Holistic Coverage and Faithfulness Evaluation of Large Vision-Language Models

Anonymous ACL submission

Abstract

Large Vision Language Models (LVLMs) suffer from *hallucination* problems, wherein the models generate plausible-sounding but factually incorrect outputs, undermining their *reliability*. A comprehensive quantitative evaluation is necessary to identify and understand the extent of hallucinations in these models. However, existing benchmarks are often limited in scope, focusing mainly on object hallucinations. Furthermore, current evaluation methods struggle to effectively address the subtle semantic distinctions between model outputs and reference data, as well as the balance between hallucination and informativeness. To address these issues, we introduce a multi-dimensional benchmark covering objects, attributes, and relations, with challenging images selected based on associative bias. Moreover, we propose an LLM-based two-stage evaluation framework that generalizes the popular CHAIR metric (Rohrbach et al., 2018) and incorporates both faithfulness and coverage into the evaluation. We provide a detailed assessment of 10 established LVLMs within our framework and results demonstrate that we provide a more comprehensive and human-correlated evaluation than existing work. Through this work, we highlight the critical balance between *faithfulness* and *coverage* of model outputs, and we hope our work encourages future progress on addressing hallucinations in LVLMs while keeping their outputs informative.

1 Introduction

Large Vision Language Models (LVLMs) (Liu et al., 2023b; Ye et al., 2023a; OpenAI, 2023) have shown remarkable performance across a broad range of vision-language tasks. Despite the promising progress, the issue of hallucination has emerged as a critical concern. *Hallucination* refers to the generation of plausible sounding but inaccurate or fabricated textual descriptions for a given image, which can compromise the reliability and trustworthiness of the models.

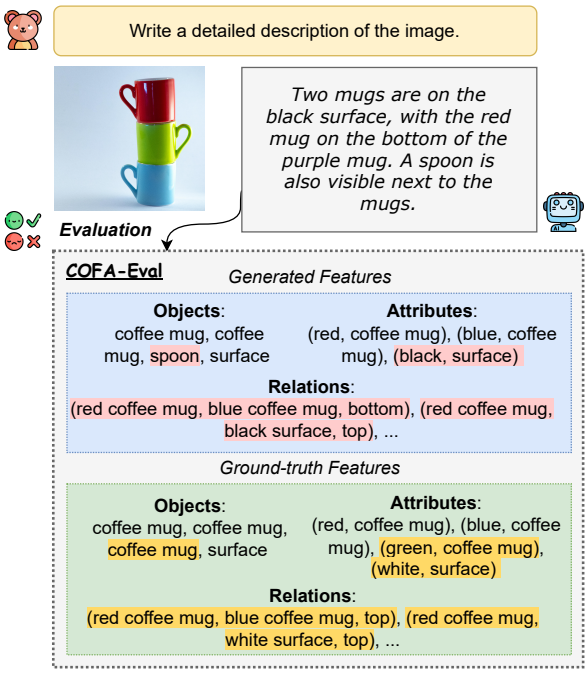

Figure 1: Example of the hallucination in open vocabulary generation task of LVLMs. Our proposed framework can identify objects, attributes, and relations from the generated captions and provide a more comprehensive evaluation of faithfulness and coverage. We highlight hallucinated features and uncovered features.

Recent studies have proposed various methods specifically designed to *evaluate* model hallucinations (Li et al., 2023b; Gunjal et al., 2023; Guan et al., 2023). These methods can be categorized into *discriminative* (Li et al., 2023b; Guan et al., 2023; Lovenia et al., 2023) and *generative* (Wang et al., 2023a; Zhai et al., 2023; Jing et al., 2023) ones, based on the type of tasks they evaluate. For example, POPE (Li et al., 2023b) evaluates LVLMs on discriminative visual question answering tasks, and HaELM (Wang et al., 2023a) employ an LLM to directly compare model outputs and references in a generation setting for hallucination evaluation. We summarize these evaluation methods in Table 1.

