Diving into Mitigating Hallucinations from a Vision Perspective for Large Vision-Language Models

Anonymous ACL submission

Abstract

Object hallucination in Large Vision-Language Models (LVLMs) significantly impedes their real-world applicability. As the primary component for accurately interpreting visual information, the choice of visual encoder is pivotal. We hypothesize that the diverse training paradigms employed by different visual encoders instill them with distinct inductive biases, which leads to their diverse hallucination performances. Existing benchmarks typically focus on coarse-grained hallucination detection and fail to capture the diverse hallucinations elaborated in our hypothesis. To systematically analyze these effects, we introduce VHBench-10, a comprehensive benchmark with approximately 10,000 samples for evaluating LVLMs across ten fine-grained hallucination categories. Our evaluations confirm encoders exhibit unique hallucination characteristics. Building on these insights and the suboptimality of simple feature fusion, we propose VisionWeaver, a novel Context-Aware Routing Network. It employs global visual features to generate routing signals, dynamically aggregating visual features from multiple specialized experts. Comprehensive experiments confirm the effectiveness of VisionWeaver in significantly reducing hallucinations and improving overall model performance.

1 Introduction

Large Vision-Language Models (LVLMs), such as GPT-4V (Achiam et al., 2023) and LLaVA (Liu et al., 2024c), demonstrate remarkable abilities to understand (Hao et al., 2023; Kojima et al., 2022) and generate (Lian et al., 2023; Zhou et al., 2023) content from visual inputs. Despite these strengths, the models frequently exhibit object hallucinations—describing objects or attributes not present in the provided images. This tendency critically undermines their reliability and applicability in real-world scenarios (Mai et al., 2023; Tang et al., 2024; Zhou et al., 2023; Huang et al., 2024; Liu et al., 2023).

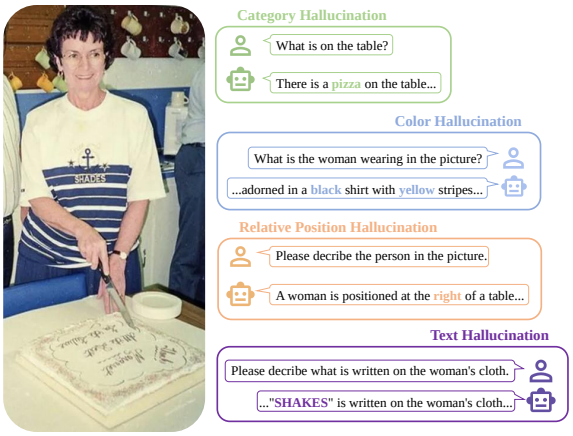

Figure 1: Examples of common hallucinations produced by LVLMs, such as misidentifying object categories, incorrect color descriptions, erroneous relative positioning, and inaccurate text recognition. It represent the types of fine-grained visual errors that our vision-centric VHBench-10 benchmark is designed to evaluate across categories like detection, segmentation, localization, and classification.

The choice of visual encoder within LVLMs is critical. This selection directly influences the capacity of model for accurate visual interpretation, which consequently affects its propensity to generate hallucinations. Furthermore, variations in training paradigms and architectural designs mean that different visual encoders introduce distinct biases and capabilities into LVLMs. These differences subsequently lead to diverse hallucination patterns observed in downstream tasks. For example, the widely adopted CLIP (Radford et al., 2021) visual encoder excels at vision-text alignment, largely due to its pre-training on extensive image-text datasets. However, it is less effective at capturing fine-grained visual details when compared to vision-focused models such as DINOv2 (Oquab et al., 2023).

To systematically investigate how different visual encoders influence hallucination behaviors in LVLMs, a more nuanced understanding of hallucination types is necessary. Existing benchmarks, such as POPE (Li et al., 2023), primarily assess object hallucinations. Their evaluation typically focuses on whether models generate descriptions of non-existent objects. While this methodology is valuable, it treats all hallucinations uniformly. This overlooks the possibility that such errors may stem from failures in distinct visual sub-tasks. These sub-tasks include object detection, segmentation, localization, or classification, each demanding unique visual perception capabilities. Deficiencies in any such capability can, in turn, lead to specific types of hallucinations.

To address this issue, we propose VHBench-10, a comprehensive vision-centric hallucination benchmark designed to evaluate LVLMs across ten fine-grained hallucination categories. These categories are systematically grouped into four main types: detection, segmentation, localization, and classification. VHBench-10 consists of approximately 10,000 samples, with each sample including an image, a detailed factual description, and a corresponding description that contains a specific hallucination. By measuring the likelihood of LVLMs generating hallucinated versus factual descriptions, VHBench-10 facilitates a precise diagnosis of deficiencies in visual perception capabilities and offers targeted insights for model refinement.

Based on evaluations conducted on VHBench-10, we observe that the choice of visual encoder significantly influences hallucination behavior. For instance, an LVLM employing the Vary visual encoder, which is specifically pre-trained on text recognition tasks, illustrates this. Such a model exhibits markedly lower hallucination rates in text-related visual tasks compared to its performance in other task domains.

Based on these findings, a natural question arises: *Can integrating diverse visual encoders help LVLMs reduce hallucinations across tasks and lower overall hallucination propensity?* However, our experiments (detailed in Sec 5.3) revealed that simple feature fusion techniques for visual encoders (e.g., feature addition or feature concatenation (Tong et al., 2024)) often yield suboptimal performance compared to using these encoders individually. To address this challenge, we introduce VisionWeaver, a Context-Aware Routing Network. Guided by the LVLM's global visual understanding, this network dynamically aggregates visual features from multiple specialized encoders. Specifically, our proposed adaptive routing module utilizes the `[CLS]` token feature from CLIP as a primary input. This feature, which encapsulates global image context and key visual information, is then processed by the module and transformed into routing signals for the specialized visual encoders.

Comprehensive experiments conducted on both established hallucination benchmarks (such as POPE (Li et al., 2023), AutoHallusion (Wu et al., 2024), and our VHBench-10) and general LVLM benchmarks demonstrate that VisionWeaver effectively reduces hallucinations while concurrently enhancing overall performance.

