ZINA: Multimodal Fine-grained Hallucination Detection and Editing

Yuiga Wada[1,2,3] Kazuki Matsuda[2] Komei Sugiura[1,2] Graham Neubig[3]
[1]Keio AI Research Center [2]Keio University [3]Carnegie Mellon University
https://yuiga.dev/zina

Abstract

Multimodal Large Language Models (MLLMs) often generate hallucinations, where the output deviates from the visual content. Given that these hallucinations can take diverse forms, detecting hallucinations at a fine-grained level is essential for comprehensive evaluation and analysis. To this end, we propose a novel task of ***multimodal fine-grained hallucination detection and editing*** *for MLLMs. Moreover, we propose* ZINA*, a novel method that identifies hallucinated spans at a fine-grained level, classifies their error types into six categories, and suggests appropriate refinements. To train and evaluate models for this task, we construct VisionHall, a dataset comprising 6.9k outputs from twelve MLLMs manually annotated by 211 annotators, and 20k synthetic samples generated using a graph-based method that captures dependencies among error types. We demonstrated that* ZINA *outperformed existing methods, including GPT-4o and Llama-3.2, in both detection and editing tasks.*

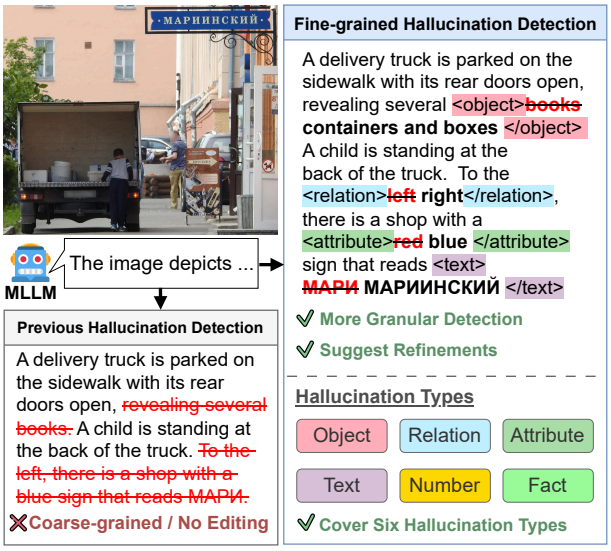

Figure 1. Overview of the proposed task. In contrast to conventional tasks, the model is expected to detect hallucinated spans at a fine-grained level, classify their types based on a taxonomy, and suggest appropriate refinements.

1. Introduction

Multimodal Large Language Models (MLLMs) have emerged as powerful systems for a broad range of vision-language tasks [1, 6, 13, 19, 30, 32, 34, 46]. MLLMs often produce *hallucinations* in image captioning tasks [11, 20, 49], undermining the reliability in practical applications. To further advance the development of MLLMs, evaluating and analyzing hallucinations is essential.

Although several works [20, 29, 49] have attempted to address hallucination detection in MLLM outputs, they have primarily focused on coarse-grained detection. While hallucinations can take diverse forms [11], coarse-grained methods often classify them into only a few broad categories (e.g. only object hallucinations; [20, 29]). Moreover, these methods typically detect hallucinations at the sentence level or across relatively long spans [20].

These approaches limit their ability to support detailed error analysis. For instance, consider the caption shown in Fig. 1, which contains diverse hallucinated spans — such as incorrect objects, colors, relationships, and scene text. While previous methods may label phrases such as "revealing several books" as hallucinated, effective analysis demands pinpointing the specific error ('books'), classifying its type ('object'), and suggesting a suitable correction ("containers and boxes")

To address these limitations, we propose **multimodal fine-grained hallucination detection and editing**, a novel task in which a model detects hallucinated spans, classifies their types according to a taxonomy, and suggests appropriate refinements. By consolidating categories from several works [4, 11, 38], our taxonomy classifies hallucinations in MLLMs' outputs into six categories. *Editing* can contribute to improving MLLMs, as they can serve as *silver-standard* data, offering span-level corrections [20, 38].

We formalize the proposed task as a tagging problem, as shown in Fig. 1. The model is expected to place tags on each hallucinated span, similarly to previous tasks designed for

text-only LLMs (e.g. [38]). The proposed task is particularly challenging because, as we will demonstrate, even advanced MLLMs, such as GPT-4o [1] and Llama-3.2 [19], perform poorly on both the *detection* and *editing* tasks.

To address this challenge, we propose ZINA, a novel method for fine-grained hallucination detection and editing in MLLMs. The method consists of two main components: a detector MLLM that identifies hallucinated spans, and a reviewer MLLM that determines where to apply tags and suggests appropriate refinements. Unlike previous works [3, 38] our method explicitly separates the responsibilities of token copying and hallucination detection/editing, reducing the complexity of each reasoning task.

Moreover, we construct the VisionHall dataset, which consists of two subsets for a detection task and an editing task. For the detection task, the dataset contains approximately 6.9k outputs generated by twelve MLLMs, with hallucinated spans manually annotated by 211 annotators. For the editing task, we create 20k synthetic samples by injecting errors into image captions using a novel graph-based approach, capturing the dependencies of errors.

The main contributions of this paper are as follows:

- We proposed a novel task of multimodal fine-grained hallucination detection and editing for MLLMs.
- We also proposed ZINA, a novel method designed for fine-grained hallucination detection and editing.
- We constructed VisionHall, a semi-synthetic dataset consisting of image captions generated by 12 MLLMs.
- ZINA outperformed existing methods on the VisionHall dataset in both detection and editing tasks. In particular, ZINA significantly outperformed both Llama-3.2 and GPT-4o, achieving gains of 28.2 and 15.8 points, respectively.

2. Related Works

Hallucination detection and editing in LLMs. Several studies have addressed hallucination detection by classifying statements based on factual correctness [9, 17, 27, 35, 37]. SelfCheckGPT [35] detects entity-level hallucinations by comparing multiple LLM outputs, treating consistent responses as factual and divergent ones as hallucinated. [27] employs GPT-4 to extract factual statements verifiable against world knowledge, and then identifies hallucinated spans within them. In contrast, FAVA [38] performs fine-grained hallucination detection and editing by leveraging retrieval-augmented language models.