In contrast to prior studies that mainly focus

Evaluation Method	Hallucination Type			Human Annotation	Faithfulness	Coverage	Open Vocab. Generation
	Object	*Attribute*	*Relation*				
POPE	✓	✗	✗	✗	✓	✗	✗
HaELM	✓	?	?	✗	✓	✗	✓
HallusionBench	✓	?	?	✓	✓	✗	✗
Halle-Switch	✓	✗	✗	✗	✓	✓	✓
NOPE	✓	✗	✗	✗	✓	✗	✗
Bingo	?	?	?	?	✓	✗	✗
FaithScore	✓	✓	✓	✗	✓	✗	✓
AMBER	✓	✓	✓	✓	✓	✓	✗
MERLIM	✓	✗	✗	✗	✓	✗	✗
Ours (CoFa)	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of existing hallucination evaluation benchmarks for LVLMs, including POPE (Li et al., 2023b), HaELM (Wang et al., 2023a), HallusionBench (Guan et al., 2023), Halle-Switch (Zhai et al., 2023), NOPE (Lovenia et al., 2023), Bingo (Cui et al., 2023), FaithScore (Jing et al., 2023), AMBER (Wang et al., 2023b), MERLIM (Villa et al., 2023). **?** refers to features not explicitly mentioned in the paper. Open Vocab represents evaluating free-form generated captions without constraints to pre-defined vocabulary.

on *object* hallucination, we introduce a human-annotated multi-dimensional evaluation benchmark CoFa[1] by breaking down hallucination into three categories: *object* (existence), *attributes* (color and count), and *relations* (positional and comparative). In addition, to make the test cases challenging, we utilize the *associative bias* presented in training datasets and select images with only one component of a commonly co-occurred pair or group, which can potentially lead models to mistakenly generate associated elements that are not present. Our experimental findings validate the effectiveness of this methodology in exposing the susceptibility of current LVLMs to such biases.

In addition to constructing the benchmark, we also propose a new evaluation framework. Existing frameworks for evaluating *open-ended* caption generation tasks, such as the widely used CHAIR (Rohrbach et al., 2018) metric, exhibit notable constraints. First, these methods rely on a predefined vocabulary, limiting their ability to identify hallucinations in an *open vocabulary* setting where understanding semantic nuances in model outputs and references–such as synonyms and variations–is important. Additionally, focusing exclusively on hallucination evaluation can favor precise but uninformative model outputs, overlooking the aspect of coverage. To address these shortcomings, we propose an evaluation framework CoFa-Eval that generalizes CHAIR and incorporates an LLM in a two-stage design, enhancing its capability to evaluate open vocabulary captions across object, attribute, and relation dimensions while also considering caption completeness.

We conduct comprehensive evaluations on 10 established LVLMs across multiple dimensions with CoFa-Eval. Our findings reveal that LVLMs tend to prioritize precision over coverage, leading to predictions with high accuracy but limited scope. This observation underscores the need for the community to focus on achieving an *balance* between faithfulness and coverage in LVLMs.

Our contributions are highlighted below:

- We introduce CoFa, a comprehensive human-annotated dataset covering three dimensions with challenging images selected based on co-occurring statistics.
- We propose an LLM-based evaluation framework CoFa-Eval that generalizes previous methods and can handle object, attribute, and relation evaluation in open vocabulary generation settings.
- We conduct experiments on 10 mainstream LVLMs on CoFa, focusing on the balance between faithfulness and coverage score. We notice that even GPT-4V(ision) (OpenAI, 2023) still suffers from hallucination, achieving a relatively low faithfulness score despite covering more information within an image than other models.

2 Existing LVLM Hallucination Evaluation Benchmarks

As shown in Table 1, existing studies (Li et al., 2023b; Wang et al., 2023a; Zhai et al., 2023; Lovenia et al., 2023; Villa et al., 2023) have primarily focused on *object-level* hallucination, with only a few recent studies (Jing et al., 2023; Wang et al., 2023b) recognizing the importance of extending

[1] CoFa is the acronym of coverage and faithfulness.