2 Related Work

2.1 Benchmarks for Hallucinations

In the scope of LVLMs, hallucinations is considered to generating incorrect or misleading text, which do not match the content for the given image. Numerous benchmarks evaluate hallucinations in LVLMs. For instance, POPE (Li et al., 2023) assesses object existence, often via polling-based queries. HallusionBench (Guan et al., 2024) probes entangled language/visual illusions and event understanding. AMBER (Wang et al., 2024b) offers an LLM-free, multi-dimensional evaluation of existence, attribute, and relation hallucinations. While these benchmarks effectively identify various hallucination types, they often categorize errors broadly (e.g., general attribute errors) without pinpointing why these occur in terms of specific visual cognitive failures. This makes it difficult to diagnose the precise visual processing weaknesses. VHBench-10 addresses this gap by grounding its taxonomy in classical vision tasks (color, shape, counting, position), enabling a fine-grained diagnosis of which specific visual perceptual abilities are deficient and contribute to hallucinations.

2.2 Mitigating Hallucinations

Multiple solutions have been proposed recently to address hallucinations. (Hu et al., 2023; You et al., 2023) try to solve the problem from the aspect of data bias, by constructing better-grounded annotated training data. There is also several works (Wang et al., 2024a; Leng et al., 2023) starting with decoding strategies for LVLMs. (Jain et al., 2024; Chen et al., 2024b) are introduced to improve their overall performance by enhancing the perception

ability of MLLMs. The closest work related to ours is (He et al., 2024), with the help of multi-task vision experts, they try to provide a more comprehensive and accurate summarization of visual inputs. Different from (He et al., 2024), we use a context-aware routing mechanism to choose the task-specific knowledge from the pool, which can preserve better performance compared with a fix visual inputs.

3 The VHBench-10 Benchmark

3.1 Vision-Centric Taxonomy

VHBench-10 is constructed based on critical observations of current methodologies. Existing hallucination taxonomy approaches (Wang et al., 2023; Liu et al., 2024a) and benchmarks (Liu et al., 2024e), while valuable, primarily address coarse-grained object existence or general inconsistencies. Existing evaluation protocols often fall short in capturing the subtleties of fine-grained visual hallucinations, such as minor attribute inaccuracies or misestimated spatial relations. Furthermore, they lack the diagnostic granularity to link these errors to specific deficiencies in underlying visual perceptual abilities. For instance, benchmarks such as POPE (Li et al., 2023) can effectively evaluate coarse-grained object existence using polling-based yes/no questions, but they inherently lack the granularity to diagnose more subtle, fine-grained visual errors. To address this methodological gap, we introduce VHBench-10. This comprehensive benchmark is specifically designed to disentangle and evaluate hallucinations in LVLMs. By centering the analysis on core visual competencies, VHBench-10 facilitates a more structured assessment of the origins and nature of hallucinations.

The core idea behind VHBench-10 is that visual hallucinations in LVLMs frequently arise from shortcomings in specific underlying visual processing sub-tasks. To enable a more insightful analysis beyond a uniform treatment of hallucinations, we introduce a hierarchical taxonomy of visual understanding. This taxonomy focuses on four visual competencies deemed fundamental to image understanding: detection, segmentation, localization, and classification. We concentrate on these four because an analysis of mainstream vision benchmarks shows that tasks in these areas represent 81%(Meta, 2025) of dataset annotations. Consequently, they form the foundational basis for the majority of contemporary vision applications.

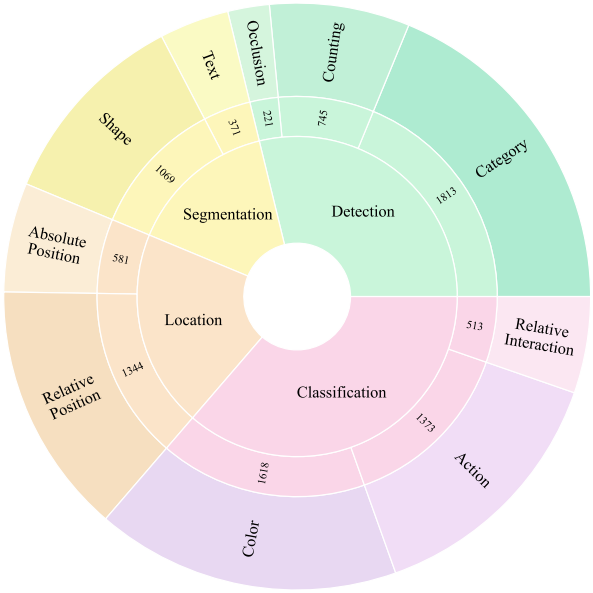

Figure 2: We classify hallucinations into four major categories, which are further subdivided into 10 fine-grained sub-tasks. The corresponding number for each sub-task represents the sample count in our VHBench-10 benchmark.

3.2 Data Construction

Guided by the vision-centric taxonomy previously detailed, our analysis of visual understanding failures resulted in defining ten distinct, fine-grained hallucination sub-categories. These sub-categories are systematically derived from the four core visual competencies: detection, segmentation, localization, and classification. VHBench-10 is meticulously structured around these ten sub-categories, offering a framework for evaluating LVLM performance. Appendix A details these ten sub-categories. Each category is designed to investigate specific aspects of visual perception, allowing for a granular diagnosis of an LVLM's weaknesses in visual understanding. For example, classification-related errors can include misidentifying object attributes (like color or material) or misclassifying an object entirely. Detection-related hallucinations might involve asserting the presence of non-existent objects. Localization errors can pertain to incorrect spatial relationships, and segmentation issues may involve misinterpreting object boundaries. By evaluating performance across these distinct categories, VHBench-10 helps shift the focus from merely identifying hallucinations to pinpointing the underlying visual perceptual failures.

Following the detailed description of the structure of VHBench-10 and task categories, we now

present its design principles and data curation methodology. The central goal was to produce targeted evaluation samples for each specific sub-category. This process involved several meticulous steps:

1. **Image and Factual Caption Selection**: We begin by carefully selecting 2,000 images from the LLaVA-ReCap-118K dataset. Each chosen image was accompanied by a detailed and factually accurate caption, serving as the ground truth (R) for the visual content.

