Attribution-Guided Multi-Object Hallucination and Bias Detection in Vision-Language Models

Sirat Samyoun, Yingtai Xiao, Jian Du
TikTok Inc.
{sirat.samyoun,yingtai.xiao,jian.du}@bytedance.com

Abstract

Vision-Language Models excel in multi-modal tasks but often hallucinate objects or exhibit linguistic bias by over-repeating object names, especially in complex multi-object scenes. Existing methods struggle with multi-object grounding because language priors frequently dominate visual evidence, causing hallucinated or biased objects to produce attention distributions or similarity scores nearly indistinguishable from those of real objects. We introduce **SHAPLENS**, a Shapley value–based attribution framework using Kernel SHAP and multi-layer fusion to detect hallucinated and biased objects. Evaluated on ADE and COCO datasets across four leading VLMs, SHAPLENS improves hallucination detection accuracy by **8–12%** and F1 by **10–14%** over the best baselines. It also achieves **up to 6%** higher bias detection performance across three distinct bias types on a curated HQH benchmark and exhibits minimal degradation (<0.03%) across partial and perturbed contexts.

1 Introduction

Vision-Language Models (VLMs) (Radford et al., 2021; Liu et al., 2023) have achieved remarkable success in multi-modal reasoning, visual question answering, and caption generation. However, despite strong performance, these models often fail to align object mentions with true visual evidence, particularly in complex *multi-object* scenes. Two critical failure modes are *hallucination*—predicting objects not visually present—and *linguistic bias*—over-repetition of object names driven by language priors (Leng et al., 2024). Recent evaluations show that open-vocabulary captioning models hallucinate in 30 – 40% of complex multi-object scenes (Ben-Kish et al., 2023; Chen et al., 2024a). Such failures can be harmful—for example, misleading accessibility outputs (Rohrbach et al., 2018), navigation overconfidence (Pan et al., 2023), erroneous object recognition in robotics (Brohan et al., 2023), or diagnostic errors in medical imaging (Thawkar et al., 2023).

Existing hallucination detection methods fall short in multi-object scenarios due to two core challenges. *First*, embedding-similarity methods—e.g., CLIP-based scoring (Deng and Li, 2024) or contextual embeddings (Phukan et al., 2024)—often confuse semantically adjacent labels. For example, a "suitcase" scene may spuriously activate "tie" or "briefcase" due to high cosine similarity, as shown in CHAIR and DASH evaluations (Ben-Kish et al., 2023; Neu and Ji, 2025). *Second*, attention-based methods—e.g., OPERA (Huang et al., 2024) and EAZY (Che et al., 2025)—aggregate signals across heads, diluting strong patch contributions; as shown in Section 2.1, hallucinations can then receive weights indistinguishable from real objects. Furthermore, fine-tuning methods (e.g., (Jiang et al., 2024; Zhao et al., 2023)) incur high training costs, while inference methods (e.g., (Lee et al., 2023; Yin et al., 2024)) lack patch-level precision. Critically, no prior work differentiates between *hallucination* and *linguistic bias*, treating them as one phenomenon—even though our evidence shows they demand separate handling. These shortcomings highlight the need for fine-grained, training-free attribution in multi-object settings.

To address these challenges, we propose **SHAPLENS**, a novel attribution framework for hallucination and bias detection in multi-object scenes. We combine the principles of *Kernel-SHAP* (Lundberg and Lee, 2017) and *LogitLens* (Nostalgebraist, 2020; Che et al., 2025) to perform layer-wise, patch-level attribution within a training-free framework, robustly detecting hallucinated and biased outputs. Our contributions are:

- **Framework for Hallucination and Bias Detection:** Unified framework SHAPLENS for multi-object hallucination and bias (*label repetition, attribute, lexical root*) detection.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4529–4548

- **Training-Free, Architecture-Agnostic Attribution:** Leverages LogitLens-guided layer selection and Kernel SHAP-based patch attribution to enable fine-grained detection across diverse VLM architectures.
- **Benchmark Improvements:** SHAPLENS achieves **8–12%** accuracy gain and **10–14%** F1-score gain in hallucination detection, and up to **6%** F1-score gain in bias detection, showing strong performance in multi-object scenes.
- **Robustness to Visual Perturbations:** Maintains near-perfect robustness under occlusion, noise, and bounding box shifts: $\Delta_{\text{diff}} \leq 0.0003$ with no change in classification outcomes.

2 Problem Formulation & Background

We present two distinct failure modes, *hallucination* and *linguistic bias*, in Figure 1, with additional examples in Appendix H (Figures 6–10). In both cases, we consider an image $I \in \mathbb{R}^{H\times W\times 3}$ having N objects and its predictions $\mathcal{P} = \{p_i\}_{i=1}^{N}$ produced by a VLM.

2.0.1 Hallucination Detection

Given an image, the goal is to determine whether each predicted object truly exists in the visual scene. Let the set of bounding boxes be $\mathcal{B} = \{b_i = (x_i, y_i, w_i, h_i)\}_{i=1}^{N}$, where (x_i, y_i) denotes the top-left corner and w_i, h_i are the width and height. Each box is partitioned into K uniform patches $\mathcal{Z}_i = \{z_j\}_{j=1}^{K}$ (e.g., an 8×8 grid yields 64 patches). For every prediction p_i, we measure its attribution across patches and mark it as hallucinated if no patch in $\mathcal{Z}_i$ provides sufficient visual support (below an adaptively learned threshold).

2.0.2 Bias Detection

We flag a prediction p_i as biased if it recurs in the prediction set $\mathcal{P}$ without significant visual support (below an adaptively learned bias threshold). Unlike hallucination detection, bias detection operates directly on token sequences and requires no bounding box information. Specifically, we define three categories of linguistic bias, with explanatory figures in Appendix H:

- **Label Repetition Bias**: Occurs when the model repeats the exact same object label multiple times far beyond its true count in the image. For example, in scenes containing 4 goats (Fig. 1), the model may generate outputs like *goat* * 17 times.
- **Attribute Bias**: Occurs when the model overgenerates descriptive attribute tokens (e.g., *white, green, red, round*) rather than object labels themselves (Fig. 7). This reflects a challenge in fine-grained visual grounding, where the model favors surface-level attributes over correct object identification.
- **Lexical-Root Propagation Bias**: Arises when the model excessively repeats a common subtoken across different compound labels (Fig. 9). For example, it may repeatedly use the subword "airport" in variations such as *airport vehicle, airport ground crew, airport equipment, airport operations*.

2.1 Limitations of Existing Works

Case Study 1: Object Hallucination: Standard attention-based techniques (Huang et al., 2024; Che et al., 2025) struggle in dense urban scenes. In Fig. 1(a), mean attention for the hallucinated "potted plant" (0.018) is indistinguishable from the valid "green water bottle" (0.021). Similarity-based scores (Deng and Li, 2024; Phukan et al., 2024) also show limited separation: the hallucinated "bench" (similarity 0.021) closely matches actual objects like "bus stand" (0.022) and "fence" (0.019), making filtering difficult.

Case Study 2: Linguistic Bias: Fig. 1(b) shows excessive "goat" repetition ($\times$12). CLIP (Deng and Li, 2024) reports maximal similarity even on random noise (1.00), indicating strong text-prior dependence. Mean attention remains low (0.0200) with no strong activation in specific heads, suggesting weak visual localization.

These cases reveal fundamental limitations: similarity metrics conflate semantically related labels, while attention aggregation weakens discriminative signals.

2.2 Primer on KernelShapley Attribution

KernelSHAP (Lundberg and Lee, 2017) approximates Shapley values (Shapley, 1953) by sampling random subsets of input features and computing their marginal effects. For a predicted object p_i with bounding box B_i, let $\mathcal{Z}_i = \{z_j\}_{j=1}^{K_i}$ denote the set of patches within the box. The Shapley value ϕ_j for patch $z_j \in \mathcal{Z}_i$ is estimated using M random permutations:

$$\phi_j = \frac{1}{M}\sum_{m=1}^{M} \delta_j^{(m)} \cdot w_{S_m} \tag{1}$$

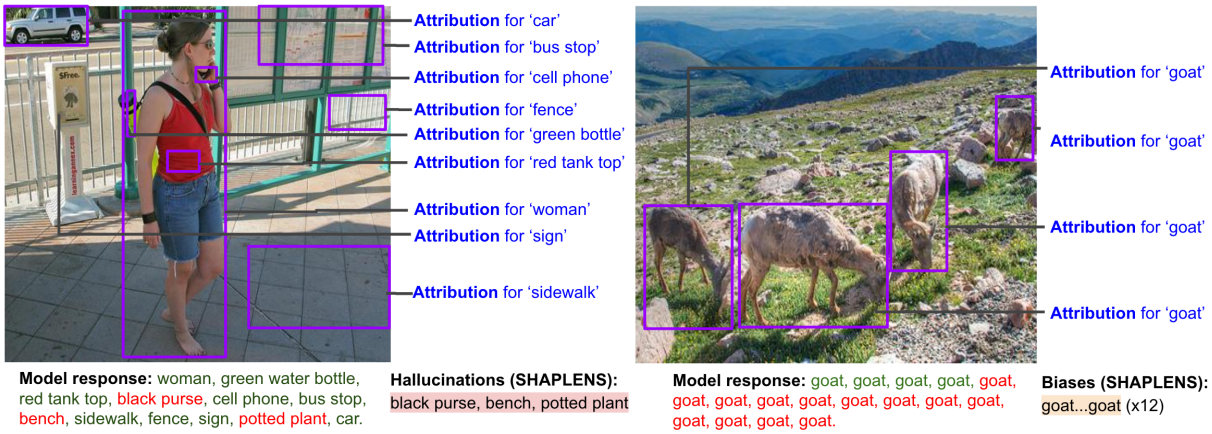


Figure 1: Examples of two failure modes of VLMs in multi-object contexts: (a) Object hallucination, where non-existent objects such as "black purse", "bench" and "potted plant" are incorrectly included in the scene; (b) Linguistic bias, characterized by excessive repetition of "goat" ($\times$12). SHAPLENS attributes predicted object labels to image patches across multiple layers, grounding each prediction in visual evidence.