Hallucination analysis in MLLMs. Various hallucination detection methods have been proposed for MLLMs [11, 20, 29, 49, 54]. POPE [29] focuses on object hallucinations in discriminative tasks by prompting MLLMs with simple yes/no questions about specific objects. Although POPE is limited to discriminative tasks, AMBER [49] addresses this gap by leveraging the CHAIR metric [42], which measures the proportion of mentioned objects that are not present in the image. Unlike these rule-based methods, MHalDetect [20] finetunes InstructBLIP and performs hallucination detection as a ternary classification task. UniHD [11] detects hallucinations in MLLMs by first extracting verifiable claims and then validating each claim using tools such as object detection and scene-text recognition systems.

Beyond these coarse-grained approaches, a few studies [25, 40] have proposed fine-grained hallucination detection methods. One such method is HalLocalizer [40], which performs token-level localization of four hallucination types using a bidirectional VisualBERT encoder with linear classification heads. A key difference from these methods is that our detection task is explicitly designed for a subsequent editing task. Specifically, replacing the detected spans should be sufficient to correct the hallucination. However, existing methods [11, 20, 25, 40] such as HalLocalizer rely on token-level detection, which does not guarantee that editing only the detected tokens corrects the caption.

3. Multimodal Fine-grained Hallucination Detection and Editing

We propose a novel task of **multimodal fine-grained hallucination detection and editing** for MLLMs. We follow prior work [20, 29, 49] in defining hallucinations as errors where the MLLM-generated description deviates from the visual content of the image.

3.1. Taxonomy

Based on prior work [4, 11, 38] and insights from the pilot annotation, we define a taxonomy comprising six error types:

- Object: Incorrect descriptions of specific objects or entities.
- Attribute: Inaccurate mentions of properties such as color, size, or shape.
- Number: Incorrect mentions of quantities or numerical values.
- Text: Inaccurate descriptions of scene text or written content visible in the image.
- Relation: Errors in semantic relationships of objects (e.g., prepositions or adjectives) within the description.
- Fact: Incorrect mentions of named entities such as people, places, or countries.

Details of the taxonomy construction are provided in Appendix A.

Table 1 presents examples of hallucinations categorized by our taxonomy. The table also shows the distribution of error types in generated captions for GPT-4o [1], Qwen2.5-VL-7B-Instruct [7], and Qwen2.5-VL-72B-Instruct on the

Types	Example	GPT-4o [%]	Q-7B [%]	Q-72B [%]
Object	A ~~dog~~ cat is sitting on the couch ...	30.13	27.32	40.42
Attribute	~~Blue~~ Red bicycles are leaning ...	34.73	33.82	36.69
Number	~~Three~~ Four people are sitting ...	10.04	9.76	6.17
Text	A sign says ~~"Restaurant"~~ "Welcome" ...	12.27	14.96	8.60
Relation	The coffee cup is on the ~~left~~ right side ...	8.23	11.38	7.31
Fact	~~Steve Jobs~~ Steve Wozniak holding ...	4.60	2.76	0.81

Table 1. Examples of hallucinations categorized by our taxonomy, along with the distribution of error types in outputs from GPT-4o, Qwen2.5-VL-7B-Instruct (Q-7B), and Qwen2.5-VL-72B-Instruct (Q-72B) on DCI dataset [47].

DCI dataset [47]. These distributions provide supporting evidence for the validity of the proposed taxonomy.

3.2. Task Definition

In this task, given an MLLM-generated description x_{desc} for an image x_{img}, along with a reference caption x_{ref}, a model detects hallucinated segments, classifies them according to our taxonomy, and suggests appropriate refinements. Specifically, the output is defined as $\hat{y} = \{\hat{\mathcal{Y}}_{\text{text}}, \hat{\mathcal{Y}}_{\text{edit}}, \hat{\mathcal{Y}}_{\text{type}}\}$, where $\hat{\mathcal{Y}}_{\text{text}}$, $\hat{\mathcal{Y}}_{\text{edit}}$, and $\hat{\mathcal{Y}}_{\text{type}}$ denote the sets of hallucinated words, their corresponding edits, and their error types, respectively. Each $\hat{y}_{\text{type}} \in \hat{\mathcal{Y}}_{\text{type}}$ is one of seven types: six from the taxonomy and one no-hallucination class.

Since reliable hallucination detection cannot be achieved using only the image as input, it is standard practice to incorporate external knowledge [11, 38]. Conventional tasks (e.g. [11]) typically assume that detectors access intermediate tools such as object detectors and OCR systems to obtain such knowledge. However, these sources often contain errors that hinder appropriate hallucination detection. Therefore, our task assumes the availability of human-written reference captions x_{ref} as a reliable source for evaluation.

3.3. Evaluation Metrics

Detection task. Following previous works [16, 38, 44], we employed F_1 score as an evaluation metric for the detection tasks. We first compute precision and recall by comparing $\{y^{(i)}_{\text{text}}, y^{(i)}_{\text{type}}\}_{i=1}^{N}$ and $\{\hat{y}^{(i)}_{\text{text}}, \hat{y}^{(i)}_{\text{type}}\}_{i=1}^{N}$, and then calculate the F_1 score as the harmonic mean of precision and recall.

Editing task. We evaluate editing models by sentence-level metrics, CLIP-S [22] and PAC-S [43]. These metrics are suitable for evaluating edited texts as they are standard metrics for image captioning [22, 23, 36, 43, 48]. Further details of the evaluation metrics are provided in Appendix B.

Overall performance. Previous works (e.g. [38]) typically assessed overall model performance with sentence-level metrics such as CLIP-S [22] and FactScore [37]. This is because edited answers often have diverse valid forms, making span-level ground truth hard to define. Sentence-level metrics evaluate editing models by computing cosine similarity between sentence embeddings. However, as they encode both edited and unedited content into a single embedding, they often overlook the impact of small edits, as unchanged portions dominate.