2. **Targeted Hallucination Generation**: For each selected image and its factual caption, we leveraged the GPT-4(Achiam et al., 2023) to generate a corresponding hallucinated caption (H). Details of the instructions can be found in Appendix B. Crucially, each generated hallucination was specifically crafted to align with one of the ten pre-defined sub-categories detailed in section 3.2, thereby ensuring that each sample in VHBench-10 probes a particular type of visual misinterpretation. This process resulted in 9,648 unique instances.

3. **Dataset Structure**: Each sample in VHBench-10 is formulated as a ternary (I, R, H), where I represents the image, R is the real, factual caption, and H is the caption containing a specific, deliberately injected hallucination tied to one of our defined sub-categories. This structure facilitates a direct comparison of an LVLM's propensity to endorse factual versus hallucinated descriptions.

3.3 Evaluation and Analysis

To validate the utility of VHBench-10 and investigate the impact of different visual encoders on hallucination patterns, we evaluated several LVLMs equipped with various vision experts, establishing initial baselines. Specifically, we input image with real caption ($I + R$) and image with hallucinated caption ($I + H$) into LVLM respectively, and calculate the probability of generating these two combinations through perplexity (ppl). A model is considered to have made an error on a VHBench-10 sample if it deems the hallucinated caption (H) more probable than the factual caption (R). The complete evaluation process can be found in Appendix C.

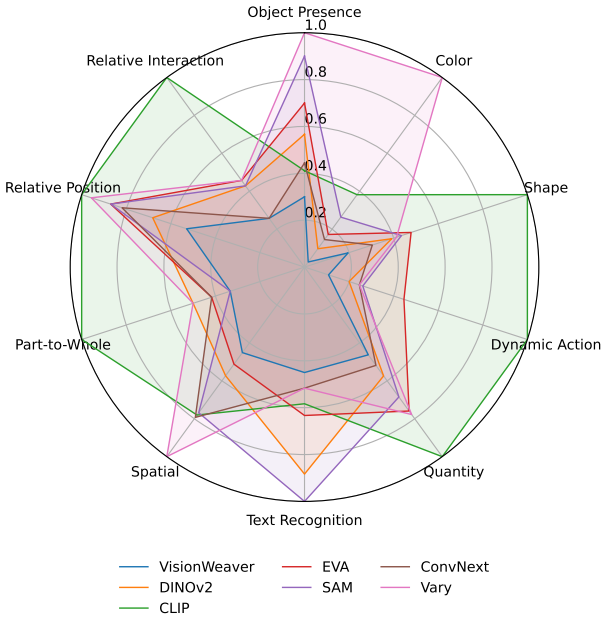

Figure 3: Results with different visual experts and our VisionWeaver on VHBench-10. The evaluation metric is the normalized error rate. Our method achieves lowest error rate in all ten tasks.

The results, summarized in Figure 3, reveal distinct hallucination characteristics correlated with the choice of visual encoder. For example, LVLMs utilizing the CLIP (Radford et al., 2021) visual encoder demonstrated lower error rates in tasks requiring global perception, such as identifying Object Presence. In contrast, models employing DINOv2 (Oquab et al., 2023), known for its focus on fine-grained details, performed better at perceiving attributes like Color and Action. Furthermore, an LVLM using the Vary visual encoder, pre-trained on text recognition tasks, exhibited significantly lower hallucination rates in text-related visual tasks within VHBench-10. The full evaluation results can be found in Appendix C.

These findings underscore the specialized strengths of different vision experts and how their individual biases influence an LVLM's susceptibility to specific types of hallucinations. Notably, when evaluating our proposed VisionWeaver (detailed in Chapter 4) on VHBench-10, it consistently achieved the lowest error rates across the full spectrum of hallucination categories. This superior performance highlights VisionWeaver's effectiveness in adaptively leveraging diverse visual expertise to mitigate a wide range of hallucinations, thereby demonstrating its capability in enhancing the reliability of LVLMs.

4 VisionWeaver

4.1 Overview

Generally, LVLMs comprise a visual perception module, a lightweight projection module, and a large language model. The visual perception module extracts visual information, the projection module aligns it with language embeddings, and the LLM generates text.

Our work focuses on mitigating object hallucinations from the visual perception module. Different visual encoders exhibit distinct hallucination behaviors; for instance, prevalent CLIP-like encoders, while extracting general semantic features, possess constrained perception for diverse visual scenes and fine-grained details. This inherent limitation motivates the integration of multiple, specialized vision experts. However, simply fusing features from these diverse experts (e.g., through feature addition or concatenation (Tong et al., 2024)) often yields suboptimal performance.

To address this challenge and effectively harness the complementary strengths of various visual encoders, we propose VisionWeaver. Instead of relying on a single, potentially limited encoder or a simplistic fusion, VisionWeaver aims to intelligently integrate multiple types of vision experts. As illustrated in Figure 4, our method primarily relies on two pivotal modules. The first is the Context-Aware Routing module, which utilizes global image features to produce soft weights, guiding the selection of the most appropriate experts for the given visual input. Second, we propose a knowledge enhancement module to effectively fuse the selected knowledge from these experts. More specifically, we utilize a linear adapter to integrate the representations from the chosen vision encoders. Through these modules, VisionWeaver can comprehensively encode visual inputs from diverse perspectives, thereby helping to reduce object hallucinations by leveraging the specialized capabilities of each integrated encoder.

4.2 Routing Vision Experts Representations

Context-Aware Expert Selection The context-aware expert routing mechanism leverages the global semantic features of an image to compute adaptive soft routing weights for selecting appropriate visual experts.

Concretely, we begin by extracting visual features from each expert. For subsequent routing, the outputs from all visual experts are combined using weighted fusion. The visual feature extraction process is defined as:

$$\mathbf{Z}_i = g_i(\mathbf{X}), \quad i = 1, \ldots, N \tag{1}$$

where g_i denotes the i-th visual experts, $\mathbf{Z}^i$ represents the i-th encoded feature.

