 84.79 | *** |
| | random remove | 7.80 | 3.52 | 70.20 | *** |
| | w/o positive | 10.55 | 3.95 | 84.61 | *** |
| | w/o negative | 2.93 | 1.87 | 49.68 | *** |
| Object | original | 3.01 | 1.51 | 63.16 | *** |
| | random remove | 3.19 | 2.06 | 49.06 | *** |
| | w/o positive | 4.78 | 2.27 | 66.75 | *** |
| | w/o negative | 1.93 | 1.18 | 51.81 | *** |
| Relation | original | 9.57 | 3.82 | 79.40 | *** |
| | random remove | 10.43 | 3.93 | 84.08 | *** |
| | w/o positive | 13.62 | 4.39 | 98.29 | *** |
| | w/o negative | 3.65 | 2.14 | 54.00 | *** |
| Overall | original | 6.62 | 3.61 | 100.49 | *** |
| | random remove | 6.75 | 3.73 | 99.17 | *** |
| | w/o positive | 9.86 | 4.46 | 121.18 | *** |
| | w/o negative | 2.88 | 1.89 | 83.49 | *** |

Table 18: Performance difference $\Delta$(pp) between the attention head rescaling method and baselines. Statistical significance is indicated with asterisks. *** indicates $p < 0.001$ (extremely significant).

| Category | Initial | 10% | 20% | 30% | 50% | Full |
|---|---|---|---|---|---|---|
| Attribute | 79.46 | 80.48 | 82.11 | 83.05 | 83.73 | **84.06** |
| Object | 91.12 | 92.78 | 93.57 | 94.37 | 94.61 | **94.72** |
| Relation | 89.26 | 91.42 | 93.56 | 95.27 | 96.10 | **96.40** |
| Average | 86.61 | 88.23 | 89.75 | 90.90 | 91.48 | **91.73** |

Table 19: Accuracy (%) of **InstructBLIP** under different sample proportions for rescaling, grouped by task categories.

| Dataset | Initial | 10% | 20% | 30% | 50% | Full |
|---|---|---|---|---|---|---|
| Color | 77.53 | 78.80 | 79.60 | 80.33 | 81.07 | **81.27** |
| Material | 81.38 | 82.15 | 84.62 | 85.77 | 86.38 | **86.84** |
| Indoor | 87.14 | 89.48 | 90.63 | 92.02 | 92.50 | **92.83** |
| Animal | 96.17 | 96.72 | 97.20 | **97.48** | **97.48** | **97.48** |
| Vehicle | 90.06 | 92.13 | 92.87 | 93.62 | **93.85** | **93.85** |
| Action | 90.88 | 92.79 | 93.52 | 94.89 | 95.86 | **96.03** |
| Spatial | 87.64 | 90.05 | 93.59 | 95.64 | 96.34 | **96.77** |

Table 20: Accuracy (%) of **InstructBLIP** under different sample proportions for attention head rescaling across the seven semantic tasks defined in Section 5.1.

| Dataset | Initial | 10% | 20% | 30% | 50% | Full |
|---|---|---|---|---|---|---|
| Color | 76.13 | 77.73 | 79.80 | 79.93 | 80.00 | **80.27** |
| Material | 78.84 | 81.23 | 82.54 | 84.46 | 85.00 | **85.31** |
| Indoor | 83.80 | 87.48 | 89.67 | 89.77 | **89.82** | 89.77 |
| Animal | 96.45 | 97.01 | **97.10** | **97.10** | **97.10** | **97.10** |
| Vehicle | 90.29 | 92.76 | 92.82 | 92.84 | **92.87** | **92.87** |
| Action | 88.21 | 90.58 | **93.29** | 93.22 | 93.26 | **93.29** |
| Spatial | 77.85 | 83.44 | 88.62 | 88.82 | 88.87 | **88.92** |

Table 21: Accuracy (%) of **LLaVA** under different sample proportions for rescaling across various VQA tasks.

the POPE dataset. We compare three strategies: (1) no attention editing ("Initial"), (2) head rescaling using POPE data ("Edit_POPE"), and (3) using attention heads and importance scores identified from GQA ("Edit_GQA"). The results show that even without re-analysis on the target dataset, attention head rescaling significantly improves VLM's performance, demonstrating good cross-task generalization capability of our method.

