HalluShift++: Bridging Language and Vision through Internal Representation Shifts for Hierarchical Hallucinations in MLLMs

Sujoy Nath
Netaji Subhash Engineering College
Kolkata, India
sujoynathofficial@gmail.com

Arkaprabha Basu
TCG Crest
Kolkata, India
arkaprabha17@gmail.com

Sharanya Dasgupta
Indian Statistical Institute
Kolkata, India
sharanyadasg@gmail.com

Swagatam Das
Electronics and Communication Sciences Unit
Indian Statistical Institute
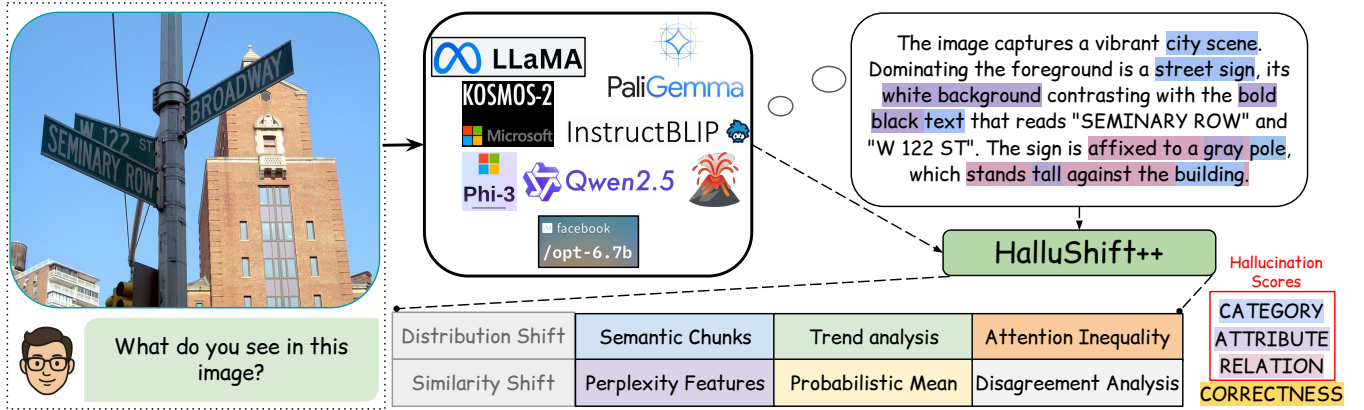Kolkata, India
swagatamdas19@yahoo.co.in

Figure 1: HALLUSHIFT++ in action: Left: MLLMs generate descriptions for visual inputs that may contain hallucinations. Right: Our framework provides hierarchical hallucination scores across three types (Category, Attribute, Relation).

Abstract

Multimodal Large Language Models (MLLMs) have demonstrated remarkable capabilities in vision-language understanding tasks. While these models often produce linguistically coherent output, they often suffer from hallucinations, generating descriptions that are factually inconsistent with the visual content, potentially leading to adverse consequences. Therefore, the assessment of hallucinations in MLLM has become increasingly crucial in the model development process. Contemporary methodologies predominantly depend on external LLM evaluators, which are themselves susceptible to hallucinations and may present challenges in terms of domain adaptation. In this study, we propose the hypothesis that hallucination manifests as measurable irregularities within the internal layer dynamics of MLLMs, not merely due to distributional shifts but also in the context of layer-wise analysis of specific assumptions. By incorporating such modifications, HALLUSHIFT++ broadens the efficacy of hallucination detection from text-based large language models (LLMs) to encompass multimodal scenarios. Our codebase is available at https://github.com/C0mRD/HalluShift_Plus.

CCS Concepts

• **Computing methodologies → Natural language generation**.

Keywords

Hallucination Detection, Multi-modal Large Language Models, Object Hallucination

1 Introduction

Nowadays, MLLMs [5, 11, 22, 24] show a promising ability to harness the language generation capabilities of LLMs [2, 29] along

with visual understanding to excel in multimodal tasks, such as image captioning [17] and visual question answering [11, 22]. These models are frequently susceptible to producing plausible sounding descriptions, therefore, being grammatically correct, but such descriptions often comes with inconsistent visual content, a phenomenon commonly referred to as hallucination [19, 23, 26].

We have observed hallucinations in MLLMs exhibit two distinct failure modes: objects that are present in images are missing from description, and describing objects that are not present in a certain image. While demonstrating object hallucination, addressing the first type is beyond the scope of this work and our primary focus resides on the second type concerning correctness being vital to human judges rather than specificity [27]. Object hallucination can be further categorized into three types: object category, object attribute (color, shape, or counting), and object relation (human-object interactions or relative positions) [4, 31]. Hallucination degrades the model performance and greatly harm the user experiences in real-world applications [14].

Current approaches in hallucination detection often uses an external LLM like GPT-4 [2] as a judge [10, 21]. These models typically produce definitive answers as either *hallucinated* or *truthful*, but unable to precisely convey the nature of the hallucination, as evidenced by their confidence score at the token level. In contemporary research, it is apparent that there exists a significant demand for an effective hallucination detection mechanism that can be effortlessly incorporated into current MLLMs. Such a mechanism should additionally provide hallucination-type annotations, as various types necessitate distinct considerations regarding the training datasets and architectural design.