To better guide the model in selecting a visual expert model suitable for the current scenario, it is essential to pick out a token that carries the key visual signals of the image. Previous studies have shown that the `[CLS]` token in the CLIP image encoder captures the key visual information of the image (Liang et al., 2022). Therefore, we select the `[CLS]` token as the indicator to guide the model. Next, based on the `[CLS]` token output by the CLIP image encoder, VisionWeaver learns to allocate the weight of each vision expert. The process can be formulated as follows:

$$\{\mathbf{I}_C, \mathbf{I}_P\} = \phi(\mathbf{X}) \tag{2}$$

$$\mathbf{A} = f(\mathbf{I}_C) \tag{3}$$

$$\mathbf{W} = \underset{1 \le j \le N}{softmax}\, \mathbf{A}_j \tag{4}$$

where ϕ is the CLIP encoder, $\mathbf{I}_C, \mathbf{I}_P$ are the CLS and patch token features after CLIP encoding, respectively. $f : \mathbb{R}^D \to \mathbb{R}^N$, D is the feature dimension of the CLIP. By now, we have already obtained the top-k vision experts and corresponding importance scores.

Expert Representation Fusion. In the CLIP vision encoder, Patch Token is obtained by dividing the input image into non-overlapping patches, flattening them into 1-dimensional vectors, and then projecting them through a linear layer. It mainly carries the local visual information of the image patches, and in the Transformer encoder, Patch Tokens interact with each other via the self-attention mechanism to help the model capture the dependencies between different image regions and learn global features, being arranged in the spatial order of the patches in the sequence. To better fuse the representation from the vision experts, we propose a simple yet effective way by aligning the router-guided representation and the patch token output by CLIP. The process can be formulated as:

$$\mathbf{Y} = \mathbf{W}_i \mathbf{Z}_i, i = 1, \ldots, N \tag{5}$$

$$\hat{\mathbf{I}} = \mathbf{I}_P + \mathbf{Y} \tag{6}$$

Here, $\mathbf{Z}_i$ denotes the representation from the i-th vision expert, and $\mathbf{W}_i$ is the corresponding learned

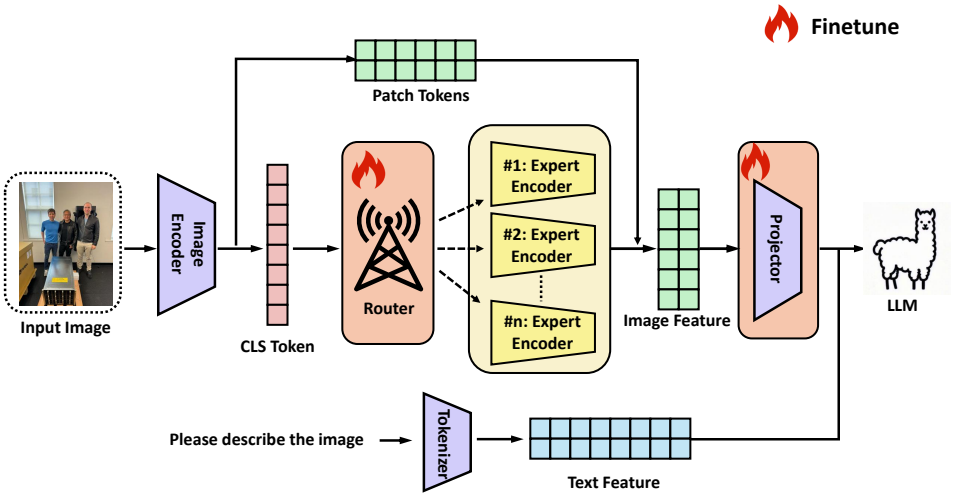

Figure 4: The pipeline of VisionWeaver. VisionWeaver performs a context-aware routing to solve a given question. The context-aware expert routing is performed in the first stage to select context-relevant experts. Next, we fuse the task-specific knowledge from these selected experts in a fine-grained manner.

weight. The aggregated expert representation is denoted as $\mathbf{Y}$, which shares the same dimensionality as both $\mathbf{Z}i$ and the CLIP patch token $\mathbf{I}_P$. The final visual representation $\hat{\mathbf{I}}$ is obtained by combining the expert features with the original CLIP representation through a residual-style connection.

The final output is then passed to the projector to map it into the LLM's embedding space(labeled as "Image Feature" in Fig.4).

5 Experiments

The present experiments were conducted based on the LLaVA-1.5 (Liu et al., 2024b) architecture. Specifically, the LLaVA-1.5 settings were followed, with CLIP-ViT-L-336px serving as the base visual encoder and a two-layer MLP acting as the visual projector. Concurrently, we substituted the LLM with the most recent versions of Llama3.2-Instruct-3B (Team, 2024) and Qwen2.5-Instruct-3B (Qwen et al., 2025). This substitution was made to ascertain the applicability of our method to the latest LLMs. The 3B version was selected due to its suitability for end-side deployment and its prevalent use in devices such as cell phones.

For multiple vision encoders in VisionWeaver, inspired by EAGLE (Shi et al., 2024), we chose ConvNext (Liu et al., 2022), EVA-02 (Fang et al., 2024), SAM (Kirillov et al., 2023), DINOv2 (Oquab et al., 2023), and Vary (Wei et al., 2025) as task-specific visual encoders, which were pre-trained on different downstream tasks with different visual capabilities. In order to align with CLIP encoders when processing images, we use interpolation to fix the input resolution of all encoders to 336×336 and the output token to 576. The output dimension is fixed to 1024 using a linear adapter.

5.1 Implementation Details

Our training pipeline consists of two stages: pre-training and supervised fine-tuning. For the pre-training phase, we trained our model on the LLaVA-Pretrain (Liu et al., 2024b) dataset using the AdamW optimizer with a batch size of 256 and a learning rate of 2×10^{-4} for 1 epoch. At this stage, we only adjust all projectors. Subsequently, in the supervised fine-tuning phase, we also use the AdamW optimizer to perform 1 epoch of fine-tuning using the LLaVA-Finetune (Liu et al., 2024b) dataset at batch size 128 and learning rate 2×10^{-5}. All parameters are adjusted at this stage. We further discuss the impact of parameter efficiency on performance in Appendix F. Our experiments were performed on 8 Nvidia A100 GPUs, with two phases using 8 and 16 hours, respectively.