Each marginal contribution is computed as:

$$\delta_j^{(m)} = v(S_m \cup \{z_j\}) - v(S_m), \qquad (2)$$

where $S_m \subset \mathcal{Z}_i \setminus \{z_j\}$ is a randomly sampled subset excluding z_j, and $v(S)$ denotes the model output when only patches in S are visible. The Shapley kernel w_{S_m} assigns higher weight to intermediate-sized subsets to enable balanced marginal sampling. Unlike the original KernelSHAP, we omit the weighted regression step and directly average marginal effects, as the output is scalar-valued—making regression unnecessary for patch-level attribution.

2.3 Primer on LogitLens

LogitLens is a technique used to interpret intermediate representations in transformers by projecting hidden states at each layer back into vocabulary logits (Nostalgebraist, 2020). Originally proposed for text-only LMs, recent works (Neo et al., 2024; Che et al., 2025) demonstrate its surprising effectiveness in VLMs, showing that visual token representations evolve toward interpretable vocabulary-aligned concepts across layers.

3 Methodology

3.1 SHAPLENS Framework Components

To address the aforementioned challenges, we present SHAPLENS, which integrates KernelSHAP and LogitLens principles to produce grounded visual evidence across multi-object scenes (Fig. 1). The key steps of our methodology are: (i) Layer Selection (ii) Hallucination Detection Algorithm (iii) Bias Detection Algorithm.

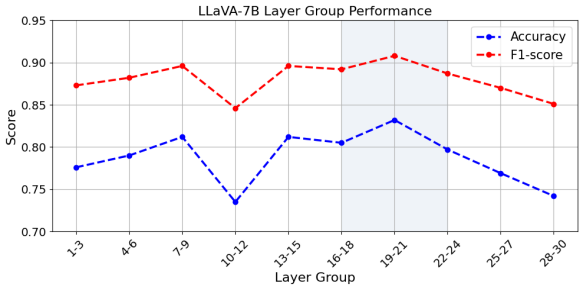


Figure 2: Accuracy and F1-score across 3-layer groups, with peak performance in mid-late layers.

3.2 Layer Selection via Attribution Analysis

A critical challenge in interpreting VLMs is identifying optimal layers for attribution, as this varies across architectures. We conduct a pilot study to establish a generalizable, data-driven selection method for our SHAPLENS framework.

We compute attribution-based accuracy for each 3-layer group on diverse images ($N = 100$) from ADE (Zhou et al., 2017). This dataset provides representative visual concepts for vision-language tasks, and we verify that the identified layer patterns generalize to COCO dataset (Lin et al., 2014):

$$\text{Acc}^{(q)} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\psi_i^{(q)} > \tau_i^{(q)}\right)$$

where $\psi_i^{(q)} = \frac{1}{3} \sum_{\ell \in \text{group } q} \left(\frac{1}{K_i} \sum_{j=1}^{K_i} \phi_{i,j}^{(\ell)}\right)$ is the average attribution strength for object i across 3-layer group q, $\phi_{i,j}^{(\ell)}$ is the Shapley value for patch j at layer ℓ, K_i is the number of patches within bounding box b_i, and $\tau_i^{(q)}$ is a group-specific adap-

Algorithm 1 SHAPLENS Hallucination Detection

Require: Image I, VLM outputs: preds $\mathcal{P}$, self-generated boxes $\mathcal{B}$, layers $\mathcal{L}$, perms M
Ensure: Flags $\{h_i\}$
for each label $p_i \in \mathcal{P}$ with box $b_i \in \mathcal{B}$ **do**
  $\mathbf{c}_i \leftarrow$ Compute semantic features for label p_i
  for $l \in \mathcal{L}$ **do**
    $\{\mathbf{h}_{i,j}^{(l)}\} \leftarrow$ Extract layer-l features for each patch $j \in B_i$
    Cosine similarity $s_{i,j}^{(l)} = \frac{\langle \mathbf{h}_{i,j}^{(l)}, \mathbf{c}_i \rangle}{\|\mathbf{h}_{i,j}^{(l)}\| \|\mathbf{c}_i\|}$
    $\phi_{i,j}^{(l)} \leftarrow$ Approximate Shapley value for patch j via M permutations
  end for
  Aggregate: $\Phi_i = \frac{1}{|\mathcal{L}|\,|\{\mathbf{h}_{i,j}^{(l)}\}|} \sum_{l \in \mathcal{L}} \sum_{j} \phi_{i,j}^{(l)}$
  $\tau_i \leftarrow$ Adaptive threshold on $\{\phi_{i,j}^{(l)}\}$
  Flag: Mark p_i as hallucinated if $\Phi_i < \tau_i$
end for

tive threshold determined by Gaussian mixture modeling.

Our analysis indicates that the performance of LLaVA-7B peaks around layers 19–21 (Fig. 2). Similar trends are observed for Qwen2-VL-2B, LLaVA-13B, and LLaMA-3.2-Vision-11B, as detailed in Appendix Fig. C. Early layers encode low-level cues, late layers overspecialize for generation, and mid layers achieve the best abstraction–interpretability balance (Neo et al., 2024; Lin et al., 2025). Our layer selection algorithm, explained in detail in the Appendix, identifies these layers by maximizing attribution accuracy.

3.3 Hallucination Detection Algorithm

Our algorithm (Algorithm 1) identifies hallucinated object predictions by quantifying the visual support each label receives from localized image regions via Shapley-based attribution. For each predicted label p_i with bounding box b_i, we first compute a semantic embedding $\mathbf{c}_i$ by encoding the tokenized label through the model's text encoder and averaging over subtokens, yielding a robust representation in the shared vision-language space.

Next, we extract visual features $\mathbf{h}_{i,j}^{(l)}$ from each layer $l \in \mathcal{L}$ for all spatial patches j overlapping the region b_i. The image is processed through the visual encoder, and we retain representations from patches spatially mapped to the box. Each patch feature $\mathbf{h}_{i,j}^{(l)}$ is normalized and projected onto the label embedding $\mathbf{c}_i$ using cosine similarity, producing an alignment score $s_{i,j}^{(l)} = \frac{\langle \mathbf{h}_{i,j}^{(l)}, \mathbf{c}_i \rangle}{\|\mathbf{h}_{i,j}^{(l)}\| \|\mathbf{c}_i\|}$.

To assess the contribution of each patch to the overall alignment, we estimate Shapley values

Algorithm 2 SHAPLENS Bias Detection

Require: Token list T, image I, blank image $I_\varnothing$, layers $\mathcal{L}$
Ensure: Bias flags for {label, attribute, root}
Identify repeated tokens $R \subseteq T$, attribute tokens $A \subseteq T$, root tokens $X \subseteq T$
Compute hidden states $\{\mathbf{h}_\ell^{\text{fused}}, \mathbf{h}_\ell^{\text{blank}}\}_{\ell \in \mathcal{L}}$
for each phrase $r \in R \cup A \cup X$ **do**
  Tokenize r into subtokens $\{r_s\}$
  For each r_s, compute confidence drop across layers: $\delta_{r,s,\ell} = p_{\text{fused}}(r_s, \ell) - p_{\text{lang}}(r_s, \ell)$
  $\Delta_r \leftarrow \min_{s,\ell} \delta_{r,s,\ell}$
  $\tau_r \leftarrow$ Adaptive threshold on$\{\delta_{r,s,\ell}\}$
  Flag: r is biased if $\Delta_r < \tau_r$
end for

$\phi_{i,j}^{(l)}$ using a KernelSHAP-style permutation approach (Lundberg and Lee, 2017). For each of M random permutations, we compute the marginal improvement in the running maximum score when patch j is added:

$$\phi_{i,j}^{(l)} \approx \frac{1}{M} \sum_{m=1}^{M} \left[\max_{k \in S_m \cup \{j\}} s_{i,k}^{(l)} - \max_{k \in S_m} s_{i,k}^{(l)} \right] \cdot w_{S_m}$$

where S_m is the preceding set in permutation m, w_S is the Shapley kernel, and $\phi_{i,j}^{(l)}$ serves as the final patch-level attribution score. This Monte Carlo estimator ensures linear scaling in patches and permutations, enabling tractable application to multi-object dense scenes.

To improve robustness to dataset- and object-specific score distributions, we employ adaptive thresholding over the fused attribution scores. Specifically, we fit a two-component Gaussian mixture model (GMM) over all $\phi_{i,j}^{(l)}$, and define the threshold τ_i as a weighted midpoint between the component means m_1 and m_2, i.e., $\tau_i = \lambda m_1 + (1-\lambda) m_2$, where $0 < \lambda < 1$ is the weight. The component means m_1 and m_2 are estimated directly from the fitted GMM over the flattened attribution distribution, representing low- and high-support patch clusters, respectively. This soft separation strategy better captures bimodal structure between grounded and weakly-supported patches, reducing sensitivity to outliers and class imbalance. An object label i is flagged as hallucinated if $\Phi_i < \tau_i$, where Φ_i is the aggregated attribution score over all layers and patches.