Our central challenge is to meaningfully extract and analyze the foundational representations embedded inside the internal probability distributions of MLLMs that directly correlate with specific types of hallucinations. First, can we derive more robust and interpretable features from these internal distributions to reliably distinguish between content that is factually grounded and content that is hallucinated? And second, if content is indeed identified as hallucinated, can these same features further enable us to precisely classify the specific type of hallucination present of its nature? To delineate this challenge in MLLMs, we propose HALLUSHIFT++, an enhanced hallucination detection framework. While HALLUSHIFT [12] works solely based on internal distribution shifts within language models, HALLUSHIFT++ adapted the distribution nature of both LLM and MLLM with additional internal representation supports. Building upon the foundation of HALLUSHIFT [12], our approach extends the distributional shift analysis to the multi-modal domain where hallucinations arise from misaligned visual-linguistic representations and inadequate visual grounding. We summarize our key contributions as follows:

- *Extended HALLUSHIFT from LLMs to MLLMs*, we introduce 15 novel features (e.g. layer consistency, attention concentration) that capture previously unexplored aspects of multimodal hallucination patterns.
- A *semantic chunking strategy* that automatically decomposes generated descriptions into semantically meaningful units (object, attribute, and relation chunks), enriched with contextual features, using part-of-speech analysis, named entity

recognition [16], and dependency parsing to enable precise data labeling across different hallucination categories.
- *Multi-class hallucination taxonomy* beyond binary hallucination detection, our membership function assigns probabilistic scores across 3 token-level object hallucination classes: category, attribute, and relation hallucination.
- We perform extensive experiments across various foundational MLLMs, resulting in highest improvement of $\sim 64.12\%$ in AUC-ROC compared to previous HALLUSHIFT in image-captioning tasks. Additionally, we achieve $\sim 3\%$ improvement on LLMs QA tasks.

2 Related Works

Hallucination Detection Methods in MLLMs Early works on hallucination detection in MLLMs focused primarily on detecting nonexisting objects in generated image captions. CHAIR [6] employs a rule-based method to determine the proportion of words generated that correspond to the ground truth annotations and the segmentations of objects, showing limitations only in the detection of object-level hallucinations. In contrast, [19] shows that the instruction design and the length of the generated captions can affect the performance of CHAIR. Thus, the authors propose a new evaluation metric, called POPE [19], transforming hallucination detection into a binary classification. This approach involves simply prompting MLLMs with multiple *Yes* or *No* questions blandly with all available classes about probing objects. Later approaches use LLMs as a judge to detect hallucinations. GAVIE [21] employs GPT-4 [2] to evaluate the instruction-following performance and assign a single accuracy score to responses. FAITHScore [15] aims to evaluate free-form responses to open-ended questions using a chain of large models to break down responses into atomic facts and verify via visual entailment. Furthermore, HaELM [30], trained a LLM based on LLaMA for hallucination detection on a set of data generated by simulating different hallucination scenarios using ChatGPT. Although, such approaches can detect hallucination on long captions, they are incapacitated of providing hallucination type annotation. HalLoc [25] addresses this challenge by employing a token-level detection mechanism to identify hallucinations of objects, attributes, relationships, and scenes in various tasks. However, most of this available methods relies solely on external LLMs as evaluators, which again makes verification prone to hallucination. The evaluator is a novel network that has been trained on either substantially larger or distinct datasets, which invariably raises questions regarding the implications of the hallucinated model when iterating through its own internal distributions.

Internal Representation Analysis Recent approaches have explored the internal states of LLM to detect hallucinations without relying on external evaluators. These methods operate on the hypothesis that hallucinated content exhibits distinctive patterns in model's internal representations that can be captured and analyzed [18]. Techniques such as covariance pattern analysis [9] and eigenvalue decomposition of hidden states [28] revealed latent level inconsistencies correlated with hallucinations. Haloscope [13], experimented with geometric subspace analysis of selected LLM layers, but it does not successfully capture any probabilistic features or shift in internal dynamics of LLMs, as it concentrates

solely on hallucinations from a geometric transformation's perspective. Taking a step forward, HALLUSHIFT [12] originally developed for text-only LLMs, represents a notable contribution by hypothesizing that hallucinations manifest as quantifiable distribution shifts in the model's internal layer dynamics, extracting features from hidden and attention state spaces across LLM decoder layers. While HALLUSHIFT significantly outperforms existing SOTA methods across multiple benchmark on text-based QA tasks, it was limited to uni-modal text generation scenarios. Our observations indicate that HALLUSHIFT faces significant challenges when extended to multimodal contexts and predominantly fails to capture complex visual-linguistic interactions, where hallucinations can emerge from inadequate visual grounding and language bias. In the context of multimodal large language models (MLLMs), research focusing on hallucination detection through the exploration of the models' internal representations is comparatively scarce. Conversely, there is a wealth of studies employing external LLM evaluators. This disparity highlights a significant research opportunity to develop specialized approaches that leverage internal representations, thereby effectively addressing the distinct challenges associated with detecting hallucinations in MLLMs.

3 Problem Formulation

Let $\mathcal{M}$ be a MLLM that processes multimodal inputs to generate textual descriptions. For any input pair $(\mathbf{I}, \mathbf{x})$ where $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ represents an image and $\mathbf{x} = (x_1, \ldots, x_n), x_i \in \mathcal{V}$ is a sequence of words from vocabulary $\mathcal{V}$ constructing text prompt. The model generates a description sequence $\mathbf{y} = (y_1, \ldots, y_m) \in \mathcal{V}^m$, where m is the maximum number of output tokens and each token y_t is sampled from the conditional distribution $P(y_t|\mathbf{I}, \mathbf{x}, y_{<t})$. The central challenge resides in the inherent tendency of MLLMs to generate *hallucinated* content, textual descriptions that are not grounded in visual input $\mathbf{I}$. Given the stochastic nature of the generation process, where $P(\mathbf{y}|\mathbf{I}, \mathbf{x}) = \prod_{t=1}^{m} P(y_t|\mathbf{I}, \mathbf{x}, y_{<t})$, the model may produce semantically plausible but factually incorrect descriptions. However, the selection of y_t is derived from a top-k sampling or greedy approach in the vocabulary layer that results from the internal layer distributions. Let $\mathcal{G} = \{g_1, \ldots, g_p\}$ represent the ground truth annotations that accurately describe the visual content of image $\mathbf{I}$, where p is the number of ground truth annotations. We define a hallucination detection problem as learning a function $H : (\mathbf{I}, \mathbf{x}, \mathbf{y}) \rightarrow \mathcal{Y}$ that maps any generated description to a hallucination category, where $\mathcal{Y} = \{\text{Correct}, \text{Category}, \text{Attribute}, \text{Relation}\}$ represents a hierarchical taxonomy of hallucination types.