5.2 Main Results

Hallucination Mitigation Evaluation of our VisionWeaver method for mitigating hallucinations in LVLMs was conducted using POPE (Li et al., 2023) and AutoHallusion benchmarks (Wu et al., 2024). POPE evaluates the level of hallucination in LVLMs by asking if there is an object ***O*** in the image. AutoHallusion evaluates the ability of LVLMs to combat hallucinations by creating conflicting

Table 1: Hallucination evaluation results on POPE (Li et al., 2023) and AutoHallusion (Wu et al., 2024). **VE** stands for Visual Encoder and **ME** stands for Multi Encoder, including CLIP, Convnext, DINOv2, EVA-02, SAM, Vary. **Avg.** is the average of the F1 metric from POPE and the Overall Accuracy metric from AutoHallusion.

LLM	Size	VE	Vision Weaver	POPE				AutoHallusion			Avg.
				Accuracy	Precision	Recall	F1	Overall	Synthetic	Real-World	
Vicuna	7B	*CLIP*	×	87.2	93.8	79.6	86.1	44.5	46.6	41.8	65.3
Llama3.2	3B	*CLIP*	×	87.7	93.4	81.1	86.8	44.3	45.7	44.8	65.6
		ME	×	88.7	94.8	81.9	87.9	47.6	46.3	49.2	67.8
		ME	✓	**89.5**	**95.1**	**83.3**	**88.8**	**48.2**	**47.0**	**49.6**	**68.5**
Qwen2.5	3B	*CLIP*	×	85.7	93.9	78.0	85.2	53.2	51.5	55.6	69.2
		ME	×	85.7	93.9	78.0	85.2	53.9	52.2	56.1	69.6
		ME	✓	**87.7**	**95.7**	**79.3**	**86.7**	**54.3**	**52.6**	**56.5**	**70.5**

images and inducing hallucinations in the model.

Table 1 shows the effectiveness of our method in mitigating hallucinations. We used three pedestal models: Vicuna-7B, which is the implementation of LLaVA-1.5 (Liu et al., 2024b), Llama3.2-Instruct3B, and Qwen2.5-Instruct-3B, with the results of Vicuna-7B serving as the baseline for our approach. The results reveal that: (1) The underlying architecture of a model can have a more significant impact on performance than its scale. The 3B models generally outperformed Vicuna-7B, confirming our suspicion that the newer model has greater capacity. (2) On the POPE benchmark, Llama3.2 with Multi Encoders and VisionWeaver achieved the strongest performance. In the AutoHallusion evaluation, Qwen2.5 demonstrated superior resistance to hallucination across both synthetic and real-world scenarios. Its overall accuracy was notably higher than both Vicuna and Llama3.2. (3) The average metric shows Qwen2.5-3B with Multi Encoders and VisionWeaver achieving the highest overall performance. This metric suggests that our VisionWeaver provides the most robust performance across different types of hallucination challenges.

Perceptual Perspective To demonstrat the broad generalizability of our method, we evaluated VisionWeaver on five standard LVLM benchmarks: MME (Fu et al., 2024), MMStar (Chen et al., 2024a), MMBench (Liu et al., 2024d), OCRBench (Liu et al., 2024f) and MathVista (Lu et al., 2024). Table 3 presents these evaluation results. We tested it with Llama3.2-3B and Qwen2.5-3B, comparing configurations where VisionWeaver was integrated (ME + VW) against baseline setups using a standard Visual Encoder (CLIP) and Multiple Encoders (ME) alone.

The experimental results demonstrate the consistent effectiveness of VisionWeaver across multiple benchmarks. It shows notable enhancements in MMBench and OCRBench tasks for Llama3.2, while delivering improvements in MME and MMStar benchmarks for Qwen2.5. These results consistently show that VisionWeaver is effective at improving model performance.

5.3 Systematic Analysis

In order to further validate the effectiveness of our VisionWeaver, we perform the validation from each of the following two perspectives: expert selection as well as fusion strategy. All experiments were performed using the Llama3.2-3B-Instruct model. The results are shown in Table 2.

Expert Selection We investigated the impact of different visual experts (VE) by conducting experiments on both POPE and AutoHallusion benchmarks. The results, as shown in Table 2, led to two key observations. First, different visual experts exhibited varying strengths and performance levels. Second, we found that simply increasing the number of visual encoders does not guarantee better performance. For instance, using all six encoders with additive fusion resulted in an average performance of 67.9%, which is slightly lower than the 68.4% achieved using only four specific encoders (CLIP, ConvNext, EVA, and SAM) with the same fusion strategy.

Fusion Strategy To evaluate the effectiveness of our proposed VisionWeaver, we conducted comprehensive experiments comparing three fusion strategies: feature summation (Add), feature concatenation (Concat), and our VisionWeaver. These strate-

Table 2: Results of a systematic analysis of expert selection and fusion strategies.

VE						Fusion	POPE				AutoHallusion			Avg.
CLIP	ConvNext	EVA	SAM	DINOv2	Vary		Acc	P	R	F1	Acc	S	R	
✓	-	-	-	-	-	-	87.7	93.4	81.1	86.8	44.3	45.7	44.8	65.6
-	✓	-	-	-	-	-	86.6	90.4	81.9	85.9	48.0	47.0	49.2	67.0
-	-	✓	-	-	-	-	87.0	87.7	86.0	86.8	47.6	47.4	47.9	67.1
-	-	-	✓	-	-	-	80.6	77.8	85.6	81.5	45.9	45.7	46.1	63.7
-	-	-	-	✓	-	-	87.7	91.9	82.6	87.0	41.5	42.0	40.9	64.3
-	-	-	-	-	✓	-	74.2	69.1	**87.4**	77.1	46.0	46.0	46.0	61.6
✓	✓	✓	-	-	-	*Add*	88.9	94.4	82.7	88.2	44.3	44.0	44.6	66.3
✓	✓	✓	✓	-	-	*Add*	89.6	93.9	84.6	**89.1**	47.6	**47.8**	47.2	68.4
✓	✓	✓	✓	✓	-	*Add*	88.2	**95.4**	80.2	87.1	44.9	46.6	42.9	66.0
✓	✓	✓	✓	✓	✓	*Add*	89.0	94.9	82.4	88.2	47.6	46.3	49.2	67.9
✓	✓	✓	✓	✓	✓	*Concat*	88.7	94.8	81.9	87.9	42.6	42.3	43.0	65.3
✓	✓	✓	✓	✓	✓	*VisionWeaver*	**89.5**	95.1	83.3	88.8	**48.2**	47.0	**49.6**	**68.5**

Table 3: Evaluation results of generalized vision benchmarks. **VE** stands for Visual Encoder, **ME** stands for Multi Encoders and **VW** stands for our VisionWeaver.