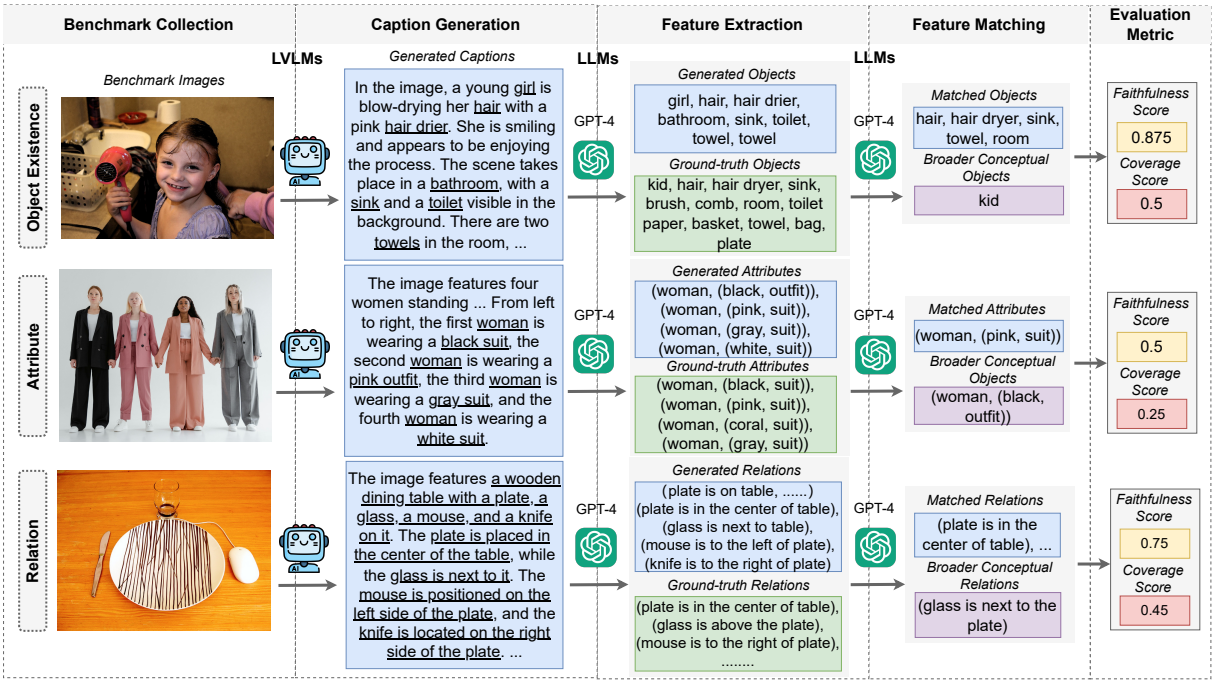

Figure 2: Overview of our proposed evaluation framework CoFa-Eval: (1) Firstly, LVLMs generate captions from benchmark images. (2) Following this, LLMs are employed to extract pivotal features that encapsulate from the generated descriptions. (3) Subsequently, these features are aligned with a pre-defined list of ground-truth features using LLMs, facilitating the creation of two essential outputs: a dictionary of matched features and a more extensive dictionary encompassing broader conceptual matches. (4) Finally, we calculate two key metrics: *faithfulness* and *coverage*. These metrics measure the LVLMs' comprehension by evaluating how well the generated captions encapsulate the salient features of the images and the breadth of concepts they cover, respectively.

hallucinations to other dimensions. Our benchmark CoFa covers hallucination evaluations of *objects*, *attributes*, and *relations*, and we further detail attributes to color and counting, and relations to positional and comparative, to provide a comprehensive and fine-grained evaluation benchmark.

Regarding benchmark annotations, many existing benchmarks employ different ways of annotating the evaluation datasets *automatically*. For example, Li et al. (2023b) employ object detectors to identify all objects in an image; Zhai et al. (2023) employ GPT-4V(ision) to generate ground-truth annotations. There are also approaches to developing models specifically for automatic evaluation, thereby bypassing the need for benchmark collections process (Wang et al., 2023a; Gunjal et al., 2023). Given the challenges and potential inaccuracies associated with automated models, our study opts to annotate the evaluation dataset *manually* to ensure the annotation accuracy and encompass the distinct categories of hallucinations.

Additionally, most existing benchmarks focus exclusively on hallucination evaluation, which can favor precise but uninformative model outputs, overlooking the aspect of coverage. To address the issue, we incorporate coverage scores in our evaluation. We note that two relevant concurrent works (Wang et al., 2023b; Zhai et al., 2023) also include the coverage scores. However, compared with our work, they are either limited in scope, focusing only on objects or simple attributes and relations, or are unable to be adopted in open-vocabulary generation settings. Besides, along with the benchmark, we propose an evaluation metric that generalizes their adopted CHAIR metric.

3 CoFa Benchmark

In this section, we outline the methodology employed to create CoFa benchmark aimed at assessing the hallucination issues of LVLMs. The construction of this benchmark involves two main stages: the *collection* of images (§3.1) and their subsequent *annotation* (§3.2).