4 Proposed Method

We present HALLUSHIFT++ to address the unique challenge of visual grounding process inherent in MLLMs, where hallucination may arise from misaligned visual-linguistic representations [4], by analyzing their internal states. This approach improves upon the original HALLUSHIFT methodology designed explicitly for LLMs, with additional feature construction, semantic decomposition that automatically parses generated descriptions into object, attribute,

relation chunks and a hierarchical membership function that performs multi-class classification to detect various object hallucination, resulting in superior hallucination detection performance in both MLLMS and LLMs.

4.1 Feature Construction

The core innovation of HALLUSHIFT++ resides in extension towards 74-dimensional feature representation that encapsulates both established hallucination indicators [12] and novel computational patterns. Our feature extraction function $\mathcal{F} : (\mathbf{I}, \mathbf{x}, \mathbf{y}, \mathcal{S}) \rightarrow \mathbb{R}^{74}$ operates in the internal state of the model $\mathcal{S} = \{\mathbf{h}^{(l)}, \mathbf{A}^{(l)}, \mathbf{p}^{(t)}\}_{l=1,t=1}^{L,m}$ comprising hidden states, attention weights, and token probabilities in the layers and generation steps.

Original HALLUSHIFT Features (1-62): We extract 62 foundational features following HALLUSHIFT [12]. It combines distributional shift analysis of internal layers with token probability based detection.

Enhanced HALLUSHIFT++ Features (63-74): While the original HALLUSHIFT features effectively capture distributional shifts in text-only LLMs, the MLLMs present unique challenges that arises from the need to bridge visual and linguistic modalities. Unlike LLMs that operate within a single textual domain with consistent semantic spaces, MLLMs must *ground* text generation in visual evidence, while managing the inherent *modality gap* between continuous visual representations and linguistic tokens. This introduces three unique failure modes absent in text-only scenarios: 1) *Cross-layer inconsistency* occurs when visual features extracted by early vision encoders become progressively distorted as they pass through language decoder layers, causing the model to loose visual grounding and generate output inconsistent with the image, 2) *Attention dispersion* occurs when models trained on datasets with noisy image-text alignments learn to rely heavily on language priors rather than visual evidence, causing cross-modal attention to become unfocused across irrelevant visual regions, especially in domain adaptation scenarios.

3) *Confidence degradation* emerges from uncertainty propagation across the vision-language integration pipeline, where ambiguous visual interpretations cascade through text generation, resulting in overconfident but incorrect descriptions. To address these MLLM specific failure modes, we introduce 12 novel features:

Layer Consistency Features, LCF (63-64): Detect *Cross-layer inconsistency* by measuring the alignment between early and late text decoder layers. When model hallucinate, there tends to be greater disagreement between what early layers represent vs what late layers represent for same visual-linguistic input.

$$c = \left[\text{cosine\_sim}(\mathbf{h}^{(early)}, \mathbf{h}^{(late)}) + 1\right]/2, \quad \mathcal{F}_{\text{consistency}} = [c, 1 - c] \tag{1}$$