To address this limitation, we introduce fine-grained metrics that compare spans based on their embeddings, rather than encoding the entire sentence as a whole. Our metrics are inspired by detection metrics based on F_1 scores computed via exact matching. However, instead of using exact matches, we compare spans based on embedding-based similarity.

Using the similarity function sim, we define precision and recall as follows:

$$\text{Precision} = \frac{\sum_{i \in N} \text{sim}(\hat{y}^{(i)}_{\text{text}}, y^{(i)}_{\text{text}})}{\sum_{i \in N} \mathbf{1}[\hat{y}^{(i)}_{\text{type}} \neq 0]}, \tag{1}$$

$$\text{Recall} = \frac{\sum_{i \in N} \text{sim}(\hat{y}^{(i)}_{\text{text}}, y^{(i)}_{\text{text}})}{\sum_{i \in N} \mathbf{1}[y^{(i)}_{\text{type}} \neq 0]}, \tag{2}$$

where N denotes the number of hallucinated words in the ground truth, respectively. We then compute the F_1 score as the harmonic mean of precision and recall. As the similarity function sim, we adopt cosine similarity computed on BERT [14] and CLIP [41] embeddings, because their similarities are widely used as evaluation metrics in image captioning [22, 56]. We refer to these metrics as BERT-F_1 and CLIP-F_1, respectively.

4. Methodology

Previous approaches (e.g. [38]) generate outputs by copying the original sentence verbatim and inserting tags where necessary. These approaches face several challenges: (i) the model should reproduce the original sentence exactly, word by word; (ii) these approaches require a model to simultaneously determine, token by token, where to insert opening and closing tags; (iii) due to exposure bias in autoregressive generation, a single mistake can compromise the structural consistency of the entire output, often resulting in malformed tag sequences.

4.1. Proposed Method: ZINA

To address these limitations, we propose ZINA, a two-stage system that detects hallucinations based on our taxonomy

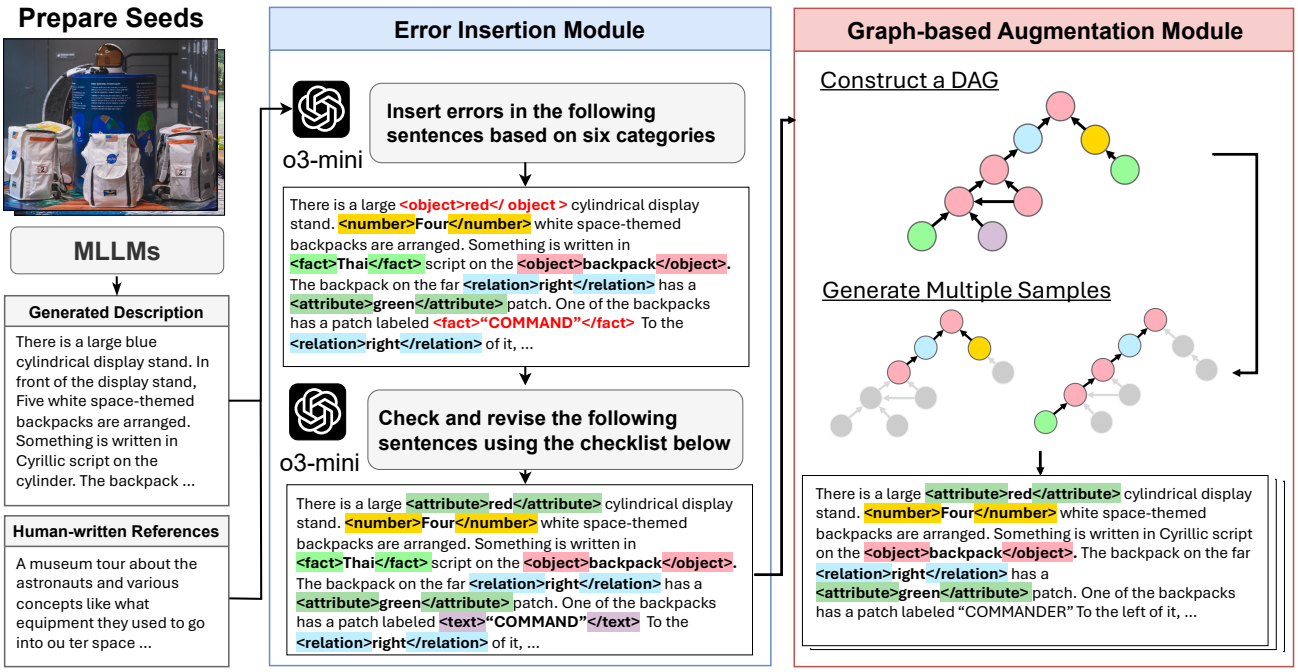

Figure 2. Overview of the graph-based synthetic data generation process. We first obtain seed descriptions by leveraging various MLLMs. Subsequently, the Error Insertion module injects errors while considering inter-span dependencies of errors. The Graph-based Augmentation module then constructs a DAG and prunes it to generate diverse training samples.

and suggests refinements. The proposed method consists of two main components: the detector MLLM $\mathcal{M}_{\text{det}}$ and the reviewer MLLM $\mathcal{M}_{\text{rev}}$. Our central strategy is to *delegate token copying to a deterministic function*, allowing the language model to focus solely on the detection and editing sub-tasks for erroneous spans. This strategy, which decouples the detection and tagging processes, can be broadly applied to existing hallucination detection methods [20, 38].

We begin by constructing the prompt p_{det} using $\{x_{\text{desc}},\ x_{\text{img}},\ x_{\text{ref}}\}$. In few-shot settings, p_{det} additionally incorporates n few-shot examples. The prompt design follows prior work (e.g., [38]), and the full prompt templates are provided in the Appendix H. We then feed p_{det} into the detector MLLM $\mathcal{M}_{\text{det}}$ to obtain hallucinated spans $\{h^{(i)}_{\text{type}}\}_{i=1}^{M}$ along with their corresponding error types $\{h^{(i)}_{\text{error}}\}_{i=1}^{M}$, where M denotes the number of predicted hallucinations.