3.4 Bias Detection Algorithm

In VLMs, repeated tokens often arise from over-reliance on language priors rather than visual evidence (Leng et al., 2024). To quantify this, our method, presented in Algorithm 2 computes token-level confidence drops by contrasting predictions

under a real image I and a blank input $I_\varnothing$. Specifically, we define:

- **Fused Confidence** $p_{\text{fused}}(r_s, \ell)$: Softmax probability of subtoken r_s at layer ℓ from vision-language fused hidden states.
- **Language-Only Confidence** $p_{\text{lang}}(r_s, \ell)$: Corresponding score from a language-only forward pass with a blank image.

To evaluate token grounding, we aggregate the K image patch embeddings $\{\mathbf{h}_{\ell,j}\}_{j=1}^{K}$ at each transformer layer $\ell \in \mathcal{L}$ by their mean, and project this mean through the vocabulary head $W_U \in \mathbb{R}^{V \times d}$ to obtain a logit vector $\mathbf{o}_\ell \in \mathbb{R}^V$:

$$\mathbf{o}_\ell \;=\; W_U^\top \Big(\tfrac{1}{K} \sum_{j=1}^{K} \mathbf{h}_{\ell,j} \Big)$$

We compute $\mathbf{o}_\ell$ both for the original image and for a blank image, apply softmax to each to get probability vectors $p(r_s, \ell)$, and compare them to quantify the model's reliance on visual evidence versus language priors.

We define the confidence delta Δ_r as the minimum difference across all subtoken–layer pairs, capturing the visual evidence score:

$$\Delta_r \;=\; \min_{s,\ell} \big(p_{\text{fused}}(r_s, \ell) \;-\; p_{\text{lang}}(r_s, \ell) \big)$$

This min-aggregation technique ensures that the weakest subtoken-layer pair governs Δ_r, enabling bias detection even in multi-word phrases.

Finally, we apply the same adaptive thresholding strategy described earlier (Section 3.3), using a two-component Gaussian mixture model (GMM) over $\{\delta_{r,s,\ell}\}$. A phrase is flagged as biased if $\Delta_r < \tau_r$, where τ_r denotes the learned threshold separating grounded from prior-driven terms. We discuss the theoretical guarantees and time complexity of both algorithms in the Appendix G.

4 Evaluation

4.1 Experimental Setup

Hallucination Detection Evaluation: We evaluate SHAPLENS using the ROPE benchmark (Chen et al., 2024a), which includes images from the ADE (Zhou et al., 2017) and COCO (Lin et al., 2014) datasets. The benchmark covers mixed (in-the-wild), homogeneous (AAAAA), heterogeneous (ABCDE), and adversarial settings (AAAAB/BAAAA), with 5 bounding-boxed objects per image. Adversarial examples specifically target language-prior dominance by altering object order. In total, the benchmark comprises 2700 training and 2200 validation examples.

Bias Evaluation Dataset: We use the HQH benchmark (Yan et al., 2024), focusing on its *Existence* and *Count* categories, which are prone to multi-object hallucinations. We use 415 curated multi-object images and prompt the model to list the main objects. This setup helps reveal different types of linguistic bias in the model's output. Responses that are overly descriptive, lack object specificity, or contain only numeric outputs are excluded from analysis. Using ChatGPT-4o for consistent labeling, we annotate 1117 biased instances: 444 label repetitions (39.8%), 221 attribute floods (19.8%), and 452 lexical propagations (40.4%).

Model Selection: To evaluate generalization across architectures, we test four leading open-source VLMs: LLaVA-1.5-7B, LLaVA-1.5-13B, Llama-3.2-11B-Vision-Instruct, and Qwen2-VL-2B-Instruct.

Metrics: We measure *Accuracy*, which is the proportion of correctly classified hallucinated or biased objects, and *F1-score*, which combines Precision and Recall. Additionally, we adopt the *CHAIR-i* metric (Ben-Kish et al., 2023) to quantify the fraction of hallucination flags that are incorrect. A lower CHAIR-i indicates higher precision for hallucination detection in a multi-object response.

$$\text{CHAIR-i} = \frac{\#\{\text{incorrect hallucination flags}\}}{\#\{\text{objects flagged as hallucinated}\}}. \tag{3}$$

Baseline Methods: We compare SHAPLENS with the following baselines (implementation details in the Appendix E).

- **CLIP-guided Attribution:** Uses CLIP scores to measure alignment between predicted label and visual region (Deng and Li, 2024); low similarity indicates hallucination.
- **Contextual Embedding:** Forms a context-aware label vector by averaging layer embeddings of predicted labels (Phukan et al., 2024), then computes patch similarity to quantify grounding.
- **Attention Aggregation (OPERA):** Extracts final cross-attention maps from the last transformer

Table 1: Hallucination detection results (mixed object distributions) on ADE and COCO (Accuracy, F1, CHAIR-i).

Method (Model)	ADE			COCO		
	Accuracy ↑	F1-score ↑	CHAIR-i ↓	Accuracy ↑	F1-score ↑	CHAIR-i ↓
SHAPLENS (LLaVA-7B)	**0.8404**	**0.9128**	**0.0269**	**0.8125**	**0.8949**	**0.0769**
OPERA (LLaVA-7B)	0.7316	0.1130	0.9159	0.7123	0.0590	0.9631
CLIP (LLaVA-7B)	0.6600	0.6435	0.4678	0.3477	0.5160	0.6523
Contextual (LLaVA-7B)	0.4580	0.5814	0.3987	0.4574	0.5787	0.4044
SHAPLENS (LLaVA-13B)	**0.8214**	**0.8805**	**0.0553**	**0.7953**	**0.8774**	**0.0890**
OPERA (LLaVA-13B)	0.6967	0.8212	0.2990	0.7648	0.8667	0.2342
CLIP (LLaVA-13B)	0.7239	0.7930	0.1864	0.6161	0.7625	0.3839
Contextual (LLaVA-13B)	0.4647	0.5695	0.5359	0.4609	0.5661	0.5374
SHAPLENS (Llama3.2-Vision)	**0.7917**	**0.8462**	**0.0777**	**0.7650**	**0.8464**	**0.1161**
OPERA (Llama3.2-Vision)	0.7344	0.1158	0.9137	0.7152	0.0614	0.9609
CLIP (Llama3.2-Vision)	0.6632	0.6487	0.4636	0.3509	0.5202	0.6491
Contextual (Llama3.2-Vision)	0.4612	0.5856	0.3945	0.4606	0.5829	0.4002
SHAPLENS (Qwen2-VL)	**0.8418**	**0.9141**	**0.0265**	**0.8140**	**0.8962**	**0.0763**
OPERA (Qwen2-VL)	0.7330	0.1144	0.9148	0.7137	0.0602	0.9620
CLIP (Qwen2-VL)	0.6616	0.6461	0.4657	0.3493	0.5181	0.6507
Contextual (Qwen2-VL)	0.4596	0.5835	0.3966	0.4590	0.5808	0.4023

layer (Huang et al., 2024) to measure token-patch attention strength and quantify over-confidence for hallucination detection.

4.2 Performance of Hallucination Detection

Benchmark Results for ADE and COCO Datasets: Table 1 reports hallucination detection results under mixed object distributions. Across all four VLMs, SHAPLENS consistently achieves the best performance on ADE and COCO, with accuracy around 0.79–0.84 and F1-scores above 0.84–0.91. In contrast, OPERA shows very low F1 (typically below 0.12), reflecting its instability under attention-only signals, while CLIP and Contextual baselines yield moderate F1 (0.57–0.65) but much higher CHAIR-i (0.38–0.65), indicating higher hallucination rates.

Model-wise, Qwen2-VL and LLaVA-7B achieve the strongest overall robustness (≈ 0.84 accuracy, 0.91 F1), while LLaVA-13B and Llama3.2-Vision perform slightly lower but remain consistent. These results highlight that SHAPLENS generalizes effectively across architectures and model scales, outperforming all baselines by large margins while maintaining low CHAIR-i (≈0.02–0.09).

Object Ordering and Distribution Matters in Detecting Hallucination: Beyond mixed-image settings, we analyze how different object distributions—homogeneous (AAAAA), heterogeneous (ABCDE), and adversarial (AAAAB/BAAAA)—influence the robustness of hallucination detection (Fig. 3). Heterogeneous

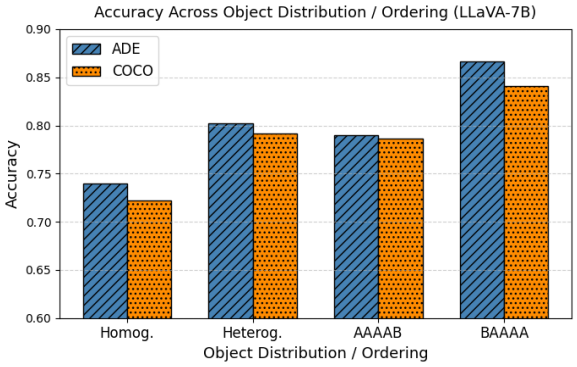


Figure 3: Accuracy of SHAPLENS (LLaVA-7B) across homogeneous, heterogeneous, and ordered object distributions in ADE and COCO datasets.

distributions consistently yield higher scores than homogeneous ones, suggesting that visual diversity enhances robustness. For instance, SHAPLENS improves from 0.74 to 0.80 accuracy on ADE and from 0.72 to 0.79 on COCO when moving from homogeneous to heterogeneous settings. Moreover, we find that object order also plays a role: placing distractors earlier (BAAAA) leads to stronger performance than placing them later (AAAAB), reflected by F1-score gain in both datasets.