LLM	VE	VW	MME	MMStar	MMB	OCRB	MathVista
Llama3.2	*CLIP*	×	1382.15	37.54	67.41	31.41	27.14
	ME	×	1375.47	37.97	67.14	33.93	27.67
	ME	✓	**1392.45**	**39.86**	**69.76**	**35.61**	**29.63**
Qwen2.5	*CLIP*	×	1444.26	40.94	64.09	29.47	31.76
	ME	×	1440.91	41.57	67.84	32.72	33.08
	ME	✓	**1465.92**	**43.65**	**69.24**	**36.48**	**35.81**

gies were assessed using all six visual experts on both POPE and AutoHallusion benchmarks, with results detailed in Table 2. The experimental results reveal several key findings. First, feature summation demonstrated superior performance compared to feature concatenation. Summation achieved a POPE Accuracy of 89.0% and an AutoHallusion Accuracy of 47.6% (Avg. 67.9%), whereas concatenation resulted in a POPE Accuracy of 88.7% and an AutoHallusion Accuracy of 42.6% (Avg. 65.3%). This performance difference can be attributed to the challenges posed by the high-dimensional feature space created through concatenation, which potentially complicates the projection of visual features into the embedding space of the LLM. Among all three fusion strategies, VisionWeaver achieved optimal performance, delivering top scores with 89.5% POPE accuracy and 48.2% AutoHallusion accuracy (Avg. 68.5%). These results suggest that VisionWeaver more effectively integrates complementary information from different visual experts while maintaining the structural integrity of the feature space, leading to more effective hallucination suppression.

6 Limitations

Despite exploring fine-grained hallucinations of LVLMs, our work still has limitations. First, although our benchmark covers the most realistic problems as well as possible, there are still about 20% of hard-to-categorize realistic samples that are difficult to classify into this benchmark because they involve more complex scenarios and require more fine-grained design. Second, our benchmark is built based on GPT-4, which inevitably introduces a slight error. Finally, due to the limitation of computational resources, our experiments are built on several smaller scale models.

7 Conclusion

In this paper, we present the VHBench-10, a new benchmark that systematically classifies the visual hallucinations of the LVLM into 10 different categories, allowing for their fine-grained analysis. By replacing the visual encoder of LVLM , we found that different encoders lead to diverse hallucinatory tendencies. Based on these insights, we propose VisionWeaver, a powerful LVLM architecture that incorporates a context-aware expert routing mechanism and a knowledge augmentation module to efficiently leverage task-specific visual expertise. Extensive experiments demonstrate the effectiveness of our approach, establishing VIsionWeaver as a powerful solution for alleviating hallucinations in LVLMs. This work opens new avenues for developing more reliable and accurate LVLMs.

References

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*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. 2024a. Are we on the right way for evaluating large vision-language models? *Preprint*, arXiv:2403.20330.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. 2024. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Xuan Gong, Tianshi Ming, Xinpeng Wang, and Zhihua Wei. 2024. Damro: Dive into the attention mechanism of lvlm to reduce object hallucination. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7696–7712.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2024. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. *Preprint*, arXiv:2310.14566.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Xin He, Longhui Wei, Lingxi Xie, and Qi Tian. 2024. Incorporating visual experts to resolve the information loss in multimodal large language models. *arXiv preprint arXiv:2401.03105*.

Hongyu Hu, Jiyuan Zhang, Minyi Zhao, and Zhenbang Sun. 2023. Ciem: Contrastive instruction evaluation method for better instruction tuning. *arXiv preprint arXiv:2309.02301*.

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.

Jitesh Jain, Jianwei Yang, and Humphrey Shi. 2024. Vcoder: Versatile vision encoders for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27992–28002.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2023. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. *Preprint*, arXiv:2311.16922.

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*.

Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. 2023. Llm-grounded video diffusion models. *arXiv preprint arXiv:2309.17444*.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. 2022. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2023. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*.

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

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024c. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024d. Mmbench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

Yufang Liu, Tao Ji, Changzhi Sun, Yuanbin Wu, and Aimin Zhou. 2024e. Investigating and mitigating object hallucinations in pretrained vision-language (clip) models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18288–18301.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024f. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12).

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *Preprint*, arXiv:2310.02255.

Jinjie Mai, Jun Chen, Guocheng Qian, Mohamed Elhoseiny, Bernard Ghanem, et al. 2023. Llm as a robotic brain: Unifying egocentric memory and control.

Meta. 2025. paperswithcode.com.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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.

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, et al. 2024. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv preprint arXiv:2408.15998*.

Shuncheng Tang, Zhenya Zhang, Jixiang Zhou, Lei Lei, Yuan Zhou, and Yinxing Xue. 2024. Legend: A top-down approach to scenario generation of autonomous driving systems assisted by large language models. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 1497–1508.

Llama 3 Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578.

Chenxi Wang, Xiang Chen, Ningyu Zhang, Bozhong Tian, Haoming Xu, Shumin Deng, and Huajun Chen. 2024a. Mllm can see? dynamic correction decoding for hallucination mitigation. *arXiv preprint arXiv:2410.11779*.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and Jitao Sang. 2024b. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *Preprint*, arXiv:2311.07397.

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*.

Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2025. Vary: Scaling up the vision vocabulary for large vision-language model. In *European Conference on Computer Vision*, pages 408–424. Springer.

Xiyang Wu, Tianrui Guan, Dianqi Li, Shuaiyi Huang, Xiaoyu Liu, Xijun Wang, Ruiqi Xian, Abhinav Shrivastava, Furong Huang, Jordan Lee Boyd-Graber, et al. 2024. Autohallusion: Automatic generation of hallucination benchmarks for vision-language models. *arXiv preprint arXiv:2406.10900*.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2023. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*.

Zihao Yue, Liang Zhang, and Qin Jin. 2024. Less is more: Mitigating multimodal hallucination from an eos decision perspective. *Preprint*, arXiv:2402.14545.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2023. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*.