Subsequently, for each hallucinated span, we generate a tagged sequence z_i as follows:

$$z_i = \mathcal{T}(x_{\text{desc}}, h^{(i)}_{\text{text}}, h^{(i)}_{\text{type}}), \tag{3}$$

where $\mathcal{T}$ is a deterministic function that inserts tags into the hallucinated span based on its error type. Unlike previous works [38], we adopt $\mathcal{T}$ as it delegates token copying to a deterministic function. This design allows $\mathcal{M}_{\text{det}}$ to focus exclusively on the hallucination detection.

Given $\{z_i\}_{i=1}^{M}$, the reviewer MLLM $\mathcal{M}_{\text{rev}}$ assesses whether the hallucinated span is appropriately tagged within its surrounding context, and then outputs $\hat{\mathcal{Y}}_{\text{text}}$ and $\hat{\mathcal{Y}}_{\text{type}}$ as follows:

$$\hat{y} = \mathcal{M}_{\text{rev}}(z_i, x_{\text{img}}, x_{\text{ref}}). \tag{4}$$

In the editing setting, $\mathcal{M}_{\text{rev}}$ additionally generates suitable refinements $\hat{\mathcal{Y}}_{\text{edit}}$ for each hallucinated span. We use Qwen2.5-VL-72B-Instruct [7] to initialize $\mathcal{M}_{\text{det}}$ and $\mathcal{M}_{\text{rev}}$. For both models, we employ cross-entropy as the loss function.

4.2. Synthetic Training Data Curation

The development of a hallucination editor requires a diverse set of training samples. In this field, it is standard to synthesize training data by injecting artificial hallucinations into correct sentences [5, 38, 40]. These methods assume hallucinations are self-contained and span-localized; however, image-grounded hallucinations in MLLMs often span multiple semantically related regions, which makes it essential to capture dependency structures among errors.

To this end, we propose a novel data construction method that explicitly handles error dependencies using graph structures. Our approach consists of two main modules: the Error Insertion (EI) module and the Graph-based Augmentation (GraphAug) module. Fig. 2 shows an overview of our data

generation pipeline. We first obtain seed descriptions. The EI module then injects errors while considering dependencies, and the GraphAug module constructs a graph and prunes it for generating diverse errors.

Seed descriptions. In general, synthetic data construction requires a set of seed descriptions as input. Prior work [38] generated seed data from external knowledge sources that are guaranteed to be factually correct. Synthetic hallucinations are then introduced by replacing specific spans within these descriptions. However, in image-grounded generative tasks, factually guaranteed descriptions are scarce, making such approaches less applicable. Although standard image captioning datasets provide reference captions [2, 31], naively replacing parts of these captions tends to produce limited variation and may be suboptimal for training $\mathcal{M}_{\text{rev}}$.

Therefore, we use MLLM-generated descriptions that were judged by annotators to be free of hallucinations as seed inputs. Specifically, we leverage hallucination-free descriptions from the newly constructed VisionHall dataset, as detailed in Section 5.

Error insertion. The EI module first injects hallucinations into the hallucination-free descriptions while simultaneously capturing the dependencies of errors. To perform the insertion, we adopt o3-mini [39], which enables simultaneously performing hallucination injection and explicit capture of inter-error dependencies. This module outputs data in XML format, which allows for the specification of hierarchical dependencies. Here, we define an inter-error dependency as a causal or referential relationship between two or more hallucinated spans, where one hallucination (e.g., a non-existent object) induces or conditions the generation of subsequent hallucinations (e.g., attributes or relations related to that object). For example, if an MLLM mentions "apples" in the initial sentence despite no apples being present in the image, it may then mention incorrect relations between the apples and other objects due to its autoregressive nature, as pointed out in [58].

After insertion, the module re-validates the dependencies for consistency, as we empirically observed that inserted errors often link to non-existent spans. The full prompt used for this process is provided in Appendix H.

Graph-based augmentation. The GraphAug module builds a directed graph representing dependencies among inserted errors. It removes cycles by detecting them and deleting descendant nodes, yielding a Directed Acyclic Graph (DAG) that captures structural error relationships.

Subsequently, the module prunes the DAG by randomly selecting a node with probability p and removing it along with its descendants. This enables the generation of diverse training samples with varying types and combinations of hallucinations for effective editing model training.

Category	Avg. Length [words]		Frequency [%]	
	Real	Synthetic	Real	Synthetic
Object	1.47	1.15	41.57	37.41
Fact	2.99	1.77	4.55	10.41
Text	1.79	1.50	3.21	7.84
Number	1.18	1.07	10.51	15.40
Relation	1.78	1.25	9.06	13.12
Attribute	1.32	1.01	31.10	15.81

Table 2. Comparison between real and synthetic samples in terms of span lengths and error frequencies. The average lengths and frequencies are generally similar across real and synthetic data.

5. VisionHall Dataset

We constructed VisionHall, a dataset specifically designed to train and evaluate the fine-grained hallucination detector and editor. Existing datasets for hallucination detection, such as AMBER [49] and M-HalDetect [20], are limited to coarse-grained labels (e.g. binary or ternary) and thus unsuitable for evaluating fine-grained detectors. Furthermore, these datasets (e.g. [11]) lack ground-truth refinements for hallucinated spans, making them unsuitable for editing tasks.

In contrast, a few datasets address fine-grained hallucination detection (e.g. [38]). However, they mainly focus on factual errors or unverifiable statements given world knowledge, and do not cover image-grounded hallucinations where the description deviates from the visual content.

To address these limitations, we constructed VisionHall, a dataset that enables comprehensive evaluation of fine-grained hallucination detection and editing. Each sample also includes the corresponding image and a human-written long reference.

Since human-written references offer a reliable basis for evaluation, reference-based evaluation is standard practice in natural language generation tasks [48, 53, 56]. Accordingly, we utilize the reference captions provided in the DCI dataset [47]. These captions are comprehensive, human-written descriptions that cover nearly all elements in each image.