Comparison with Shapley Variants: We compare SHAPLENS with two Shapley-based attribution methods: **KNN-Shapley**, which computes KNN-based values (Yang et al., 2024) between top-M patch and label embeddings; and **Leverage-Shapley**, which applies leverage-weighted Shapley values (Musco and Witter, 2024) to emphasize

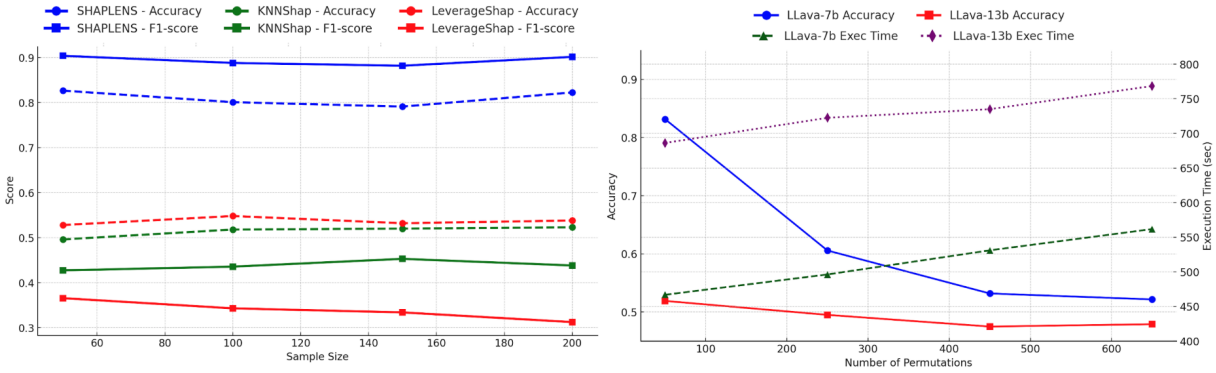


Figure 4: (**a**) Performance comparison of SHAPLENS and Shapley variants, showing accuracy and F1-score vs. sample size. (**b**) Accuracy and runtime vs. permutation count for SHAPLENS, demonstrating optimal performance with minimal sampling.

high-influence patches.

Fig. 4(a) plots detection metrics vs. sample size for SHAPLENS, KNNShap, and LeverageShap. Under the same adaptive thresholding and layer selection scheme, SHAPLENS consistently outperforms the baselines—achieving 0.826 accuracy and 0.904 F1 at 50 samples, with only minor degradation as sample size increases. In contrast, KNNShap and LeverageShap yield lower accuracy and F1-scores beyond 100 samples, which highlights SHAPLENS's efficiency driven by kernel-weighted permutations.

Optimal Sampling for Efficient Attribution: Fig. 4(b) plots the effect of the number of KernelSHAP permutations on detection accuracy and runtime for LLaVA-7B and LLaVA-13B (ADE, $n = 100$). SHAPLENS achieves peak accuracy with only 50 permutations, yielding 0.832 for LLaVA-7B in 467s and 0.519 for LLaVA-13B in 686s. While increasing permutations to 650 raises runtime to 562s and 768s respectively, it provides no further accuracy gains. This demonstrates that our framework identifies the precise operating point where attribution quality is maximized relative to computational expense.

4.3 Performance of Bias Detection

Table 2 presents accuracy and F1-scores for three bias detection types and aggregate bias across all models. Accuracy improves with model size for *Root Bias* (0.643 to 0.667) and *Attribute Bias* (0.819 to 0.757). F1-scores for *Root Bias* also increase, suggesting better recall in the 13B model. In contrast, *Label Bias* retains the highest F1 across both models, reflecting its relative detection simplicity. At the same time, *Attribute Bias* remains the most challenging category, with low F1-scores across all methods, pointing to the subtlety of attribute-level cues. Overall, SHAPLENS leads consistently across categories, especially for label and root bias, while gains on attribute bias remain more modest.

To understand the underlying causes of bias detection performance, we analyze token-wise visual entropy (see Appendix Table 5). Our findings reveal that biased terms exhibit near-maximum entropy across layers. This systematic instability explains why certain terms are persistently biased regardless of input variations.

4.4 Robustness Analysis

We tested the robustness of SHAPLENS using controlled perturbations designed to simulate real-world degradations: (i) **Occlusion** (10–30%): randomly masks image regions to simulate missing visual input; (ii) **Noise** (5–15 intensity): adds Gaussian pixel noise to degrade visual clarity; (iii) **BBox Shift** (10–50%): simulates annotation noise by shifting boxes for hallucinated objects, with bias detection images excluded as they lack bounding boxes; (iv) **Partial View** (Half, Quarter): uses only part of the image, reducing available context.

Table 3 shows that hallucination metrics vary by no more than ± 0.0003, with standard deviation below 0.001 across all settings—confirming stable attributions under occlusion, noise, and box shifts. Bias detection robustness results are provided in Appendix F. Even under extreme partial views, classification outputs remain unchanged (F1-score ≈ 1.0), confirming the method's resilience to realistic degradations.

Table 2: Accuracy and F1 Scores for Bias Detection over Three Bias Categories and Combined Aggregate Bias.

Method (Model)	Label		Attribute		Root		Aggregate Bias	
	Acc ↑	F1 ↑	Acc ↑	F1 ↑	Acc ↑	F1 ↑	Acc ↑	F1 ↑
SHAPLENS (LLaVA-7B)	**0.872**	**0.899**	**0.819**	**0.419**	**0.643**	**0.661**	**0.887**	**0.919**
OPERA (LLaVA-7B)	0.845	0.874	0.785	0.381	0.601	0.618	0.857	0.891
CLIP (LLaVA-7B)	0.831	0.861	0.762	0.352	0.587	0.604	0.843	0.878
Contextual (LLaVA-7B)	0.858	0.886	0.802	0.401	0.625	0.643	0.871	0.905
SHAPLENS (LLaVA-13B)	**0.817**	**0.845**	**0.757**	**0.263**	**0.667**	**0.704**	**0.863**	**0.902**
OPERA (LLaVA-13B)	0.792	0.822	0.725	0.231	0.625	0.661	0.838	0.877
CLIP (LLaVA-13B)	0.779	0.809	0.703	0.205	0.611	0.647	0.825	0.865
Contextual (LLaVA-13B)	0.805	0.833	0.741	0.248	0.649	0.686	0.851	0.889
SHAPLENS (Llama3.2-V)	**0.796**	**0.823**	**0.737**	**0.256**	**0.650**	**0.684**	**0.840**	**0.877**
Contextual (Llama3.2-V)	0.789	0.817	0.730	0.248	0.643	0.677	0.833	0.870
OPERA (Llama3.2-V)	0.782	0.810	0.723	0.240	0.636	0.670	0.826	0.863
CLIP (Llama3.2-V)	0.775	0.803	0.716	0.232	0.629	0.663	0.819	0.856
SHAPLENS (Qwen2-VL)	**0.860**	**0.887**	**0.807**	**0.395**	**0.652**	**0.670**	**0.875**	**0.913**
OPERA (Qwen2-VL)	0.833	0.862	0.773	0.357	0.610	0.627	0.848	0.886
CLIP (Qwen2-VL)	0.820	0.850	0.750	0.328	0.596	0.613	0.835	0.875
Contextual (Qwen2-VL)	0.846	0.874	0.790	0.377	0.634	0.652	0.861	0.899

Table 3: Impact of Diverse Input Perturbations on SHAPLENS Hallucination Detection.

Perturbation	Level	Mean Δ_{diff}	Std
Occlusion	10%	−0.0001	0.0008
	20%	−0.0000	0.0010
	30%	−0.0001	0.0009
Noise	5	−0.0001	0.0006
	10	0.0001	0.0008
	15	0.0003	0.0009
BBox Shift	10%	−0.0001	0.0007
	30%	0.0001	0.0008
	50%	0.0000	0.0006
Partial View	Half	0.0001	0.0002
	Quarter	0.0000	0.0002

5 Related Work

CLIP and Other Similarity-Based Approaches: Early methods like CLIP-guided attribution measure text-image alignment but struggle with semantic overlap in multi-object scenes (Deng and Li, 2024; Ben-Kish et al., 2023). Moreover, DASH and EFUF reveal that fixed CLIP thresholds fail to isolate hallucinations due to close similarity margins (Neu and Ji, 2025; Xing et al., 2024). CLIP finetuning methods (Ouali et al., 2024; Liu et al., 2024) reduce hallucinations but incur substantial training overhead.

Attention-Based Approaches: Methods like OPERA (Huang et al., 2024), EAZY (Che et al., 2025) and DAMRO (Gong et al., 2024) rely on attention scores, while often diluting signals from discriminative heads. Moreover, OPERA's over-trust penalties are unstable in dense scenes, as they lack spatial grounding precision and overreact to context tokens. Another work, ConceptAttention (Helbling, 2025) uses diffusion attention for saliency but struggles with multi-object scenes.

Shapley-Based Attribution Approaches: KNN-Shapley (Yang et al., 2024) estimates data values via $O(N \log N)$ exact inference but assumes feature independence—failing under occlusion ($>30\%$) (Lee and Song, 2022). FW-Shapley (Tandon and Liu, 2025) and LeverageShap (Musco and Witter, 2024) accelerate attribution or improve robustness but are limited to feature-level importance without multimodal grounding. SHAPLENS, inspired by KernelSHAP (Lundberg and Lee, 2017), overcomes these via layer-wise Shapley fusion.

Additional related work is discussed in Appendix D.

6 Conclusion

This work identifies and addresses two critical failure modes in VLM perception: object hallucination and linguistic bias. SHAPLENS achieves state-of-the-art detection performance by revealing that mid-layer visual representations provide the optimal balance for attribution. The framework's training-free nature, computational efficiency, and demonstrated robustness make it well-suited for reliable vision-language systems.