where $\mathbf{h}^{(early)}$ and $\mathbf{h}^{(late)}$ are flattened hidden states from specific text decoder layers. The early layer index is $\min(\text{text\_start} + 5, \text{max\_layer} - 10)$ (5th text decoder layer or 10 layers before the end), and the late layer index is $\text{max\_layer} - 2$ (2nd to last layer). We normalize the cosine similarity from [-1,1] to [0,1] range and extract both consistency (c) and inconsistency ($1 - c$) as complementary features to provide the classifier with dual perspectives on layer
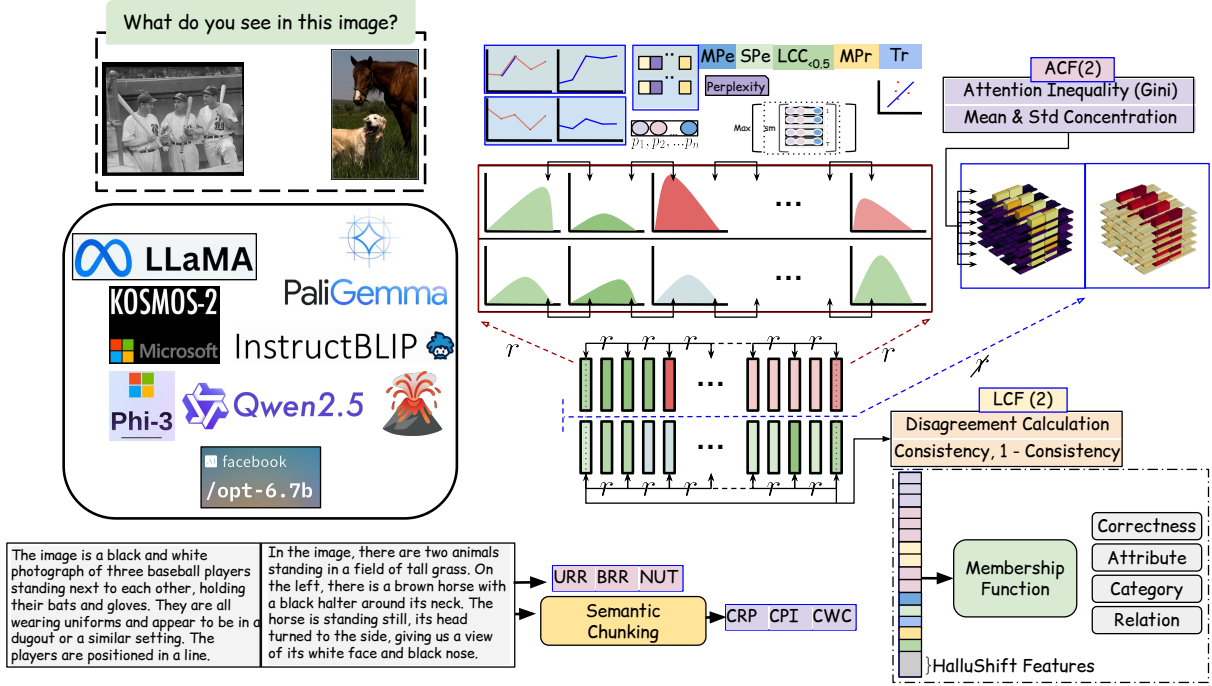
Figure 2: Illustration of our proposed HALLUSHIFT++ framework. First, we extract features from the internal states of MLLMs including: (1) original HALLUSHIFT features (1-62), and (2) novel HALLUSHIFT++ features (63-74, highlighted in blue boxes) comprising 2 attention concentration features (ACF) measuring focus dispersion via Gini coefficients, 2 layer consistency features (LCF) detecting cross-layer disagreement, and perplexity/confidence features capturing uncertainty patterns. Color transitions from green to red in internal distributions indicate hallucination-correlated patterns. Second, semantic chunking decomposes generated text into object, attribute, and relation components. Finally, our membership function processes the combined 74-dimensional feature vectors to assign hierarchical hallucination scores across four categories (Correctness, Attribute, Category, Relation) rather than binary classification.

alignment for optimal decision boundaries.

Attention Concentration Features, ACF (64-65): We address *Attention dispersion* by measuring attention inequality using Gini coefficients, higher values indicate desirable concentrated attention on specific visual regions, while lower values signal problematic scattered attention associated with hallucination. For each attention layer, we compute the Gini coefficient as:

$$G_l = \left[2 \sum_{i=1}^{n} \sum_{j=1}^{i} s_j \right] / \left[n \sum_{i=1}^{n} s_i \right] - 1, \quad (2)$$

$$\mathcal{F}_{\text{concentration}} = \left[\text{mean}(\{|G_l|\}_{l=-3}^{-1}), \text{std}(\{|G_l|\}_{l=-3}^{-1}) \right] \quad (3)$$

where s_i and s_j both refer to elements of the same sorted flattened attention weights for layer l, n is the total number of attention weights, and G_l is the Gini coefficient measuring concentration. When attention is properly focused (grounded generation), most weights concentrate on few relevant regions yielding high Gini values ($G_l \rightarrow 1$). When attention disperses across many irrelevant regions (hallucination-prone), weights distribute more uniformly yielding low Gini values ($G_l \rightarrow 0$). We analyze the last 3 attention layers and compute mean concentration level and consistency

deviation (std) to detect this hallucination-predictive attention dispersion pattern.

Perplexity and Confidence Features (67-71): Address *Confidence degradation* by detecting when models generate text with deteriorating certainty, indicating potential hallucination due to weak visual grounding. For each generated token position $t \in \{1, 2, \ldots, T\}$, we extract logits l_t containing raw scores for the entire vocabulary. We convert this to probabilities and extract the maximum probability.

$$\mathbf{p}_t = \text{softmax}(l_t[0]), \quad p_{\text{max}}^{(t)} = \max(\mathbf{p}_t) \quad (4)$$

where, $l_t[0]$ selects the first batch dimension, and $p_{\text{max}}^{(t)}$ represents the model's confidence in the chosen token.

Perplexity is computed as the reciprocal of maximum probability,

$$\text{ppl}_t = 1/p_{\text{max}}^{(t)} \quad (5)$$

After processing all T token positions, we construct two lists, $\{p_{\text{max}}^{(1)}, \ldots, p_{\text{max}}^{(T)}\}$ (maximum probabilities) and $\{\text{ppl}_1, \ldots, \text{ppl}_T\}$ (perplexities). From these, we extract 5 features:

Mean Perplexity (MPe): Captures the average model uncertainty across all generated tokens. Higher values indicate lower overall

confidence in the generated content.

$$f_1 = \frac{1}{T} \sum_{t=1}^{T} \text{ppl}_t \qquad (6)$$

Perplexity Standard Deviation (SPe): Measures the consistency of model confidence throughout generation. High variance suggests unstable confidence patterns associated with hallucination.

$$f_2 = \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} (\text{ppl}_t - f_1)^2} \qquad (7)$$

Confidence Trend (Tr): Uses linear regression to detect whether model confidence increases or decreases over the sequence. Negative slopes indicate deteriorating confidence, signaling potential hallucination.

$$f_3 = \frac{T \sum_{t=1}^{T} (t-1) \cdot p_{\max}^{(t)} - \sum_{t=1}^{T} (t-1) \sum_{t=1}^{T} p_{\max}^{(t)}}{T \sum_{t=1}^{T} (t-1)^2 - \left(\sum_{t=1}^{T} (t-1) \right)^2} \qquad (8)$$

Mean Confidence (MPr): Represents the average maximum probability across all token positions, providing a direct measure of model certainty.

$$f_4 = \frac{1}{T} \sum_{t=1}^{T} p_{\max}^{(t)} \qquad (9)$$

Low-Confidence Token Fraction (LCC): Counts the proportion of tokens with maximum probability $p_{\max}^{(t)}$ below 0.5, indicating systematic uncertainty in generation.

$$f_5 = \frac{|\{t : p_{\max}^{(t)} < 0.5\}|}{T} \qquad (10)$$

These five features collectively capture different aspects of confidence degradation that correlate with hallucination in MLLMs, where weak visual grounding leads to uncertain text generation patterns.