As the DCI dataset provides only image–reference pairs, we collected MLLM-generated descriptions from twelve representative MLLMs and image captioning models: GPT-4o [1], Qwen2.5-VL 7B [7], Qwen2.5-VL 72B, LLaVA-NeXT [33], LLaVA-1.5 [32], MultimodalGPT [18], Qwen-VL-Chat [6], ShareGPT4V [10], InstructBLIP [13], InternVL [12], BLIP2 [28], and GIT [50]. We employed the official prompts for each model to generate these outputs.

Subsequently, human annotators labeled hallucinated spans in the generated descriptions according to our taxonomy. The annotation was conducted via a crowdsourcing platform, and responses exhibiting suspicious behavior were excluded to preserve data quality. Full annotation guidelines

Method	Detection	Editing		Overall	
	F_1	CLIP-S	PAC-S	BERT-F_1	CLIP-F_1
LLaVA-1.5-7B [32]	0.82	64.01	72.72	0.66	0.93
Qwen2-VL-7B [51]	3.36	64.79	73.01	3.62	4.98
LLaVA-OV-Qwen2-7B [26]	3.39	64.06	72.40	3.39	3.39
LLaVA-1.5-13B [32]	4.73	64.74	73.02	5.08	6.71
LLaVA-NeXT-Qwen-32B [33]	19.09	65.34	73.47	24.29	<u>31.06</u>
Llama-3.2-90B-Vision-Instruct [19]	16.92	65.28	73.54	14.56	17.62
Qwen2.5-VL-72B-Instruct [7]	21.31	64.38	72.99	18.85	23.67
LLaVA-OV-Qwen2-72B [26]	25.70	<u>65.74</u>	73.91	20.81	26.81
GPT-4o (w/o images) [1]	27.02	65.66	<u>73.99</u>	23.34	27.99
GPT-4o [1]	<u>29.37</u>	65.58	73.86	<u>24.89</u>	30.19
ZINA (Ours)	**45.15** (+15.8)	**66.08** (+0.34)	**74.36** (+0.37)	**44.02** (+19.1)	**50.39** (+20.2)

Table 3. Quantitative comparison with baseline methods on the VisionHall dataset. **Bold** font indicates the best, and <u>underlined</u> font indicates the second best. Our proposed methods outperformed the baselines in both tasks.

and interface are provided in the Appendix C.

For the detection task, VisionHall comprises 6,854 MLLM-generated descriptions for 4,759 images, collected from 211 annotators. For the editing task, our dataset contains 20k synthetic samples by our novel graph-based method, as detailed in Section 4.2. Further details on the VisionHall dataset are provided in Appendix C and F.

Table 2 shows the comparison between real and synthetic samples in terms of span lengths and error frequencies. For most categories, the span lengths between real and synthetic samples are generally comparable. The overall distributions are also similar across real and synthetic samples. Further details of comparisons between real and synthetic samples are provided in Appendix D.

6. Experiments

6.1. Setup

Datasets. To assess the practicality of the methods, it is essential to evaluate them on both in-domain and out-of-domain datasets. However, to the best of our knowledge, there is no publicly available dataset for our proposed tasks. Therefore, in addition to the VisionHall dataset, we employed the MHaluBench [11] dataset to evaluate performance on the coarse-grained hallucination detection task. This dataset provides samples with coarse-grained hallucination annotations in both Image-to-Text and Text-to-Image settings. We evaluated the baselines and our proposed method on the "Image-to-Text" subset of the MHaluBench dataset because our task focused on image captioning.

Baselines. We adopted the following MLLMs as baselines on the VisionHall dataset: LLaVA-1.5-7B [32], LLaVA-1.5-13B, Qwen2-VL-7B [51], LLaVA-OV-Qwen2-7B [26], LLaVA-NeXT-Qwen-32B [33], LLaVA-OV-Qwen2-72B, Llama-3.2-90B-Vision-Instruct [19], Qwen2.5-VL-72B-Instruct [7], and GPT-4o [1]. These models were selected as they are standard and representative MLLMs. A modified version of the FAVA prompt [38] with a 3-shot setting was used for all evaluations on VisionHall. For the evaluation on MHaluBench, we employed the same baseline models as Chen et al [11]. Implementation details and prompts are provided in Appendices E and H.

6.2. Results

Quantitative results. Table 3 presents a quantitative comparison with baseline methods on the VisionHall dataset. Our proposed method achieved an F_1 score of 45.15 in the detection task. Moreover, for the editing task, it achieved scores of 44.02, 50.39, 66.08, and 74.36 on BERT-F_1, CLIP-F_1, CLIP-S, and PAC-S, respectively. These results demonstrate that our method outperformed baseline models by 15.78 points on F_1, and by 19.14, 19.33, 0.34, and 0.37 points on BERT-F_1, CLIP-F_1, CLIP-S, and PAC-S, respectively.

Table 4 shows the quantitative results on the "Image-to-Text" subset of the MHaluBench dataset. Following Chen et al. [11], we reported the evaluation results at both the segment-level and the claim-level. As a result, our proposed method outperformed the baselines on 9 out of 10 metrics at the claim-level and on 8 out of 10 metrics at the segment-level. These results indicate that ZINA also demonstrated strong performance on out-of-domain data. Further quantitative results are provided in the Appendix.