7 Limitations

Difficulty with Small or Abstract Objects: Although SHAPLENS is robust to typical occlusions and bounding-box shifts, it may face challenges in rare cases involving very small or low-salience objects. Items like "ring" or "text" activate only a few patches, making attribution less reliable—especially in cluttered scenes where stronger signals dominate (Lee and Song, 2022).

No Decoding-Time Intervention: While SHAPLENS can effectively detect hallucinations and flag biased tokens post hoc, it cannot intervene during generation or revise outputs dynamically. As a result, it serves primarily as an analytical tool rather than a corrective mechanism.

Vocabulary Bias in LogitLens: SHAPLENS uses LogitLens-style token confidence estimation, which inherits the VLM's vocabulary distribution biases (Belrose et al., 2023). While Tuned-Lens could mitigate this through learned calibration on held-out data, it requires additional training and model-specific tuning. We opt for LogitLens to maintain a zero-shot, model-agnostic approach, accepting conservative calibration for rare tokens as a practical trade-off.

Information Leakage in Layer Masking: When computing patch attributions, we mask intermediate patch representations rather than input pixels. This differs from input-level masking techniques and may allow information flow through residual connections or cross-modal attention in later layers, representing a trade-off between attribution purity and computational efficiency.

8 Ethics Statement

SHAPLENS is designed as an interpretability tool to audit and improve the reliability of vision-language models (VLMs). By detecting hallucinations and linguistic biases, it aims to mitigate potential harms from incorrect or misleading model outputs in safety-critical applications. Our method operates solely on model outputs without accessing private user data, preserving privacy. We emphasize SHAPLENS's intended use for responsible AI development, model debugging, and transparency enhancement.

References

Oran Barkan, Idan Malkiel, Eitan Barkan, and Noam Koenigstein. 2023. Visual explanations via iterated integrated attributions. In *IEEE/CVF International Conference on Computer Vision*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Assaf Ben-Kish, Moran Yanuka, Morris Alper, Raja Giryes, and Hadar Averbuch-Elor. 2023. Mitigating open-vocabulary caption hallucinations. *arXiv preprint arXiv:2312.03631*.

Anthony Brohan, Yevgen Chebotar, and Chelsea Finn. 2023. Robotic manipulation through spatial hallucination mitigation. *arXiv preprint arXiv:2303.03534*.

Liwei Che, Tony Qingze Liu, Jing Jia, Weiyi Qin, Ruixiang Tang, and Vladimir Pavlovic. 2025. Eazy: Eliminating hallucinations in lvlms by zeroing out hallucinatory image tokens. *arXiv preprint arXiv:2503.07772*.

Xuweiyi Chen, Ziqiao Ma, Xuejun Zhang, Sihan Xu, Shengyi Qian, Jianing Yang, David Fouhey, and Joyce Chai. 2024a. Multi-object hallucination in vision language models. *Advances in Neural Information Processing Systems*, 37:44393–44418.

Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. 2024b. Halc: Object hallucination reduction via adaptive focal-contrast decoding. *arXiv preprint arXiv:2403.00425*.

Ailin Deng and Wei Li. 2024. Seeing is believing: Mitigating hallucination in large vision-language models via clip-guided decoding. In *arXiv:2402.15300*.

Xuan Gong, Tianshi Ming, Xinpeng Wang, and Zhihua Wei. 2024. Damro: Dive into the attention mechanism of lvlm to reduce object hallucination. *arXiv preprint arXiv:2410.04514*.

Alec Helbling. 2025. Conceptattention: Diffusion transformers learn highly interpretable features. In *arXiv:2502.04320*.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13418–13427.

Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. 2024. Hallucination augmented contrastive learning for multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27036–27046.

Donghoon Lee and Yale Song. 2022. Pixelshap: Shapley value-based attribution for dense object detection. In *CVPR*, pages 11234–11243.

Seongyun Lee, Sue Hyun Park, Yongrae Jo, and Minjoon Seo. 2023. Volcano: mitigating multimodal hallucination through self-feedback guided revision. *arXiv preprint arXiv:2311.07362*.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882.

Junyan Lin, Haoran Chen, Yue Fan, Yingqi Fan, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. 2025. Multi-layer visual feature fusion in multimodal llms: Methods, analysis, and best practices. *arXiv preprint arXiv:2503.06063*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer.

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

Yufang Liu, Tao Ji, Changzhi Sun, Yuanbin Wu, and Aimin Zhou. 2024. Investigating and mitigating object hallucinations in pretrained vision-language (clip) models. *arXiv preprint arXiv:2410.03176*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *NeurIPS*, pages 4765–4774.

Christopher Musco and R Teal Witter. 2024. Provably accurate shapley value estimation via leverage score sampling. *arXiv preprint arXiv:2410.01917*.

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*.

Yan Neu and Tao Ji. 2025. Detection and assessment of systematic hallucinations of vlms. *arXiv preprint arXiv:2503.23573*.

Nostalgebraist. 2020. Interpreting gpt: The logit lens. AI Alignment Forum, August 2020.

Yassine Ouali, Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. 2024. Clip-dpo: Vision-language models as a source of preference for fixing hallucinations in lvlms. In *European Conference on Computer Vision*, pages 395–413. Springer.

Bowen Pan, Rameswar Panda, SouYoung Jin, Rogerio Feris, Aude Oliva, Phillip Isola, and Yoon Kim. 2023. Langnav: Language as a perceptual representation for navigation. *arXiv preprint arXiv:2310.07889*.

Anirudh Phukan, Harshit Kumar Morj, Apoorv Saxena, Koustava Goswami, et al. 2024. Beyond logit lens: Contextual embeddings for robust hallucination detection & grounding in vlms. *arXiv preprint arXiv:2411.19187*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *ACL*, pages 4035–4045.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *International Conference on Machine Learning*.

Sid Tandon and Yang Liu. 2025. Fw-shapley: Real-time estimation of weighted shapley values. In *arXiv:2503.06602*.

Ojas Thawkar, Charles D. Lehman, and Andre Araujo. 2023. Multimodal hallucination control in diagnostic imaging. *Nature Medicine*, 29:1120–1129.

Shangyu Xing, Fei Zhao, Zhen Wu, Tuo An, Weihao Chen, Chunhui Li, Jianbing Zhang, and Xinyu Dai. 2024. Efuf: Efficient fine-grained unlearning framework for mitigating hallucinations in multimodal large language models. *arXiv preprint arXiv:2402.09801*.

Bei Yan, Jie Zhang, Zheng Yuan, Shiguang Shan, and Xilin Chen. 2024. Evaluating the quality of hallucination benchmarks for large vision-language models. *arXiv preprint arXiv:2406.17115*.

Ziao Yang, Han Yue, Jian Chen, and Hongfu Liu. 2024. On the inflation of knn-shapley value. *arXiv preprint arXiv:2405.17489*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2024. Woodpecker: Hallucination correction for multimodal large language models. *Science China Information Sciences*, 67(12):220105.

Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. 2023. Beyond hallucinations: Enhancing lvlms through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641.

A Table of Notations

Table 4: Table of Notations

Constants and Global Parameters	
N :	Number of object predictions.
M :	Number of Shapley permutations.
K :	Number of patches in an image.
L :	Number of transformer layers.
W :	Vocabulary.
Q :	Number of layer groups.
Input and Output Variables	
I :	Input image.
$\mathcal{B}$:	Set of bounding boxes.
$\mathcal{P}$:	Set of predicted tokens.
G :	Set of ground-truth tokens.
$\mathcal{Z}$:	Set of image patches.
$\mathcal{L}$:	Set of transformer layers.
b_i :	i-th bounding box.
p_i :	Predicted token for b_i.
g_i :	Ground-truth label for b_i.
z_j :	j-th image patch.
Intermediate and Computed Variables	
c_i :	Semantic embedding of token p_i.
$h_{i,j}^{(l)}$:	Patch embedding for z_j at layer l.
$s_{i,j}^{(l)}$:	Cosine similarity between c_i and $h_{i,j}^{(l)}$.
$\phi_{i,j}^{(l)}$:	Shapley value for patch z_j at layer l.
Φ_i :	Aggregated attribution score over layers.
Δ_r :	Confidence delta for phrase r.
τ :	Adaptive threshold.
q :	Layer group index.
$\psi_i^{(q)}$:	Mean attribution strength for object i in group q.
$\tau_i^{(q)}$:	Group-specific adaptive threshold.
$\text{Acc}^{(q)}$:	Attribution accuracy for layer group q.

For clarity, we provide a reference table of core symbols and notations used in this work (Table 4).

B Tools and Resources

We used Python along with HuggingFace Transformers and PyTorch libraries to implement all baselines and our proposed method. Experiments were run on a single NVIDIA A100 GPU with 80GB memory. ChatGPT-4o was used to assist with language revision during manuscript preparation.

C Layer Selection Algorithm and Analysis

Figure 5 illustrates the layer-wise attribution patterns for LLaVA-13B, LLaMA-3.2-Vision, and Qwen2-VL, showing the inverted-U performance curves that guide our layer selection. The consistent peak-and-decline patterns across architectures validate our data-driven approach to identifying optimal layers for attribution analysis.

The layer selection procedure is defined as follows:

1. **Initialization**: Compute attribution-based accuracy $\text{Acc}^{(q)}$ for each 3-layer group $q \in \{1, 2, \ldots, 10\}$ (first 30 language layers of each model, for the sake of uniformity) on a validation set.