Token Pattern Features (72-74): 3 token pattern features are constructed to detect when model generate text that lacks visual grounding by analyzing repetition and diversity patterns that differ from naturally grounded descriptions.

$$URR = 1 - |U|/|T|, \quad BRR = 1 - |B_u|/|B|, \quad NUT = |U|/|T| \quad (11)$$

where, URR represents unique repetition ratio, BRR bi-gram repetition ratio, NUT normalized unique tokens, U is the set of unique tokens, T is all tokens, B_u is unique bigrams, and B is all bigrams in the generated text. Visually grounded descriptions exhibit natural diversity and varied phrasing, while hallucinated content shows repetitive patterns as models use memorized language patterns. High repetition ratios ($URR, BRR \rightarrow 1$) and low diversity ($NUT \rightarrow 0$) indicate over-reliance on language priors rather than visual evidence.

4.2 Semantic Chunking Strategy

A fundamental limitation of the original HALLUSHIFT approach lies in its treatment of generated text as monolithic units, assigning binary hallucination labels to entire responses, which fails at places where hallucinations manifest at different semantic granularities within the same response. This necessitates a semantic decomposition that can isolate and evaluate individual units independently.

Algorithm 1: Semantic Chunk Extraction

Input: Text string T

1   $D \leftarrow \text{NLP}(T);$ $C \leftarrow \emptyset;$
2   **foreach** $t \in D$ **do**
3   **if** t.pos $\in \{\text{NOUN}, \text{PROPN}\}$ **and** t *not stopword* **then**
4   Add object chunk for t to C;
5   **if** t.pos $= \text{ADJ}$ **and** t.head.pos $\in \{\text{NOUN}, \text{PROPN}\}$ **then**
6   Add attribute chunk for $(t, t.\text{head})$ to C;
7   **if** t.dep $\in \{\text{prep}, \text{agent}\}$ **or** t.pos $= \text{VERB}$ **then**
8      $O \leftarrow$ nouns among t's children;
9   **if** $|O| \geq 2$ **then**
10   Add relation chunk for $(O[0], t, O[1])$ to C;

11   Remove duplicate/short chunks from C; sort by start index;
12   **return** C

To address this, we implement a semantic decomposition approach, adapting [25] for Vision-Language tasks. Specifically, we extract three types of chunks from generated text: *object mentions* (individual nouns and proper nouns representing entities in the description), *attribute-object pairs* (adjective-noun combinations capturing descriptive properties) and relationship patterns (preposition/verb based connections between multiple objects that encode spatial or action-based relations). Each extracted chunk is then characterized by three core positional and structural features: Chunk Word Count (CWC: number of tokens in the chunk), Chunks Per Image (CPI: the total chunk count k for that image), and Chunk Relative Position (CRP: computed as i/k, where i is the position of the current chunk) capturing semantic density, descriptive granularity, and temporal ordering respectively.

4.3 Hierarchical Ground Truth Matching

The critical challenge for training data preparation becomes determining whether each chunk represents factual content grounded in the image or hallucinated content to generate supervised learning labels. Traditional approaches such as CHAIR [6] rely on simple keyword matching against object vocabularies, that fails to capture the nuanced relationships between objects, their properties, and spatial arrangements. Our hierarchical ground truth matching system addresses this limitation by implementing a verification process that respects the inherent hierarchy, if an object doesn't exist in the image (category error), then discussing its attributes or relationships is meaningless. Conversely, if an object exists but has wrong attributes, the spatial relationships might still be partially correct. This motivates our three-stage verification process.

Ground Truth Extraction Functions: We define three extraction functions that parse ground truth annotations $\mathcal{G}$ to create searchable representations. Firstly, the object extraction function, processes caption to identify all mentioned entities using named entity recognition [16] and noun phrase extraction, returning a set of canonical object labels {person, car, ...}. Secondly, the attribute extraction function considers object labels to analyze dependency relationships for object-attribute pairs, creating tuples {(car, red), (building, tall), ...} where each attribute is linked to its

corresponding object. Thirdly, the relation extraction function identifies spatial and action relationships through prepositional phrase analysis and verb-argument structures, generating relational triplets {(car, parked-next-to, building), …}. To delineate this, we utilize predefined lexical dictionaries including sequence of objects, attribute categories (i.e., color, size, shape, condition, material) and spatial-action relations.

Chunk Analysis Functions: After processing ground truths, we extract list of generated chunks **c** from generated text **y**. To expedite ground truth matching, we define three chunk analysis functions that extract comparable information from generated chunks. Here, the object function identifies the primary entity within each generated chunk c_i using the same NLP techniques applied to ground truth. Similarly, the attribute function forms attribute claims from the chunk, while relation function identifies relational assertions made within the chunk.

This hierarchical approach ensures that category-level errors (non-existent objects) take precedence over attribute-level errors (incorrect properties), which in turn take precedence over relation-level errors (incorrect spatial/action relationships).