Qualitative results. Fig. 3 shows examples from the VisionHall dataset. In the example on the left, the original description included hallucinations of the Object, Attribute, and Text types. GPT-4o failed to detect

	Model	Hallucinatory			Non-Hallucinatory			Average			
		P	R	F_1	P	R	F_1	Acc.	P	R	Mac.F_1
Claim-level	Gemini-based Self-Check ($n = 0$)	83.17	42.15	55.95	55.64	89.48	68.61	63.34	69.41	65.82	62.28
	Gemini-based Self-Check ($n = 2$)	84.24	66.75	74.48	67.35	84.60	75.00	74.74	75.80	75.68	74.74
	GPT-based Self-Check ($n = 0$)	84.78	80.07	82.35	61.64	69.01	65.12	76.56	73.21	74.54	73.73
	GPT-based Self-Check ($n = 2$)	86.54	85.13	85.83	69.05	71.48	70.24	80.80	77.80	78.30	78.04
	Gemini-based UniHD	84.44	72.44	77.98	71.08	83.54	76.80	77.41	77.76	77.99	77.39
	GPT-based UniHD	82.54	85.29	83.89	81.08	77.74	79.38	81.91	81.81	**81.52**	81.63
	ZINA (Ours)	**84.91**	**89.52**	**87.15**	**84.91**	**89.52**	**87.15**	**85.39**	**86.07**	80.28	**83.07**
Segment-level	Gemini-based Self-Check ($n = 0$)	89.30	47.71	62.19	43.76	**87.68**	58.38	60.38	66.53	67.69	60.29
	Gemini-based Self-Check ($n = 2$)	**90.44**	71.08	79.60	57.35	83.80	68.10	75.11	73.89	77.44	73.85
	GPT-based Self-Check ($n = 0$)	79.37	74.17	76.68	70.52	76.22	73.26	75.09	74.94	75.19	74.97
	GPT-based Self-Check ($n = 2$)	82.00	79.98	80.98	76.04	78.35	77.18	79.25	79.02	79.16	79.08
	Gemini-based UniHD	88.77	78.76	83.46	63.17	78.52	70.02	78.68	75.97	78.64	76.74
	GPT-based UniHD	87.03	91.01	88.98	78.52	70.77	74.44	84.60	82.77	80.89	81.71
	ZINA (Ours)	89.53	**93.47**	**91.46**	**84.38**	76.33	**80.15**	**88.06**	**86.95**	**84.90**	**85.80**

Table 4. Quantitative results on the "Image-to-Text" subset of the MHaluBench dataset. F_1 denotes the Micro-F_1 score, and Mac-F_1 represents the Macro-F_1 score. n denotes the number of examples used in the few-shot setting. **Bold** indicates the best and underlined indicates the second best. We employed the same baseline models as Chen et al [11].

ZINA: The image features the facade of the charming, vintage bookstore named "<text> ~~La Bibliothèque Librairie~~ **La Bibliothèque idéale Librairie** </text>". The exterior is painted a deep <attribute>~~blue~~ **green**</attribute>, and features elegant, cursive lettering in a beige color at the top … Two large windows display numerous <object>~~posters~~ **books**</object>, arranged neatly and aesthetically. Above each window, there is a decorative wrought iron balcony railing from an upper floor. The bookstore is situated on a street corner, marked as number 7. There's a small sign on the door that reads <text> ~~Éditions xeno~~ **Éditions aden**</text>. Additional stickers and signs are visible on the door and windows, indicating store information or promotions. Small lights are positioned above the windows to illuminate the storefront display.

GPT-4o: The image features the facade of the charming, vintage bookstore named **La Bibliothèque Librairie**. The exterior is painted a deep <attribute>~~blue~~ **green**</attribute>, and features elegant, cursive lettering in a beige color at the top … Two large windows display numerous **posters** arranged neatly and aesthetically. Above each window, there is a decorative wrought iron balcony railing from an upper floor. The bookstore is situated on a street **<relation>street</relation>**, marked as number 7. There's a small sign on the door that reads <text> ~~Éditions xeno~~ **Éditions aden**</text>. Additional stickers and signs are visible on the door and windows, indicating store information or promotions. Small **<object>books</object>** are positioned above the windows to illuminate the storefront display.

(a)

ZINA: The image captures a scene at a train station in <fact>~~Germany~~ **Italy**</fact>, where a row of red and silver ticket vending machines stands in a line. Each machine is adorned with a white sign that proudly displays "Self Service" <relation>~~in the middle~~ **on the right side**</relation>, indicating their purpose. There are <number>~~eight~~ **at least eight**</number> identical machines lined up from the foreground to the background. The floor beneath them is a light beige color, providing a contrast to the red and silver machines. The background is blurred, drawing focus to the ticket vending machines. The overall scene suggests a well-organized and modern train station.

GPT-4o: The image captures a scene at a train station in <fact>~~Germany~~ **Italy**</fact>, where a row of red and silver **<object>ticket</object> machines** stands in a line. Each machine is adorned with a white sign that proudly displays "Self Service" **in the middle**, indicating their purpose. There are <number>~~eight~~ **several**</number> identical machines lined up from the foreground to the background. The floor beneath them is a light beige color, providing a contrast to the red and silver machines. The background is blurred, drawing focus to the **ticket machines**. The overall scene suggests a well-organized and modern train station.

(b)

Figure 3. Qualitative results on the VisionHall dataset. Each subfigure shows the image and the edited descriptions generated by GPT-4o and ZINA. Edited spans are enclosed in tags; strikethrough text indicates the original hallucinated phrase, while **bold** text shows the suggested refinement. Red spans indicate errors such as incorrect tagging or missed detections.

scene text errors related to the names of the bookstores. GPT-4o also misclassified error types and failed to suggest refinements, as seen in the incorrectly tagged span "<relation>street</relation>". In contrast, ZINA correctly detected the hallucinated spans and successfully provided appropriate refinements.

In the example on the right, the original description contained hallucinations of the Fact, Relation, and Number types. GPT-4o failed to preserve the original phrasing—for example, modifying "ticket vending machine" to

Model	$\mathcal{M}_{\text{rev}}$	Backbone	n	Detection	Editing		Overall	
				F_1	CLIP-S	PAC-S	BERT-F_1	CLIP-F_1
(i)		Qwen2.5-VL-72B	3	21.91	63.32	71.69	15.54	17.88
(ii)	✓	Qwen2.5-VL-32B	3	32.55	61.20	72.63	27.52	34.66
(iii)	✓	LLaVA-OV-72B	3	34.41	65.78	74.02	31.39	36.10
(iv)	✓	Qwen2.5-VL-72B	1	43.25	65.18	73.30	42.53	49.54
(v)	✓	Qwen2.5-VL-72B	2	44.21	65.99	74.32	43.39	50.21
(vi)	✓	Qwen2.5-VL-72B	3	**45.15**	**44.02**	**50.39**	**66.08**	**74.36**

Table 5. Ablation study of the proposed method. $\mathcal{M}_{\text{rev}}$ indicates whether the reviewer LLM is used in addition to the detector, and n denotes the number of examples used in the few-shot setting.