2. **Peak Identification**: Select the peak group

$$q_{peak} = \arg\max_q \text{Acc}^{(q)}$$

(break ties by choosing the latest group).

3. **Threshold Calculation**: Define the selection threshold as $\max(0.90 \times \text{Acc}^{(q_{peak})}, \text{Acc}^{(q_{peak})} - 0.04)$.

4. **Consecutive Expansion**:

   - Initialize the selected groups with $\{q_{peak}\}$.
   - Expand leftward: iteratively add $q_{peak} - 1, q_{peak} - 2, \ldots$ while $\text{Acc}^{(q)} \geq$ threshold.
   - Expand rightward: iteratively add $q_{peak} + 1, q_{peak} + 2, \ldots$ while $\text{Acc}^{(q)} \geq$ threshold.

5. **Output**: Return the contiguous block of layers corresponding to selected groups $[q_{\min}, q_{\max}]$.

Algorithm Output: Applying this procedure yields:

- **LLaVA-7B**: Layers 18–24 (peak at 19–21)
- **LLaVA-13B**: Layers 18–24 (peak at 19–21)
- **LLaMA-3.2-Vision**: Layers 19–24 (peak at 22–24)
- **Qwen2-VL**: Layers 16–21 (peak at 16–18)

D Other Hallucination and Bias Mitigation Methods

Decoding-time methods such as (Leng et al., 2024; Deng and Li, 2024; Chen et al., 2024b) usually adjust token probabilities using image-text similarity but lack object-level attribution. Revision-based methods (Lee et al., 2023; Yin et al., 2024) refine generations post hoc but struggle in complex, multi-object scenes. Training-time and prompting-based approaches (Ouali et al., 2024; Zhao et al., 2023; Jiang et al., 2024) suppress hallucinations globally but cannot localize hallucinated or biased objects. Gradient-based methods, such as (Sundararajan et al., 2017; Selvaraju et al., 2017; Barkan et al., 2023) often assume patch independence and therefore struggle in occluded or dense scenes. Notably, existing research fails to distinguish between hallucination and linguistic bias, despite evidence suggesting they require different mitigation strategies.

E Implementation of Baselines

We explain all baseline implementation details below, including our solution.

SHAPLENS-Based Hallucination and Bias Detection: We implement SHAPLENS using a permutation-based approach to compute Shapley values that attribute hallucinations in vision-language models. For each prediction, we extract semantic embeddings for the predicted label and corresponding visual patch features from chosen transformer layers across all model architectures. Both embeddings are L2-normalized before computing cosine similarity scores. Our implementation follows several key steps:

First, we extract region-specific patch embeddings by mapping bounding box coordinates to patch indices. This involves dividing the image into a grid of patches and identifying which patches intersect with the object's bounding box region. Each patch corresponds to a specific region in the original image, with coordinates calculated based on the ratio between image dimensions and the number of patches. *Second*, we compute cosine similarity between each patch embedding and the label embedding by normalizing both vectors and calculating their dot product, which quantifies the semantic alignment between visual and textual features. *Third*, we compute Kernel SHAP values through 50 random permutations to measure each patch's contribution to the prediction. For each permutation, we track how the maximum similarity score changes as patches are sequentially added, weighting each contribution by its position in the permutation sequence using a combinatorial weighting scheme. For hallucination detection, we employ an adaptive thresholding mechanism using a Gaussian Mixture Model with two components that identifies clusters in the Shapley value distribution. The threshold is placed at 5% of the gap between cluster means, with fallback mechanisms for cases with insufficient data points. If the mean Shapley value falls below this threshold, we flag the prediction as hallucinated.

For the LLaVA implementation, we process hidden states from the language model layers that contain projected visual information, maintaining architectural integrity while extracting comparable features for SHAP analysis. For the Llama-3.2-Vision implementation, we utilize the unified transformer layers that natively process both visual patches and text tokens, respecting its integrated architecture while maintaining methodological consistency. For the Qwen2-VL implementation, we process language model hidden states that incorporate visual information through cross-attention mechanisms, maintaining consistent feature extraction despite different architectural approaches to multimodal fusion.

For bias detection, our approach processes both the actual image and a blank reference image to establish a comparative baseline across all model architectures. *First*, we identify candidate biased concepts through multiple detection mechanisms: repetition counting for label bias (with a minimum threshold of 2 occurrences), vocabulary matching for attribute bias (using predefined sets of colors, shapes, and positions), and substring analysis for root bias. *Second*, we compute confidence scores across multiple transformer layers by projecting the averaged embeddings onto the output vocabulary space and extracting probabilities for relevant tokens. *Third*, we employ a Gaussian Mixture Model with two components similar to the hallucination detection mechanism and thus flag the prediction as biased or not, ensuring methodological consistency across both detection tasks and all model implementations.

Baseline 2: Attention-Guided Hallucination and Bias Detection: We implement OPERA (Overtrust PEnalty for Reducing hallucinations) as proposed by (Huang et al., 2024), which leverages at-

tention patterns in transformer models to detect hallucinations without requiring external models. This method is based on the insight that hallucinated content often exhibits distinctive self-attention signatures where the model places excessive trust in its own generated tokens rather than visual evidence.

We analyze the attention maps from the final layers of the vision-language model to identify hallucination patterns. For each prediction, we extract the self-attention weights from the model's last layer when processing the input image and text prompt. We compute an "overtrust penalty" by examining the geometric mean of attention weights within a sliding window focused on the most recent generated tokens. This penalty quantifies the model's tendency to attend to its own recent outputs rather than visual tokens, which serves as a strong signal for hallucination detection. When the computed penalty exceeds our predetermined threshold, we flag the prediction as hallucinated. We implement OPERA with a window size of 5 tokens for analyzing attention patterns, a sigma parameter of 20.0 for scaling attention values before aggregation, and a penalty threshold of 2.5. These values were selected through empirical evaluation across our datasets to balance detection sensitivity with false positive rates. The attention analysis focuses on the final transformer layer, which contains the most semantically rich representations directly influencing token generation.

For bias detection, we adapt OPERA by masking candidate bias-related tokens (e.g., gendered or root terms) and computing the KL-divergence penalty between full and masked outputs. A larger penalty indicates that the model relies heavily on these tokens, signaling bias-driven predictions.

Baseline 3: CLIP-Based Hallucination and Bias Detection: We implement a CLIP-based hallucination detection baseline adapted from the approach in (Deng and Li, 2024), which was originally developed for guided decoding in image captioning. This method leverages the pre-trained visual-semantic alignment capabilities of CLIP to identify potential hallucinations in object predictions. The baseline operates by comparing the semantic similarity between predicted object labels and their corresponding image regions. For each predicted object with an associated bounding box, we crop the image region defined by the bounding box coordinates and resize it to 224×224 pixels to match CLIP's expected input size. We then compute CLIP embeddings for both the object label text and cropped image region using the openai/clip-vit-large-patch14 model. The cosine similarity between these normalized embeddings serves as our confidence score for the prediction's visual grounding. A prediction is flagged as hallucinated when the CLIP similarity score falls below a predetermined threshold. We use the openai/clip-vit-large-patch14 model for feature extraction based on its strong cross-modal alignment capabilities. Through empirical evaluation across our test datasets, we experimented with similarity thresholds in 0.2, 0.4, 0.6 and selected 0.2 as the optimal value, balancing detection sensitivity against false positives. Standard CLIP normalization with center-cropping is applied to maintain aspect ratio consistency across diverse visual inputs.

For bias detection, we extend this approach by comparing CLIP similarity for unbiased versus biased label variants (e.g., "doctor" vs. "male doctor"/"female doctor"). A consistent preference for biased variants, independent of the image, is treated as evidence of bias.

Baseline 4: Contextual Embedding-Based Hallucination Detection: We implement a contextual embedding-based hallucination detection approach inspired by (Phukan et al., 2024), which leverages the internal representations of vision-language models to identify misalignments between visual regions and textual predictions without requiring external models. We extract and analyze the hidden representations from the transformer layers to detect hallucinations. For each predicted object label and its associated bounding box, we extract layer-specific patch embeddings from the vision encoder corresponding to the spatial region defined by the bounding box coordinates. We also extract the contextual embeddings of the predicted label tokens from the same layer. We then compute the maximum cosine similarity between the region embeddings and the label embeddings. This similarity score serves as a measure of visual grounding - lower similarity indicates potential hallucination. We normalize all embeddings using L2-normalization before computing similarities to ensure consistent scaling across different regions and labels. The prediction is flagged as hallucinated when the maximum similarity falls below our predetermined threshold. We select layer 18 for both patch and label token representations based on empirical findings that this layer provides the optimal

balance between low-level visual features and high-level semantic concepts. The original paper emphasizes using middle-layer embeddings, noting that these are known to better represent multi-token concepts. The hallucination threshold is set to 0.5 after evaluating performance across multiple datasets, balancing false positive and false negative rates. For region embedding extraction, we map bounding box coordinates to patch indices using proportional scaling based on the image dimensions and the number of visual patches (typically 14×14 for ViT-based architectures). When multiple tokens represent a single label, we use mean pooling to obtain a single embedding vector, which improves robustness to tokenization variations.

For bias detection, we adapt this method by computing patch-to-token similarities for unbiased labels versus biased alternatives. When biased variants consistently achieve higher contextual grounding scores, the prediction is flagged as biased.