3.1 Image Collection

We aim to select images that can maximally expose the model hallucination issue. When models are repeatedly exposed to specific combinations of features – such as object existence, object at-

tributes, and object relations – during training, they develop a pronounced *associative bias*, which leads the models to expect these co-occurring features in similar situations. Consequently, when a model is presented with an image that contains only one element of a familiar combination, it may erroneously infer the presence of the associated feature. This associative bias is one primary source of model hallucinations (Li et al., 2023b). To utilize this feature for exposing the model hallucination problem, we initially analyze the co-occurrence statistics of *object-object*, *object-attribute*, and *object-relation-object* combinations within the extensively annotated GQA (Hudson and Manning, 2019) dataset. Subsequently, we curate a collection of images that represent *frequently* and *infrequently* co-occur object-object, object-attribute, object-relation-object pairs and triples. This way, we identify the most challenging images to construct a benchmark, to which we then add detailed human annotations for a nuanced evaluation.

3.1.1 Definition

We first define three principal features for assessing hallucination issues in LVLMs. **Object existence** (object-object) includes all visual entities within an image, including both *foreground* and *background* elements. **Attribute** (object-attribute) refers to the characteristics of objects within an image, with a particular focus on *color* and *counting* in our work. We categorize this analysis into two segments: *objects* and *people*. For objects, we focus on the color and count of each non-people-related item. For people, we emphasize the attire colors and the total number of individuals depicted. **Relation** (object-relation-object) refers to the relational information between the objects in the image, and we focus on *positional* and *comparative* relation in this paper. Specifically, the former tests the relative position between the objects, and the latter analyzes the understanding of "which object is larger than the other."

3.1.2 Quantifying Co-Occurring Features

To utilize co-occurring features, the first step involves calculating the *statistical dependencies* between different features. This analysis helps identify prevalent co-occurrence patterns in the data, highlighting strongly associated features that the model may have internalized. We utilize two kinds of statistics to obtain the dependencies – *frequencies* and *conditional probabilities*.

Frequency. By counting the number of times a particular feature appears in conjunction with specific objects, attributes, or relations, this metric offers insights into the raw distribution of features across the dataset.

Conditional Probability. To delve deeper, we calculate the conditional probability, which quantifies the likelihood of encountering a specific feature given the presence of an object:

$$P(\text{feature}|\text{object}) = \frac{\text{frequency}(\text{feature, object})}{\text{frequency}(\text{object})}, \quad (1)$$

where feature $\in$ {object, attribute, relation}. To identify objects with the most significant skew in their conditional probability distributions, we explore five different kinds of metrics based on conditional probabilities, such as the maximum conditional probabilities. The definitions and explanations of these five metrics can be found in Appendix B.

3.1.3 Utilizing Co-Occurrence Statistics for Image Extraction

Leveraging the identified co-occurrence statistics, we systematically extract images from existing datasets. The process includes several critical steps:

- Identify objects (**O**) that exhibit the *most pronounced* co-occurrence dependencies, including frequency and conditional probabilities.
- Select features (**I**) that are *minimally associated* with each identified object, thereby spotlighting instances where common co-occurrences are absent.
- Determine features (**H**) that are *most frequently* co-occurring with each object, serving as strong associative tendencies.
- Collect images for each feature in **I** corresponding to an object in **O**, with the chosen images *including* the specified feature and object, yet *excluding* any features from **H**, to create clear cases for testing the model's associative bias.

For each feature defined in §3.1.1, we follow the above steps to extract images from the GQA dataset. We then manually check the collected images to ensure we keep only those of high quality with clear annotations. These processes allow us to gather a set of images to evaluate the object

Category	# Images	Source
Object Existence	50	GQA
Attribute: Color & Count (Object)	27	Pixel
Attribute: Color & Count (People)	34	Pixel
Relation: Positional	50	GQA
Relation: Comparative	40	GQA

Table 2: The data statistics of CoFa.

existence and the relations. However, extracting images that accurately represent specific *attributes* proved challenging due to limited attribute annotation in GQA. To address this, we retrieve source images that are free from copyright issues from the Internet, guided by the attribute-related statistics we had collected in the previous step. The statistics of our proposed benchmark are detailed in Table 2.

3.2 Annotation

For each image within the distinct feature subsets, we manually annotate them, adhering to the definition we have discussed in §3.1.1. Figure 2 shows three examples from our collected benchmark.