4.4 Membership Function

Following HalluShift [12] membership function approach, we extend the framework to handle multiple hallucination types. In contrast to the original HalluShift that performs binary classification (hallucinated vs. truthful), HalluShift++ implements a membership function that assigns probabilistic scores across four classes: $\mathcal{Y}$ = {Correct, Category, Attribute, Relation}. We combine the 74-dimensional, HalluShift++ features with three additional chunk-specific contextual features $\phi(c_i) \in \mathbb{R}^3$ (CWC, CRP, CPI), where $\mu(c_i, y_j) = f_\theta([\mathcal{F}_{\text{HalluShift++}}(c_i) \in \mathbb{R}^{74}; \phi(c_i) \in \mathbb{R}^3])_j$ employs attention mechanisms on these 3 chunk features to weigh feature importance dynamically across different hallucination types.

5 Experimental Analysis

5.1 Dataset Overview

We evaluate HalluShift++ on two prominent vision-language benchmarks: MS-COCO [20] and LLaVA [22]. MS-COCO provides a diverse collection of natural images with detailed captions, enabling comprehensive evaluation of object recognition, attribute assignment, and spatial relationship understanding. Contrastingly, LLaVA benchmark focuses on instruction-following capabilities in vision-language tasks, providing a challenging testbed for hallucination detection in conversational contexts.

5.2 Experimental Setup

We engage eight state-of-the-art MLLMs with several architectural paradigms: Llama-3.2-11B-Vision-Instruct and Llama-3.2-11B-Vision [29], PaliGemma-3B-Mix-224 [5], Kosmos-2-Patch14-224 [24], InstructBLIP-Vicuna-7B [11], Phi-3.5-Vision-Instruct [1], Qwen 2.5-VL-3B-Instruct [3], and LLaVA-1.5-7B-HF [22]. For consistent evaluation, we generate model responses using greedy decoding with a maximum output length of 64 tokens. Ground truth annotations are established through careful manual verification against dataset annotations, with hallucination labels assigned according

to our hierarchical taxonomy: Category (non-existent objects), Attribute (incorrect properties), Relation (spatial/action inconsistencies), and Correct (factually accurate). We extract both HalluShift and HalluShift++ features from internal layers of the model with range 2. We employ a three-layer neural architecture with progressive feature integration as the membership estimator. We use AdamW optimizer with an initial learning rate of 1e-4 and cosine annealing schedule with batch size 32.

Training employs early stopping based on validation AUC-ROC with patience of 10 epochs. We apply class balancing techniques like SMOTE [8] and class weighting to handle potential dataset imbalances. SMOTE generates synthetic samples for minority classes, while enhanced class weighting applies higher penalties to misclassifications of rare hallucination types, particularly addressing the severe imbalance in attribute hallucinations which constitute less than 5% of samples. All experiments are conducted on NVIDIA A40 GPU with 48 GB VRAM using mixed-precision training for computational efficiency.

Algorithm 2: Ground Truth-Aware Chunk Classification

Input: Chunk c, Ground Truth Set $\mathcal{G}$
Output: Label $\mathcal{L} \in$ {CORRECT, CATEGORY_HALLUC, ATTRIBUTE_HALLUC, RELATION_HALLUC}

1 Extract $O \leftarrow$ objects from $\mathcal{G}$;
2 Extract $\mathcal{A} \leftarrow$ attributes from $\mathcal{G}$;
3 Extract $\mathcal{R} \leftarrow$ relations from $\mathcal{G}$;
4 **if** c.type = object **then**
5   **if** c.text $\notin O$ *(match)* **then**
6   **return** CATEGORY_HALLUC
7 **else**
8   Let $objs \leftarrow$ associated objects in c (1 or 2);
9   **if** *any* $o \in objs$ *not in* O **then**
10   **return** CATEGORY_HALLUC
11   **if** c.type = attribute **then**
12   **if** c.attribute $\notin \mathcal{A}$ *(match/sim)* **then**
13   **return** ATTRIBUTE_HALLUC
14   **else**
15   **if** c.relation $\notin \mathcal{R}$ *(match/sim)* **then**
16   **return** RELATION_HALLUC

17 **return** CORRECT

---

6 Comparative Analysis

6.1 Performance of Vision-Language Tasks

Table 1 presents a comprehensive comparison of HalluShift++ against the original HalluShift method across multiple MLLMs and datasets. We didn't compare with external evaluation metrics such as POPE [19] and FAITHScore [15] because our study focuses exclusively on methods that leverage internal model representations. Our results reveal several critical insights about hallucination detection in MLLMs. HalluShift++ achieves consistent improvements of 27.7 - 64.1% AUC-ROC gains across all tested models,

Table 1: Comparative analysis of hallucination detection methods. Metrics reported include AUC-ROC, precision, and f1-score.