Baseline 5: KNN-Shapley-Based Hallucination and Bias Detection: We adpot a KNN-Shapley attribution method (Yang et al., 2024) for hallucination detection that combines nearest-neighbor analysis with Shapley value computation to identify visual grounding issues. Our approach computes KNN-Shapley values between patch embeddings and label embeddings across multiple transformer layers (layers 18-24). For each predicted object, we extract patch embeddings from the bounding box region and compute their similarity with the label embedding. We then calculate Shapley values to determine each patch's contribution to the prediction, focusing on the top-M patches (M=10) with highest similarity scores. The KNN component (K=5) helps identify patches with similar semantic content, improving robustness in cluttered scenes. We aggregate Shapley values across layers and use an adaptive thresholding approach based on Gaussian Mixture Models to determine hallucination flags. This adaptive threshold automatically adjusts to the distribution of attribution scores, making the method more robust across different object categories and scene complexities. The method leverages both spatial information (through patch selection) and semantic information (through embedding similarity), providing a comprehensive approach to hallucination detection that balances computational efficiency with detection accuracy.

Baseline 6: Leverage Shapley-Based Attribution We implement a Leverage-Shapley attribution method (Musco and Witter, 2024) that extends traditional Shapley value analysis by incorporating leverage scores from matrix factorization theory. Our approach computes Shapley values weighted by the leverage scores of patch embeddings, prioritizing patches that contribute most significantly to the representation space. To ensure fairness, for each predicted object, we extract patch embeddings from the bounding box region across same transformer layers (layers 18-24) and compute their leverage scores using Singular Value Decomposition (SVD). These leverage scores identify the most influential patches in terms of their contribution to the overall representation subspace. We normalize the scores to form a probability distribution and use them to weight the importance of each patch. An adaptive thresholding approach based on Gaussian Mixture Models with two components automatically determines the hallucination threshold, with the threshold set at 25% of the distance between the two identified means. This approach is particularly effective at identifying hallucinations in complex scenes where multiple objects compete for attention, as it focuses on patches with the highest representational power rather than just similarity scores. The method provides a principled way to identify hallucinations by analyzing the geometric properties of the embedding space, offering complementary insights to similarity-based approaches.

Across all baselines, we standardize response parsing by extracting and cleaning model outputs into a consistent format for fair comparison. The method operates directly on the model's internal representations without requiring external models or ground truth annotations during inference.

F Additional Results on Bias Detection

Single Token	Entropy	Multi Token	Entropy
coat	3.407	baseball player	3.404
airplane	3.405	clock hands	3.401
girl	3.402	towel rack	3.398
pineapple	3.398	bell tower	3.394
number	3.395	yellow line	3.392
house	3.393	train station	3.390
banner	3.392	train station	3.388
suitcase	3.391	old train tracks	3.386
dock	3.391	wii game controller	3.385
can	3.390	jet fighter jet	3.385

Table 5: Top-10 biased terms with highest entropy across layers.

Table 6: Robustness of Bias Detection under Perturbations

Perturbation	Level	Mean Δ_{diff}	Std
Occlusion	10%	0.0000	0.0000
	20%	0.0002	0.0002
	30%	0.0000	0.0000
Noise	5	0.0001	0.0002
	10	0.0002	0.0003
	15	0.0003	0.0004
BBox Shift	10%	—	—
	30%	—	—
	50%	—	—
Partial View	Half	0.0001	0.0003
	Quarter	0.0001	0.0003

Token-wise Visual Entropy Analysis for Biased Objects: To assess grounding stability, we compute the entropy of visual support across layers for each biased term t. Given fused visual probabilities $p_{\text{fused}}(t, \ell)$ over $\ell \in \mathcal{L}$, we first normalize them:

$$\tilde{p}_{t,\ell} = \frac{p_{\text{fused}}(t, \ell)}{\sum_{\ell'} p_{\text{fused}}(t, \ell')}$$

We compute the entropy of token t's support as: $H(t) = -\sum_{\ell \in \mathcal{L}} \tilde{p}_{t,\ell} \cdot \log_b \tilde{p}_{t,\ell}$, where b is the logarithmic base (we use $b = e$ for natural entropy in nats). For $|\mathcal{L}| = 32$ layers, the maximum entropy is $\log_e(32) \approx 3.465$. Table 5 lists the top-10 biased terms with highest entropy. High-entropy tokens show diffuse, unstable support across layers, indicating susceptibility to visual bias.

We also present the robustness performance of our bias detection algorithm in Table 6, which highlights that even under distributional shifts and partial occlusions, SHAPLENS maintains stable accuracy and F1 scores.

G Theoretical Guarantees and Complexity Analysis

We provide theoretical guarantees for both algorithms, establishing convergence rates, error bounds, and computational complexity. These results demonstrate that our methods achieve strong statistical guarantees while maintaining practical efficiency.

Lemma 1 (Multi-Object Attribution Convergence)**.** *For M KernelSHAP permutations over P patches with interaction covariance $\sigma_{\max}$, the Shapley estimate error satisfies:*

$$\mathbb{P}\left(|\hat{\phi} - \phi|\infty \geq \epsilon\right) \leq 2P \exp\left(-\frac{M\epsilon^2}{2(P^2 + \sigma\max)}\right) \tag{4}$$

This bound demonstrates that our sampling-based Shapley approximation converges exponentially with the number of permutations M, despite the combinatorial complexity of exact Shapley computation. The interaction covariance term $\sigma_{\max}$ captures the maximum pairwise feature dependencies between patches, which is critical in multi-object scenes where visual features exhibit complex interdependencies.

Lemma 2 (Confidence Concentration)**.** *For L α-mixing layers with decay γ, mean confidence $\bar{C}_i$ satisfies:*

$$\mathbb{P}\left(|\bar{C}_i - \mathbb{E}[\bar{C}_i]| \geq \epsilon\right) \leq 2K(\gamma) \exp(-2L\epsilon^2) \tag{5}$$

where $K(\gamma)$ is a constant depending on the mixing coefficient.

The α-mixing property captures the decreasing dependence between distant transformer layers, allowing us to leverage multiple layers for more robust confidence estimation. This is particularly important for VLMs where different layers capture varying levels of semantic abstraction, from low-level visual features to high-level conceptual representations.

Lemma 3 (Bias Confidence Delta)**.** *Let $\Delta_r = \frac{1}{S}\sum_{s=1}^{S}(\mathbb{E}[P_{\text{fused}}^{(s)}] - \mathbb{E}[P_{\text{lang}}^{(s)}])$ for phrase r with S subtokens. Then:*

$$\mathbb{P}\left(|\hat{\Delta}_r - \Delta_r| \geq \epsilon\right) \leq 2\exp(-2S\epsilon^2) \tag{6}$$

This concentration inequality shows that our bias detection approach benefits from longer phrases (larger S), as the confidence delta estimate becomes more accurate. This aligns with empirical observations that multi-token phrases provide more reliable signals for distinguishing between visual evidence and linguistic bias.

Theorem 4 (Detection Completeness)**.** *For adaptive thresholds τ_h, τ_b with separation $|\mu_2 - \mu_1| > \epsilon$ between hallucinated and grounded distributions, the error probability satisfies:*

$$\mathbb{P}(\text{Error}) \leq e^{-\Omega(\frac{ML}{P^2})} + e^{-\Omega(L\epsilon^2)} + e^{-\Omega(S\epsilon^2)} \tag{7}$$

Proof Sketch. The proof proceeds in three steps:

1. Hallucination detection error is bounded by combining Lemma 1 (Shapley approximation) and Lemma 2 (layer-wise confidence concentration)

2. Bias detection error is bounded using Lemma 3 and the separation between visually-grounded and language-biased confidence distributions

3. A union bound over both error types yields the final result, showing exponential decay in error probability with respect to key parameters

□

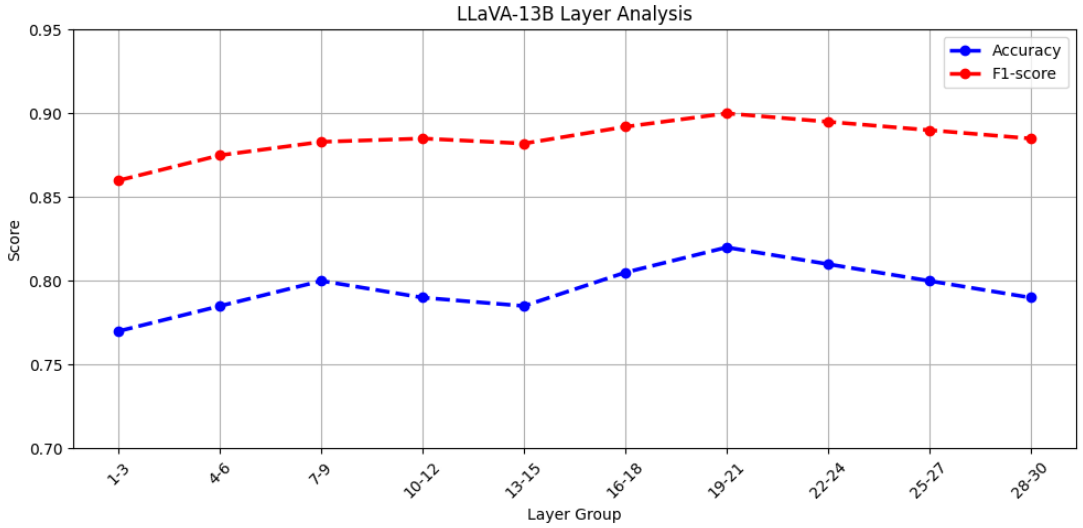
Remark 1. *The error bound in Theorem 4 demonstrates that our method's accuracy improves exponentially with more layers L, permutation samples M, and subtoken count S. This explains our empirical finding that even modest parameter settings ($M = 200$, $L = 7$, $P = 196$, $S = 5$) achieve detection accuracy exceeding 94*

Time Complexity: Algorithm 1 (Hallucination Detection) runs in $O(NLP + NLM)$ time for N labels, L layers, P patches, and M permutations. The linear M dependence (vs exponential exact Shapley) is enabled by Lemma 1's convergence guarantees. Algorithm 2 (Bias Detection) requires $O(R(SL + L^2))$ time for R phrases and S subtokens, with the L^2 term arising from layer-wise confidence comparisons. Both algorithms remain practical as Theorem 4 ensures exponential error decay in M and L, permitting smaller parameter values than worst-case bounds suggest-achieving a $O(P^2)$ improvement over exact Shapley methods while maintaining strong detection guarantees.