Object Existence. In an image with "a lady is sitting on a bench in front of a building," the extracted objects are "lady", "bench", and "building".

Attribute. For *objects*, in an image with "two green apples on a white table," the identified attributes are "(green, apple)" for each apple and "(white, table)" for the table. For *people*, in a scene showing "a woman wearing a red jacket," the identified attribute is "(woman, (red, jacket))".

Relation. For *positional* relation, an example can be "the bed is to the left of the table" as the positional relation between "bed" and "table". Similarly, the inverse statement "the table is to the right of the bed" is also considered accurate and annotated in our benchmark. Furthermore, we include annotations such as "a bed is on the left side of the image" to capture the positional relationships of objects at the image level. For *comparative relation*, we adopted an annotation with numerical rank based on object size, ordering objects from largest to smallest (e.g. "1. bed, 2. table, 3. cup").

4 Evaluation Framework

The CHAIR metric is widely adopted in existing studies (Wang et al., 2023a; Zhai et al., 2023; Wang et al., 2023b) yet is inflexible because of its reliance on a pre-defined vocabulary. In this work, we propose a framework CoFa-Eval that generalizes CHAIR and incorporates both the *faithfulness* and *coverage* aspects into the evaluation. As shown in Figure 2, our evaluation process has three steps: caption generation (§4.1), feature extraction and matching (§4.2), and evaluation metrics (§4.3).

4.1 Caption Generation

CoFa provides a set of triples (I, p_G, F_G), where I denotes the image, p_G represents the prompt designed for LVLMs generation, and F_G is the feature annotations of the image. The designed prompts are shown in Appendix C for each subset.

language hallucination

4.2 Feature Extraction and Matching

We start the process by generating an initial response, denoted as R, using a specific LVLM with the input pair (I, p_G). Following this, we leverage an LLM to analyze R and extract key features. This is achieved through a series of prompts p_E, outlined in Appendix D, which are designed to extract features from object existence, attributes, and relations, respectively, resulting in a comprehensive lists of extracted features from R, denoted as $F_R = \{f_{R_1}, f_{R_2}, ..., f_{R_m}\}$. Next, we utilize an LLM to align the extracted features list F_R with a pre-annotated ground-truth features list $F_G = \{f_{G_1}, f_{G_2}, ..., f_{G_m}\}$. This alignment is facilitated through a set of carefully crafted prompts p_M, outlined in Appendix E, tailored to each feature subset, aiming to identify correlations and correspondences. Unlike previous evaluation metrics that rely on a fixed feature list and direct mapping, our approach eschews pre-processing and instead utilizes LLMs' nuanced language comprehension capabilities to semantically match extracted features with their ground-truth counterparts. This process yields two key outputs: **matched features** dictionary (D_M) and **broader conceptual matches** dictionary (D_B).

D_M contains features $f_{R_{i'_m}}$ from f_R that semantically aligned with the features $f_{G_{i_m}}$ from F_G, ensuring *precision*. For example, if we have the extracted "(plaid, shirts)" and the candidate ground-truth feature is "(checkered, shirt)," we can establish a match between these two because "plaid" and "checkered" are conceptually similar patterns often used interchangeably in the context of textiles.

D_B includes features $f_{R_{j'_n}}$ from f_R that have broader conceptual meanings than the features $f_{G_{j_n}}$ from F_G, adding *conceptual depth* to the evaluation. For instance, if we have the extracted "(red,