A Ten Hallucination Sub-Categories

To provide a clear and structured understanding of the various errors that can occur in vision-language models, this section meticulously defines and categorizes ten distinct hallucination sub-types. These categories are grouped under broader error classes such as Detection, Segmentation, Localization, and Classification Hallucinations, offering a comprehensive taxonomy for analyzing model failures.

1. **Detection Hallucination**: In computer vision tasks requiring precise object recognition and localization, we categorize detection hallucinations into three distinct subtypes based on error manifestations:
   (a) **Category Hallucination** (Object Presence Misidentification): Occurs when the model incorrectly identifies the presence of an object category absent in the visual context. *Example*: While the image solely depicts a beach and sea, the model erroneously reports "a man surfing".
   (b) **Counting Hallucination** (Object Quantity Misestimation): Arises from the model's failure to accurately enumerate instances of detected objects. *Example*: An image containing three felines is incorrectly described as "two cats playing".
   (c) **Occlusion Hallucination** (Partial Observation Fallacy): Results from making holistic object judgments based on incomplete visual evidence. *Example*: Inferring a complete car's presence solely from visible tire segments.

2. **Segmentation Hallucination**: Unlike object detection tasks that operate at the instance level, segmentation tasks require a pixel-level understanding of object boundaries and spatial characteristics. We categorize segmentation hallucination into two subtypes:
   (a) **Text Hallucination**: Character-level misinterpretation in scene text recognition. This occurs when models confuse visually similar glyphs despite accurate localization. *Example*: Misrecognizing "**Cloud**" as "**Clown**" due to font artifacts.
   (b) **Shape Hallucination**: Geometric distortion in object contour perception. Pixel-level errors in boundary prediction lead to incorrect shape interpretations. *Example*: Describing a quadrilateral table as circular when partial occlusion disrupts edge continuity.

3. **Localization Hallucination**: Refers to systematic errors in spatial perception where models misinterpret coordinate systems or geometric relationships. We identify two distinct manifestations:
   (a) **Absolute Positioning Hallucination**: Failure in Cartesian coordinate comprehension. Models exhibit metric measurement inaccuracies in defined coordinate frames. *Example*: Locating a tree at (x_1, y_1) while its true position is (x_2, y_2), resulting in "left-right" inversion descriptions.
   (b) **Relative Positioning Hallucination**: Breakdown in spatial relation reasoning. Models fail to preserve topological relationships between entities. *Example*: A car approaching from behind is localized as preceding the pedestrian due to motion parallax misinterpretation.

4. **Classification Hallucination**: Systematic errors in categorical attribution across visual-semantic alignment. We dissect this phenomenon through three perceptual failure modes:
   (a) **Color Hallucination**: Spectral sensitivity breakdown in color perception. Models confuse the color of the object despite correct object recognition. *Example*: Describing a red car as blue.
   (b) **Action Hallucination**: Temporal-semantic disconnection in motion parsing. Models misinterpret static poses as dynamic actions. *Example*: Classifying a static "holding basketball" pose as the dynamic "dunking" action due to lack of temporal context.
   (c) **Relative Interaction Hallucination**: Failure in social signal processing. Models incorrectly infer interpersonal dynam-

ics from spatial configurations. *Example*: Interpreting two agents facing each other with 1.2m distance as "handshaking" rather than "conversing".

B Detailed Benchmark Construction

To elucidate the methodology behind the creation of our novel benchmark, VHBench-10, this section provides a step-by-step account of its construction process. The aim is to ensure transparency and reproducibility in how the benchmark was developed to systematically evaluate specific hallucination types.

For the VHBench-10 benchmark construction, each image was paired with only one type of hallucination in each generated instance. The process followed these steps:

1. We randomly selected 2,000 images with their corresponding detailed captions from the LLaVA-ReCap-118K dataset.

2. For each image-caption pair, we developed 10 different specialized prompts—one for each type of hallucination (Category, Counting, Occlusion, Text, Shape, Absolute Positioning, Relative Positioning, Color, Action, and Relative Interaction).

3. Each prompt directed GPT-4 to modify the original caption to introduce a specific type of hallucination while maintaining consistency with the rest of the description. The general structure of the prompt provided to GPT-4 is detailed in Table 8. For example, when creating text hallucinations, specific instructions within this prompt structure guided GPT-4 to:

    - Determine if there was modifiable text content (such as signs, books, screen displays).
    - If present, modify only the text content interpretation while keeping other elements unchanged.
    - Maintain the original level of detail and keep the context plausible.

4. We applied all 10 prompts (each tailored for a specific hallucination type but following the general structure outlined in Table 8) to each image-caption pair. Depending on the image content, an image might yield between 0-10 hallucinated captions. For instance, if an image did not contain any text, it would not generate a text hallucination caption. Similarly, if an image did not contain multiple objects, it might not support a counting hallucination.

5. Importantly, each generated hallucinated caption contained exactly one type of hallucination (not multiple types), making it possible to precisely evaluate model performance against specific hallucination categories.

6. This resulted in our final dataset of 9,648 instances, each containing a ternary of (I, R, H) where I is the image, R is the real caption, and H is the caption with a specific type of hallucination.

This approach allowed us to create a more focused benchmark that could systematically evaluate an LVLM's vulnerability to specific types of hallucinations.

C Evaluation Process on VHBench-10

To detail how models are assessed using our benchmark, this section describes the specific evaluation protocol employed on VHBench-10. This includes the input prompting strategy and the metric used to determine model error rates against different hallucination types.

A single sample of our VHBench-10 contains a ternary (I, R, H). We utilize the following prompt, where the model is given either R or H for the `<caption>` field:

```
'<image>\nDescribe the image: <caption>'
```

We input $(I + R)$ and $(I + H)$ to the model separately to test the PPL of its output.

If it exhibits $PPL(I + R) > PPL(I + H)$, it means the model erroneously assigns a higher probability to H over R, which means the model is wrong. We replaced the vision encoder of LLaVA-1.5 with different experts and performed the above operation on all samples of VHBench-10, recording the error rate of each expert.

We evaluated the error rates of the original LLaVA (CLIP), five different expert encoders, and our VisionWeaver on 10 hallucination types. The results are shown in Table 4.

D Details of Parameter Efficiency

To demonstrate the effectiveness of our proposed method in resource-limited settings, we conducted two sets of experiments:

Table 4: Error rates of different visual encoders on various vision tasks.