"ticket machine", which can be problematic in practical applications. In contrast, our two-step strategy delegates token copying to a deterministic function, which guarantees correctness by construction, as shown in Fig. 3. This property offers a clear advantage in real-world scenarios where it is crucial to edit only the erroneous parts while preserving the rest of the text exactly as is.

6.3. Ablation Studies

Table 5 presents the quantitative results of the ablation studies. To assess the contribution of each module in our proposed method, we conducted three ablation studies.

Decoupling strategy ablation. To evaluate the contribution of our core strategy (i.e. decoupling detection and tagging), we replaced our pipeline with a single MLLM that performs both jointly. For fairness, Model(i) was fine-tuned with the same training budget as ZINA. As shown in Table 5, the comparison between Model (i) and Model (vi) reveals a substantial performance drop, with the F_1 score for the detection task decreasing by 23.24 points, partially due to exposure bias. Similarly, in the editing task, performance decreased by 28.48, 32.51, 2.76, and 2.67 points on BERT-F_1, CLIP-F_1, CLIP-S, and PAC-S, respectively. These results highlight the effectiveness of our decoupling strategy in improving both detection and editing performance.

Moreover, taken together with Table 3, these results indicate that the performance improvements mainly stem from the combination of synthetic data and the two-step generation strategy rather than either component in isolation. One-step systems (e.g. [38]) insert tags directly into the sentence; even a single misplaced tag can cause distributional shifts due to the autoregressive nature of LLMs. In contrast, as discussed in Section 4, the two-step strategy mitigates exposure bias and enables more effective use of the synthetic dataset, as evidenced by ZINA (Model (vi)).

Backbone ablation. We investigated the effect of different backbones by replacing the Qwen2.5-VL-72B backbone with Qwen2.5-VL-32B and LLaVA-OV-72B [26]. The latter was selected due to its strong performance and a model size comparable to Qwen2.5-VL-72B. Table 5 shows that Model (vi) outperformed both Model (ii), which uses the Qwen2.5-VL-32B backbone, and Model (iii), which uses the LLaVA-OV-72B backbone. Specifically, Model (vi) improved the F_1 score on the detection task by 12.60 points compared to Model (ii). Similarly, in the editing task, it also outperformed Model (ii) by 16.5, 15.73, 4.88, and 1.73 points on BERT-F_1, CLIP-F_1, CLIP-S, and PAC-S, respectively. These results demonstrate that the Qwen2.5-VL-72B backbone contributed to overall performance.

Few-shot ablation. Table 5 shows that Model (vi) outperformed Model (iv) with $n = 1$ and Model (v) with $n = 2$. Specifically, Model (vi) achieved improvements of 0.94, 0.63, 0.18, 0.09, and 0.04 in F_1, BERT-F_1, CLIP-F_1, CLIP-S, and PAC-S, respectively, compared to Model (v) with $n = 2$. These results indicate that the current few-shot setting ($n = 3$) had a substantial impact on overall performance.

7. Conclusion

We focused on the automatic fine-grained hallucination detection and editing for MLLMs. The contributions of this paper are as follows: (i) We proposed ZINA, a novel method for fine-grained hallucination detection and editing. (ii) We constructed the VisionHall dataset, which comprises outputs generated by twelve MLLMs, with hallucinated spans manually annotated according to our taxonomy. (iii) ZINA outperformed existing methods on VisionHall and MHaluBench.

Limitations. While our method achieved strong performance on both detection and editing tasks, it has several limitations. First, the two-stage architecture results in relatively high inference time, which may hinder deployment in real-time or resource-constrained settings. Second, despite outperforming baseline models overall, our method often underdetects hallucinations. This may stem from limitations in the reviewer MLLM, which tends to be overly conservative in generating tags.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Number 23K28168, JST Moonshot, and JSPS Fellows Grant Number JP25KJ2069.

References

[1] Josh Achiam, Steven Adler, et al. GPT-4 Technical Report. arXiv preprint arXiv:2303.08774, 2023. 1, 2, 5, 6, 3, 4

[2] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: Novel Object Captioning at Scale. In ICCV, pages 8948–8957, 2019. 5

[3] Shayan Ali Akbar, Md Mosharaf Hossain, Tess Wood, Si-Chi Chin, Erica M Salinas, Victor Alvarez, and Erwin Cornejo. HalluMeasure: Fine-grained hallucination measurement using chain-of-thought reasoning. In EMNLP, pages 15020–15037, 2024. 2

[4] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic propositional image caption evaluation. In ECCV, pages 382–398, 2016. 1, 2

[5] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. arXiv preprint arXiv:2310.11511, 2023. 4

[6] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities. In ICLR, 2024. 1, 5

[7] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923, 2025. 2, 4, 5, 6, 3

[8] Tianyi Bai, Yuxuan Fan, Qiu Jiantao, et al. Hallucination at a Glance: Controlled Visual Edits and Fine-Grained Multimodal Learning. In NeurIPS, 2025. 2

[9] Anthony Chen, Panupong Pasupat, Sameer Singh, Hongrae Lee, and Kelvin Guu. Purr: Efficiently editing language model hallucinations by denoising language model corruptions. arXiv preprint arXiv:2305.14908, 2023. 2

[10] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. ShareGPT4V: Improving Large Multi-Modal Models with Better Captions. In ECCV, pages 370–387, 2024. 5

[11] Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. Unified Hallucination Detection for Multimodal Large Language Models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3235–3252, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. 1, 2, 3, 5, 6, 7, 4

[12] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In CVPR, pages 24185–24198, 2024. 5

[13] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. InstructBLIP: Towards General-Purpose Vision-Language Models with Instruction Tuning. In NeurIPS, 2023. 1, 5