Model	Dataset	Method	AUC_ROC (%)	Precision (%)	F1-Score (%)
Llama-3.2-11B-Vision-Instruct	mscoco	ITI	49.80	48.30	57.20
		HALLUSHIFT	52.10	52.10	60.50
		HALLUSHIFT++	**86.10**	**77.40**	**62.60**
	llava	ITI	48.20	31.50	40.10
		HALLUSHIFT	51.60	34.30	43.30
		HALLUSHIFT++	**82.80**	**67.90**	**59.60**
paligemma-3b-mix-224	mscoco	ITI	58.40	62.10	61.50
		HALLUSHIFT	61.70	65.40	64.20
		HALLUSHIFT++	**87.00**	**77.50**	**69.00**
	llava	ITI	57.90	54.30	50.20
		HALLUSHIFT	61.70	57.20	53.00
		HALLUSHIFT++	**83.80**	**70.60**	**61.80**
kosmos-2-patch14-224	mscoco	ITI	50.30	40.80	38.20
		HALLUSHIFT	54.00	43.70	41.00
		HALLUSHIFT++	**85.40**	**67.20**	**57.60**
	llava	ITI	51.80	40.20	39.50
		HALLUSHIFT	55.40	43.40	42.30
		HALLUSHIFT++	**85.90**	**68.30**	**63.40**
instructblip-vicuna-7b	mscoco	HALLUSHIFT	51.70	34.30	43.30
		HALLUSHIFT++	**85.00**	**67.70**	**56.40**
	llava	HALLUSHIFT	51.50	25.00	33.30
		HALLUSHIFT++	**84.80**	**67.60**	**61.50**
Phi-3.5-vision-instruct	mscoco	HALLUSHIFT	54.90	45.90	54.30
		HALLUSHIFT++	**85.30**	**71.40**	**56.90**
	llava	HALLUSHIFT	58.40	47.50	49.50
		HALLUSHIFT++	**84.30**	**64.80**	**58.30**
Qwen2.5-VL-3B-Instruct	mscoco	HALLUSHIFT	55.90	61.10	56.20
		HALLUSHIFT++	**83.60**	**72.90**	**58.50**
	llava	HALLUSHIFT	55.50	51.70	44.80
		HALLUSHIFT++	**84.00**	**66.80**	**59.60**
Llama-3.2-11B-Vision	mscoco	HALLUSHIFT	65.80	52.50	60.90
		HALLUSHIFT++	**91.70**	**83.00**	**71.80**
	llava	HALLUSHIFT	71.10	53.20	61.60
		HALLUSHIFT++	**88.20**	**76.40**	52.20
llava-1.5-7b-hf	mscoco	HALLUSHIFT	53.10	42.10	43.10
		HALLUSHIFT++	**84.70**	**67.20**	**58.40**
	llava	HALLUSHIFT	58.80	74.00	79.50
		HALLUSHIFT++	**96.50**	**94.80**	**88.60**

indicating that our multimodal specific features capture fundamental hallucination patterns independent of architectural design. This consistency validates our hypothesis that cross-layer inconsistency, attention dispersion, and confidence degradation represent universal failure modes in multimodal systems.

A particularly compelling aspect of our results is the scale agnostic effectiveness of HALLUSHIFT++. Even a very small model like Kosmos-2-Patch14-224 [24] with only 1.66B params show ∼ 58% performance gain in AUC-ROC, while models with 7B params like LLaVA-1.5-7B-HF [22] and InstructBLIP-Vicuna-7B [11] have
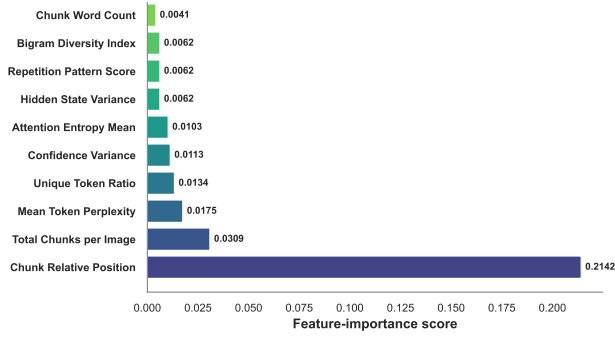
Figure 3: Feature importance analysis showing top 10 features for the best performing model (Llama-3.2-11B-Vision). The horizontal bar chart shows the contribution of each feature to hallucination detection accuracy, with semantic context features dominating the rankings while HalluShift++ features provide significant improvements. Feature names are interpretable rather than numerical for clarity.

shown a significant improvement of $\sim 64\%$ than base HalluShift counterpart. These smaller models demonstrate that effective hallucination detection does not require massive computation resources, making our approach practical for resource constrained environments. Simultaneously, larger models with 11B params like Llama-3.2-11B-Vision [29] achieve 91.7% AUC-ROC, showcasing the effectiveness of HalluShift++ across the model parameter spectrum.

Table 2: Performance comparison between HalluShift [12], HalluShift++, HaloScope [13], and CCS* [7] on LLMs for question-answering tasks.

Model	Dataset	Method	AUC_ROC
OPT-6.7B	TruthfulQA	CCS*	63.91
		HaloScope	73.17
		HalluShift	89.91
		HalluShift++	**92.68**
	TyDiQA-GP	CCS*	64.62
		HaloScope	80.98
		HalluShift	85.11
		HalluShift++	**87.58**
LLaMA-2-7B	TruthfulQA	CCS*	67.95
		HaloScope	78.64
		HalluShift	89.93
		HalluShift++	**90.82**
	TyDiQA-GP	CCS*	80.38
		HaloScope	**94.04**
		HalluShift	87.61
		HalluShift++	89.66

6.2 Performance on Language Model QA Tasks

To demonstrate the generalizability of HalluShift++ beyond vision-language tasks, we evaluate our approach on text-only question-answering scenarios using large language models. Table 2 presents comparative results on TruthfulQA and TyDiQA datasets using

Table 3: Semantic chunking strategy effectiveness showing performance gains from hierarchical text decomposition.

Chunking Strategy	Avg. Chunks/Image	AUC-ROC
No Chunking	1.0	0.612
Sentence-level	2.3	0.698
Object-only	8.7	0.743
Object + Attribute	12.1	0.798
Complete Semantic	14.2	0.847

OPT-6.7B and LLaMA-2-7B models.

The results demonstrate that HalluShift++ maintains its effectiveness in text-only settings, achieving consistent improvements over the baseline HalluShift method. The most substantial improvement is observed for OPT-6.7B on TruthfulQA (2.77 point improvement), while more modest but consistent gains are achieved across other configurations. These results validate that our enhanced feature extraction methodology generalizes effectively beyond multimodal scenarios, suggesting broader applicability for hallucination detection in diverse language model architectures and task domains.