**Performance on COCOQA Dataset** Table 23 shows the direct comparison between GQA-edited VLMs and unedited baselines on the COCOQA dataset, which contains open-ended questions covering color, object, and location categories. The results show consistent performance improvements across all tasks for the edited models, further prov-

| Task | Metric | Initial | Edit_POPE | Edit_GQA |
|---|---|---|---|---|
| Popular | Acc | 0.842 | **0.861** | 0.857 |
| | Recall | 0.731 | 0.782 | **0.795** |
| | F1 | 0.823 | **0.843** | 0.841 |
| Adversarial | Acc | 0.830 | 0.846 | **0.848** |
| | Recall | 0.731 | **0.787** | 0.783 |
| | F1 | 0.811 | 0.828 | **0.829** |
| Random | Acc | 0.852 | **0.872** | 0.869 |
| | Recall | 0.731 | 0.786 | **0.790** |
| | F1 | 0.835 | **0.853** | 0.854 |

Table 22: Performance of **InstructBLIP** on POPE under different attention head rescaling strategies. "Initial" refers to the performance without any intervention. "Edit_POPE" refers to applying head rescaling based on POPE. "Edit_GQA" denotes using heads and scores discovered on GQA.

| Model | Initial | Edit_GQA |
|---|---|---|
| *Color Task* | | |
| InstructBLIP | 65.54 | **68.49** |
| LLaVA | 80.48 | **82.35** |
| *Object Task* | | |
| InstructBLIP | 78.40 | **80.12** |
| LLaVA | 85.15 | **86.40** |
| *Location Task* | | |
| InstructBLIP | 67.39 | **69.01** |
| LLaVA | 61.84 | **62.98** |

Table 23: Performance of InstructBLIP and LLaVA models on the *Color*, *Object*, and *Location* sub-tasks of COCOQA before and after applying attention head rescaling. Accuracy is reported in percentage.

ing the generalizability of our method.

In summary, these appendix results comprehensively demonstrate the effectiveness and generalizability of the Attention head rescaling method across different models, data quantities, and tasks, providing strong support for the arguments presented in the main body of this paper.

## I Computational Resources

All experiments were conducted on a single machine equipped with two NVIDIA Tesla V100 GPUs (32GB memory), an Intel(R) Xeon(R) Gold 6230R CPU running at 2.10GHz (8 cores), and 62GB of RAM.

Our interpretability analyses, including activation patching, attention head attribution, and visual perturbation generation, were implemented using PyTorch and Hugging Face Transformers.

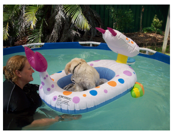
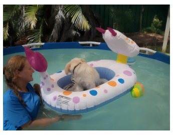
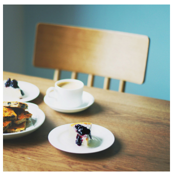
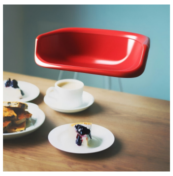
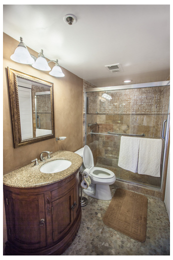
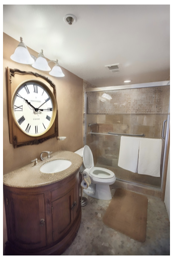
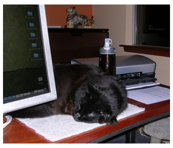
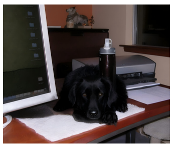
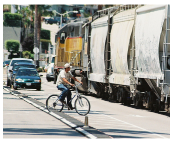
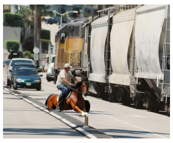
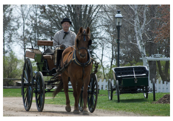
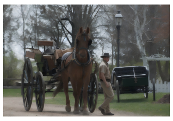
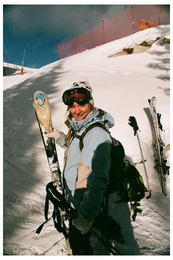
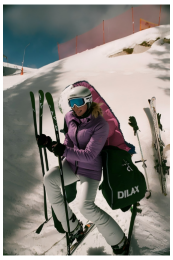
| Question | Original Image | Original Prediction | Perturbed Image | Perturbed Prediction |
|---|---|---|---|---|
| Is the shirt blue? |  | No |  | Yes |
| Is the chair made of wood? |  | Yes |  | No |
| Is there a mirror in this picture? |  | Yes |  | No |
| Is there a dog in this picture? |  | No |  | Yes |
| Is there a bike in this picture? |  | Yes |  | No |
| Is the man right of the horse? |  | No |  | Yes |
| Is the woman holding the skis? |  | Yes |  | No |

Table 24: Qualitative examples of semantic perturbation on seven VQA tasks using LLaVA. Each row presents the VQA question, original image and model prediction, and the perturbed image with the updated prediction.