Method	Category	Color	Shape	Action	Counting	Text	Absolute Position	Occlusion	Relative Position	Relative Interaction
CLIP	2.04	4.39	4.30	4.73	7.25	9.43	11.02	5.43	10.27	6.82
ConvNext	2.21	1.67	1.31	1.17	3.76	8.36	11.19	2.26	8.41	1.75
DINO	2.81	1.11	1.68	0.95	4.16	14.29	8.09	2.71	6.99	2.92
EVA	3.47	1.98	2.06	2.11	5.50	10.24	7.23	2.26	8.93	3.12
SAM	4.47	3.03	1.87	1.24	4.97	16.17	11.19	1.81	8.93	2.92
Vary	4.96	11.50	1.78	1.17	5.64	8.36	14.11	2.71	9.82	3.12
VisionWeaver	1.49	0.31	0.84	0.51	3.36	7.28	6.37	1.81	5.43	1.75

1. Freeze the vision part and only full fine-tune the projectors and LLM.

2. Freeze the vision part and fine-tune the projectors and LLM using LORA.

We used POPE to evaluate our method. As shown in Table 5, our approach maintains its advantages in resource-constrained scenarios as well.

Table 5: Performance of different training strategies.

Method	Accuracy	Precision	Recall	F1
LLaVA-1.5-Llama3.2-3B (w/o vision)	87.0	94.6	78.5	85.9
LLaVA-1.5-Llama3.2-3B (w/ vision)	87.7	93.4	81.1	86.8
VisionWeaver (w/o vision, LORA)	88.1	94.7	80.7	87.1
VisionWeaver (w/o vision)	88.6	**95.2**	81.4	87.8
VisionWeaver (w/ vision)	**89.5**	95.1	**83.3**	**88.8**

E Comparison with SOTA

To demonstrate that our VisionWeaver can alleviate model hallucinations compared to competing methods, this section presents a comparative analysis of VisionWeaver against other state-of-the-art (SOTA) methods on the POPE benchmark. These methods are:

1. **SEOSS**(Yue et al., 2024) proposes a method to reduce multimodal hallucinations. It achieves this by improving how models make the end-of-sequence decision. This adjustment aims to prevent the generation of ungrounded information.

2. **OHD-Caps**(Liu et al., 2024e) introduces a counterfactual data augmentation method. This method is designed to mitigate object hallucinations in CLIP models. It is also effective for larger vision-language models that utilize CLIP as their visual encoder.

3. **DAMRO**(Gong et al., 2024) presents a training-free strategy to address object hallucinations in Large Vision-Language Models (LVLMs). The method targets hallucinations caused by misdirected attention to background tokens, an issue often linked to the visual encoder. Specifically, the DAMRO strategy employs the Vision Transformer's CLS token to identify and then suppress the influence of these outlier tokens during the decoding process.

4. **DeCo**(Wang et al., 2024a) develops a training-free dynamic correction decoding strategy for Multimodal Large Language Models (MLLMs). It addresses hallucinations that occur when correct visual information, initially recognized in earlier model layers, is suppressed by strong language priors in deeper layers. The DeCo strategy mitigates these hallucinations by leveraging this preceding-layer knowledge to adjust the final output logits.

As shown in Table 6, we have compared our approach on POPE against other SOTA methods, and the results demonstrate that our method performs competitively with these methods.

Table 6: Performance Comparison on POPE Dataset.

Method	Accuracy	Precision	Recall	F1
LLaVA-1.5-Llama3.2-3B	87.7	93.4	81.1	86.8
SEOSS (Yue et al., 2024)	86.8	93.5	79.5	86.0
OHD-Caps (Liu et al., 2024e)	81.2	90.9	85.1	87.9
DAMRO (Gong et al., 2024)	85.3	88.8	81.1	84.7
DeCo (Wang et al., 2024a)	-	-	-	86.7
VisionWeaver	89.5	95.1	83.3	88.8

F Computational Efficiency of Our Method

To underscore the practical viability of our approach for real-world applications, this section elaborates on the design aspects that contribute to its computational efficiency and presents empirical measurements of inference time.

Our approach can avoid introducing significant latency during inference. This is achieved through several key design aspects:

1. **Lightweight Visual Encoders**: Our n visual encoder experts have a combined size of approximately 1 billion parameters (for all 5 experts), which is relatively small compared to the language model (LLM) component (over 3 billion parameters). As a result, the main computational load resides within the LLM component.

2. **Efficient Token Aggregation**: The VisionWeaver module performs a weighted aggregation of visual tokens from various experts onto the output tokens of the CLIP encoder. Crucially, this process maintains the same number of visual tokens that are input to the large model. This prevents any additional computation time being introduced in the large model component.

3. **KV Caching Utilization**: During the inference stage, we utilize KV caching. This means that our method processes the image through the experts only once, during the prefill phase. Following this, all visual tokens are cached, which eliminates redundant computations in the subsequent generation steps.

We empirically measured the average inference time with and without the VisionWeaver module when generating 100 random captions. The results are presented in Table 7.

Table 7: Comparison of average inference time.

Methods	**Prefill Time** (ms)	**Inference Time** (ms)	**Prefill Percentage** (%)
LLaVA-1.5-Llama3.2-3B	50.99	1273.20	3.85
VisionWeaver	99.46	1201.52	7.64

Table 8: Unified Prompt Template for Generating Specific Hallucinations.

Prompt Template

 # Task Description

Based on the input image description, determine if there are modifiable

`{{MODIFICATION_TASK_SPECIFICS}}`.

If present, modify only the

`{{MODIFICATION_TASK_SPECIFICS}}`

while keeping other elements unchanged.

 # Input Format

Image description text

 # Output Format

If no `{{EXISTENCE_CONDITION_DESCRIPTION}}` exists, output: NO

If exists, output modified description

 # Guidelines

- First determine if `{{EXISTENCE_CONDITION_DESCRIPTION}}` exist
- Modified `{{MODIFIED_ELEMENTS_NAME}}` must be logically consistent
- `{{UNCHANGED_CONSTRAINT_TEXT}}`
- Maintain original level of detail
- Context should remain plausible

 # Input

`{input}`

 # Output