[14] Jacob Devlin, Ming-Wei Chang, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL, pages 4171–4186, 2019. 3

[15] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In Proceedings of the 32nd ACM International Conference on Multimedia, pages 11198–11201, 2024. 3

[16] Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. FactKB: Generalizable factuality evaluation using language models enhanced with factual knowledge. In EMNLP, 2023. 3

[17] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. RARR: Researching and revising what language models say, using language models. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2022. 2

[18] Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. MultiModal-GPT: A Vision and Language Model for Dialogue with Humans. arXiv preprint arXiv:2305.04790, 2023. 5

[19] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, pages arXiv–2407, 2024. 1, 2, 6, 3, 4

[20] Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. In AAAI, pages 18135–18143, 2024. 1, 2, 4, 5

[21] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654, 2020. 1

[22] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In EMNLP, pages 7514–7528, 2021. 3, 1, 5

[23] Shinnosuke Hirano, Yuiga Wada, Kazuki Matsuda, Seitaro Otsuki, and Komei Sugiura. LLM-Free Image Captioning Evaluation in Reference-Flexible Settings. arXiv preprint arXiv:2512.21582, 2025. 3

[24] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank Adaptation of Large Language Models. In ICLR, 2022. 3

[25] Minchan Kim, Minyeong Kim, Junik Bae, Suhwan Choi, Sungkyung Kim, and Buru Chang. Exploiting semantic reconstruction to mitigate hallucinations in vision-language models. In ECCV, pages 236–252, 2024. 2

[26] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy Visual Task Transfer. arXiv preprint arXiv:2408.03326, 2024. 6, 8, 3, 4

[27] Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. The dawn after the dark: An empirical study on factuality hallucination in large language models. arXiv preprint arXiv:2401.03205, 2024. 2

[28] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In ICML, pages 19730–19742, 2023. 5

[29] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating Object Hallucination in Large Vision-Language Models. In EMNLP, pages 292–305, 2023. 1, 2

[30] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On Pre-training for Visual Language Models. In CVPR, pages 26679–26689, 2024. 1

[31] Tsung Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft COCO: Common Objects in Context. In ECCV, pages 740–755, 2014. 5

[32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. In CVPR, pages 26296–26306, 2024. 1, 5, 6, 3, 4

[33] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, 2024. 5, 6, 3, 4

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In NeurIPS, pages 34892–34916, 2023. 1, 3

[35] Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023. 2

[36] Kazuki Matsuda et al. VELA: An LLM-Hybrid-as-a-Judge Approach for Evaluating Long Image Captions. In EMNLP, pages 8691–8707, 2025. 3, 1

[37] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In EMNLP, 2023. 2, 3

[38] Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. Fine-grained hallucination detection and editing for language models. In COLM, 2024. 1, 2, 3, 4, 5, 6, 8

[39] OpenAI. OpenAI o3 and o4-mini System Card, April 2025. 5

[40] Eunkyu Park, Minyeong Kim, and Gunhee Kim. Halloc: Token-level localization of hallucinations for vision language models. In CVPR, pages 29893–29903, 2025. 2, 4, 5

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models from Natural Language Supervision. In ICML, pages 8748–8763, 2021. 3, 1

[42] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. arXiv preprint arXiv:1809.02156, 2018. 2

[43] Sara Sarto, Manuele Barraco, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Positive-Augmented Contrastive Learning for Image and Video Captioning Evaluation. In CVPR, pages 6914–6924, 2023. 3, 1, 5

[44] Tal Schuster, Adam Fisch, and Regina Barzilay. Get your vitamin C! robust fact verification with contrastive evidence. In NAACL, 2021. 3

[45] Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, et al. FOIL it! Find One Mismatch Between Image and Language caption. In ACL, pages 255–265, 2017. 1

[46] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, and Amelia Glaese. Gemini: A Family of Highly Capable Multimodal Models. arXiv preprint arXiv:2312.11805, 2023. 1

[47] Jack Urbanek, Florian Bordes, Pietro Astolfi, Mary Williamson, Vasu Sharma, and Adriana Romero-Soriano. A Picture is Worth More Than 77 Text Tokens: Evaluating CLIP-Style Models on Dense Captions. In CVPR, pages 26700–26709, 2024. 3, 5

[48] Yuiga Wada, Kaneda Kanta, Saito Daichi, and Komei Sugiura. Polos: Multimodal Metric Learning from Human Feedback for Image Captioning. In CVPR, pages 13559–13568, 2024. 3, 5

[49] Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, et al. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. arXiv preprint arXiv:2311.07397, 2023. 1, 2, 5

[50] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. GIT: A Generative Image-to-text Transformer for Vision and Language. TMLR, 2022. 5

[51] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191, 2024. 6, 3, 4

[52] Ziwei Yao, Ruiping Wang, and Xilin Chen. HiFi-Score: Fine-Grained Image Description Evaluation with Hierarchical Parsing Graphs. In ECCV, pages 441–458, 2024. 1

[53] Weizhe Yuan, Graham Neubig, et al. BARTScore: Evaluating Generated Text as Text Generation. In NeurIPS, volume 34, pages 27263–27277, 2021. 5

[54] Bohan Zhai, Shijia Yang, Chenfeng Xu, Sheng Shen, Kurt Keutzer, Chunyuan Li, and Manling Li. HallE-Control: controlling object hallucination in large multimodal models. arXiv preprint arXiv:2310.01779, 2023. 2

[55] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-CLIP: Unlocking the Long-Text Capability of CLIP. In ECCV, pages 310–325, 2024. 5

[56] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In ICLR, 2020. 3, 5, 1

[57] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Mod-

els. In ACL (Volume 3: System Demonstrations), Bangkok, Thailand, 2024. ACL. 3

[58] Weihong Zhong, Xiaocheng Feng, Liang Zhao, Qiming Li, Lei Huang, Yuxuan Gu, Weitao Ma, Yuan Xu, and Bing Qin. Investigating and Mitigating the Multimodal Hallucination Snowballing in Large Vision-Language Models. In ACL, pages 11991–12011, 2024. 5