7 Ablation Study

7.1 Feature Importance Analysis

We utilize best performing model, Llama-3.2-11B-Vision, for further analysis of feature importance, providing interpretable feature names and their contributions to hallucination detection (Fig 3) Our detailed analysis reveals that *Chunk Relative Position* emerges as the most essential feature with an importance score of +0.2142, indicating that the location of content within generated responses is highly predictive of hallucination. This finding suggests that hallucinated content tends to appear in specific positions within model outputs, potentially reflecting patterns in the generation process. The second most dominant feature, *Total Chunks per Image* (+0.0309), captures the complexity of generated descriptions. Models generating overly detailed descriptions may be more prone to hallucination, as they attempt to provide information beyond what can be reliably extracted from the visual input. HalluShift++ effectively captures this phenomenon through the attention concentration features detecting when model lose focus across too many visual regions while attempting detailed descriptions. Additionally, the confidence degradation features identify increasing uncertainty as models venture beyond reliable visual evidence. *Mean Token Perplexity* (+0.0175) shows strong contribution cause models exhibit higher uncertainty when generating content not grounded in visual evidence compared to describing what they actually observe in images. Fruthermore, the *Unique Token Ratio* (+0.0134) proves effective as models exhibit repetitive patterns based memorized language prior from training data, when lacking visual grounding.

7.2 Semantic Chunking Effectiveness

We further evaluate the impact of our semantic chunking strategy on detection performance, demonstrating the value of semantic decomposition in Table 3.

8 Conclusion

In this work, we introduced HALLUSHIFT++, a framework that advances hallucination detection from traditional binary classification to hierarchical understanding of multimodal hallucination patterns. By extending the distributional shift analysis from text-only LLMs to the complex visual-linguistic landscape of MLLMs, we demonstrated that hallucinations manifest as quantifiable perturbations across multiple dimensions: cross-layer inconsistency, attention dispersion, and confidence degradation, phenomena that remain invisible to conventional approaches.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, and Ahmed Awadallah et al. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219 [cs.CL] https://arxiv.org/abs/2404.14219

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[3] Jinze Bai, Shuai Bai, Yunfei Chu, and Zeyu Cui. 2023. Qwen Technical Report. arXiv:2309.16609 [cs.CL] https://arxiv.org/abs/2309.16609

[4] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930* (2024).

[5] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. 2024. PaliGemma: A versatile 3B VLM for transfer. arXiv:2407.07726 [cs.CV] https://arxiv.org/abs/2407.07726

[6] Florian Bordes, Richard Yuanzhe Pang, and Anurag Ajay. 2024. An Introduction to Vision-Language Modeling. arXiv:2405.17247 [cs.LG] https://arxiv.org/abs/2405.17247

[7] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827* (2022).

[8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[9] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744* (2024).

[10] Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024. Unified hallucination detection for multimodal large language models. *arXiv preprint arXiv:2402.03190* (2024).

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. arXiv:2305.06500 [cs.CV] https://arxiv.org/abs/2305.06500

[12] Sharanya Dasgupta, Sujoy Nath, Arkaprabha Basu, Pourya Shamsolmoali, and Swagatam Das. 2025. Hallushift: Measuring distribution shifts towards hallucination detection in llms. *arXiv preprint arXiv:2504.09482* (2025).

[13] Xuefeng Du, Chaowei Xiao, and Sharon Li. 2024. Haloscope: Harnessing unlabeled llm generations for hallucination detection. *Advances in Neural Information Processing Systems* 37 (2024), 102948–102972.

[14] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys* 55, 12 (2023), 1–38.

[15] Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. 2023. FaithScore: Fine-grained Evaluations of Hallucinations in Large Vision-Language Models. *arXiv preprint arXiv:2311.01477* (2023).

[16] Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. Recent Advances in Named Entity Recognition: A Comprehensive Survey and Comparative Study. arXiv:2401.10825 [cs.CL] https://arxiv.org/abs/2401.10825

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[18] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems* 36 (2023), 41451–41530.

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

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs.CV] https://arxiv.org/abs/1405.0312

[21] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2023. Mitigating hallucination in large multi-modal models via robust instruction tuning. *arXiv preprint arXiv:2306.14565* (2023).

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. arXiv:2304.08485 [cs.CV] https://arxiv.org/abs/2304.08485

[23] Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253* (2024).

[24] Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. 2024. Kosmos-G: Generating Images in Context with Multimodal Large Language Models. arXiv:2310.02992 [cs.CV] https://arxiv.org/abs/2310.02992

[25] Eunkyu Park, Minyeong Kim, and Gunhee Kim. 2025. HalLoc: Token-level Localization of Hallucinations for Vision Language Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 29893–29903.

[26] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156* (2018).

[27] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2016. Movie Description. arXiv:1605.03705 [cs.CV] https://arxiv.org/abs/1605.03705

[28] Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. Llm-check: Investigating detection of hallucinations in large language models. *Advances in Neural Information Processing Systems* 37 (2024), 34188–34216.

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL] https://arxiv.org/abs/2302.13971

[30] Junyang Wang, Yiyang Zhou, Guohai Xu, Pengcheng Shi, Chenlin Zhao, Haiyang Xu, Qinghao Ye, Ming Yan, Ji Zhang, Jihua Zhu, et al. 2023. Evaluation and analysis of hallucination in large vision-language models. *arXiv preprint arXiv:2308.15126* (2023).

[31] Mingrui Wu, Jiayi Ji, Oucheng Huang, Jiale Li, Yuhang Wu, Xiaoshuai Sun, and Rongrong Ji. 2024. Evaluating and analyzing relationship hallucinations in large vision-language models. *arXiv preprint arXiv:2406.16449* (2024).