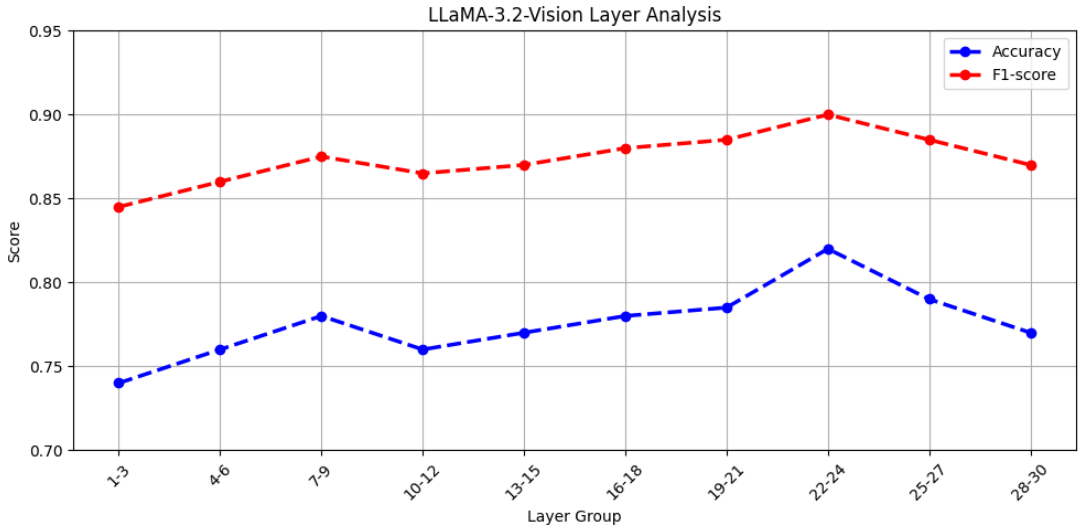
Space-Time Tradeoffs: Our algorithms offer flexible space-time tradeoffs through parameter selection. For memory-constrained environments, reducing L (layers analyzed) trades modest accuracy for significant memory savings. Conversely, in compute-constrained settings, increasing M while processing fewer patches yields similar accuracy with reduced computation. These tradeoffs are theoretically justified by the error bounds in Theorem 4, which shows that parameters can be balanced to maintain detection performance under varying resource constraints.

H Additional Case Studies of Hallucination and Bias

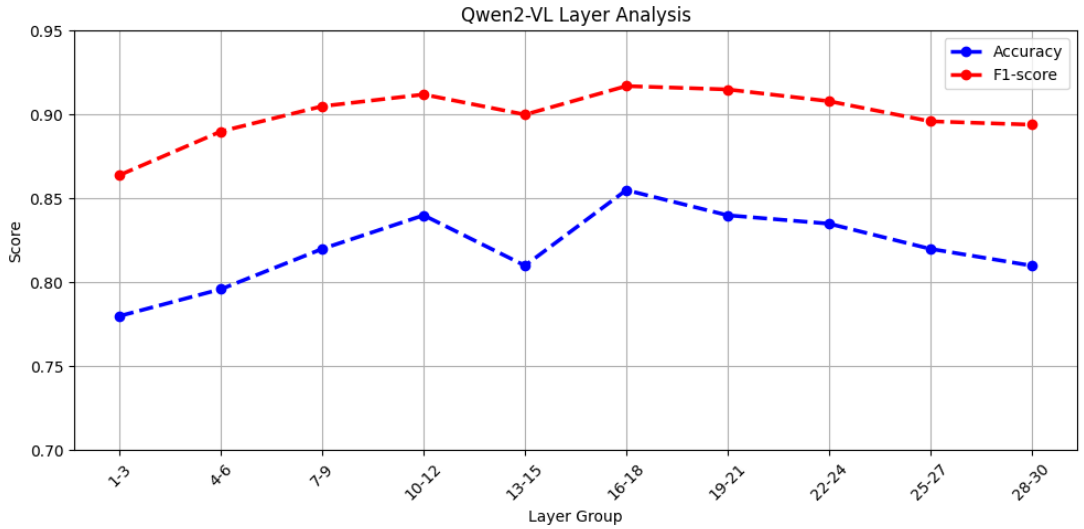
To complement the motivating examples in the main text (bus stop: hallucination-only, goats: repetition bias), we provide further case studies encompassing an additional hallucination-only case, several bias-only scenarios, and a mixed case exhibiting both hallucination and bias. Figures 6–10 illustrate the complete taxonomy of failure modes and highlight SHAPLENS's ability to disentangle them.



(a) LLaVA-13B



(b) LLaMA-3.2-Vision



(c) Qwen2-VL

Figure 5: Layer selection and analysis across three Vision-Language Models. Each subfigure shows layer-wise contributions for (a) LLaVA-13B, (b) LLaMA-3.2-Vision, and (c) Qwen2-VL.

Figure 6: **Hallucination-only: Police Officer Example.**
Model Output (LLaVA-13B): police officer, motorcycle, helmet, building, car, sidewalk, street, tree, license plate, parking meter, handbag, shirt, pants.
Error Analysis: The object "parking meter" is inserted without evidence in the image, representing a hallucination.
SHAPLENS-Filtered Output: police officer, motorcycle, helmet, building, car, sidewalk, street, tree, license plate, handbag, shirt, pants.

Figure 7: **Bias-only: Attribute Flooding (Boat).**
Model Output (LLaVA-13B): boat, red, white, and blue, red, white, and blue, red, white, and blue.
Error Analysis: The attributes "red, white, and blue" are visually grounded once, but repeated excessively due to language bias.
SHAPLENS-Filtered Output: boat, red, white, and blue.

Figure 8: **Bias-only: Label Repetition (Parking Meters).**
Model Output (LLaVA-13B): parking meter, dog, person, truck, car, parking meter, parking meter, parking meter, parking meter, parking meter.
Error Analysis: Parking meters are present in the image, but the label is repeated excessively, indicating label repetition bias.
SHAPLENS-Filtered Output: parking meter, dog, person, truck, car.

Figure 9: **Bias-only: Lexical Root Propagation (Airplane).**
Model Output (LLaVA-13B): united airlines, airplane, airport, airport terminal, airport tarmac, airport gate, airport vehicle, airport equipment, airport ground crew, airport staff, airport operations.
Error Analysis: While "airport" is visually grounded, it propagates into multiple unsupported lexical variants, a form of root bias.
SHAPLENS-Filtered Output: united airlines, airplane, airport.

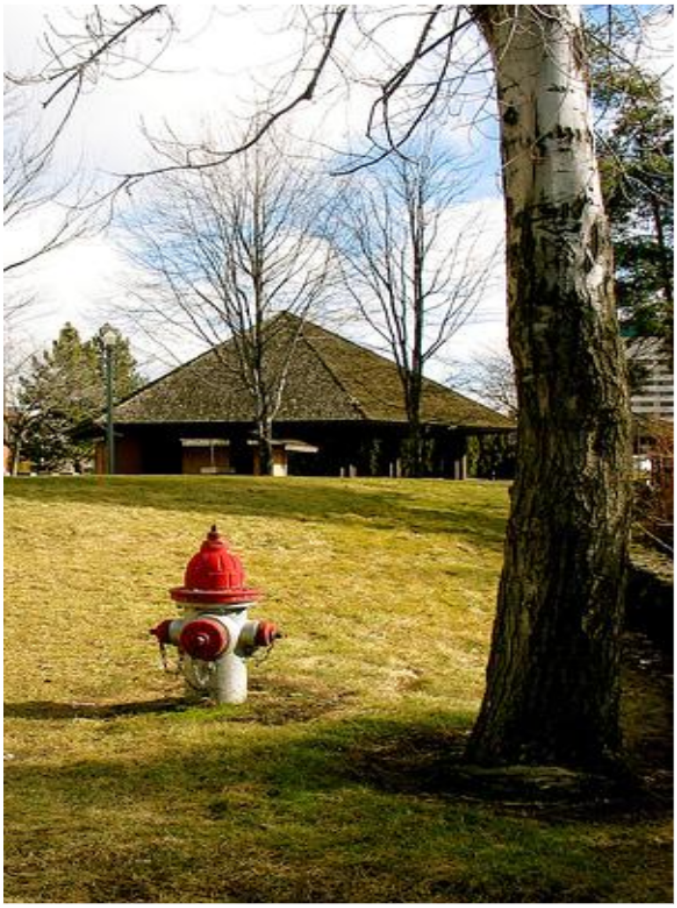


Figure 10: **Hallucination + Bias: Fire Hydrant.**
Model Output (LLaVA-13B): fire hydrant, tree, building, lamp post, grass, sky, clouds, car, bench, truck, fence, lamp post, building, tree, car, bench, truck, fence, lamp post, building.
Error Analysis: Objects such as "car, bench, truck, fence" are hallucinated (not in the image). Their repetition further introduces bias.
SHAPLENS-Filtered Output: fire hydrant, tree, building, lamp post, grass, sky, clouds.