Model	Object		Attribute				Relation				Average Faithful. Score (%)	Average Cover. Score (%)
	Existence		*Color & Counting*				*Positional*		*Comparative*			
			Object		People							
	Faithful$_\uparrow$	Cover$_\uparrow$	Faithful$_\uparrow$	Cover$_\uparrow$	Faithful$_\uparrow$	Cover$_\uparrow$	Faithful$_\uparrow$	Cover$_\uparrow$	Faithful$_\uparrow$	Cover$_\uparrow$		
InstructBLIP	74.5	24.8	72.0	23.9	47.1	9.3	50.0	13.6	66.9	35.6	62.1	21.44
LLaVA-1.5	72.1	24.7	74.6	37.8	43.3	12.1	64.8	14.9	51.9	40.1	61.34	25.92
MiniGPT-4 v2	65.0	25.4	64.5	17.9	38.9	11.6	38.8	33.1	44.7	11.2	50.38	19.84
mPLUG-Owl2	71.5	24.8	79.9	32.7	39.7	16.2	45.2	10.8	41.6	30.6	55.58	23.02
BLIVA	77.7	21.9	73.3	24.3	37.6	11.6	39.5	9.7	68.0	29.9	59.22	19.48
CogVLM	71.2	35.5	75.3	24.3	43.7	22.4	51.9	10.5	49.0	35.9	58.22	25.72
InternLM-XComposer2	82.5	23.9	75.8	26.3	50.4	13.8	62.6	11.1	64.1	38.4	67.08	22.7
Qwen-VL-Chat	70.6	28.4	75.1	38.6	38.8	16.0	56.9	8.5	51.9	24.3	58.66	23.16
Emu2	94.2	14.1	66.7	10.4	54.3	1.9	72.2	1.8	87.5	12.3	74.98	8.1
GPT-4V	61.6	38.8	78.5	36.3	34.7	23.8	46.7	12.6	51.6*	28.5*	54.62	28.0

Table 3: The overall evaluation results of object, attribute, and relation hallucination in benchName using GPT-4 as the LLM Agent. The highest is highlighted in blue, while the worst performance is highlighted in yellow. Faithfulness and coverage scores are in percentage (%). For images that contain people, GPT-4V refrains from generating comments and we marked this score with an asterisk (∗).

clothes)" from an image, and the ground-truth annotation is "(red, dress)," we can still consider these features to match. This is because "clothes" is a broader category that encompasses "dress." Therefore, despite the slight difference in specificity, the extracted features can be aligned with the ground-truth annotations based on their semantic relationship, where "dress" is a sub-type of "clothes."

4.3 Evaluation Metrics

We introduce two metrics to evaluate the hallucinations in two dimensions: *faithfulness* and *coverage*.

Faithfulness. In the context of image captioning, faithfulness measures how closely captions match an image's content, emphasizing *accuracy* in depicting visual elements and their attributes and relations without introducing hallucinations. It is calculated by comparing generated features against actual image features, considering both *direct* (D_M) and *broader* conceptual similarities (D_B):

$$\text{Faithfulness}(R, F_G) = \frac{|D_M \cup set(D_B)|}{|F_R|} \in [0, 1]. \quad (2)$$

Coverage. It measures the *comprehensiveness* of the generated captions in capturing the key elements and attributes depicted in the image. It evaluates the proportion of ground-truth features that are successfully captured in the generated response, only through *direct* matches (D_M):

$$\text{Coverage}(R, F_G) = \frac{|set(D_M)|}{|F_G|} \in [0, 1]. \quad (3)$$

5 Experiment

In this section, we perform experiments to evaluate different existing LVLMs within our framework (§5.1), demonstrate our evaluation method correlates with human evaluation (§5.2), and verify the importance of each of our designs (§5.3).

5.1 Model Coverage-Faithfulness Evaluation

We use the framework CoFa-Eval to evaluate various LVLMs listed in Table 7, employing GPT-4 as the evaluation LLM agent.

In the evaluation of various models as shown in Table 3, Emu2 distinguishes itself by achieving the highest average faithfulness score of 87.5, signifying its consistent capability to generate responses that accurately reflect the content of the input image. However, Emu2's performance in terms of coverage is less impressive, with the lowest average score of 8.1, suggesting that its responses, while accurate, may not comprehensively cover all elements of the image. When broken down into specific dimensions, Emu2 excels in faithfulness across categories—scoring 94.2 in object existence, 54.3 in attribute-people, 72.2 in relation-positional, and 87.5 in relation-comparative. Conversely, it lags in coverage, with scores of 14.1 in object existence, 10.4 in attribute-object, 1.9 in attribute-people, and 1.8 in relation-positional. These results point to a potential trade-off between faithfulness and coverage in Emu2's design, where the model prioritizes accuracy at the expense of a broader scope in its responses. This pattern supports the initial hypothesis that some LVLMs may intention-

Category	Faithful. (ρ)	Cover (ρ)
Object Existence	0.91	0.89
Attribute: Color & Count (Object)	0.99	0.98
Attribute: Color & Count (People)	0.98	0.96
Relation: Positional	0.78	0.86
Relation: Comparative	0.92	0.98

Table 4: Pearson correlation ρ between our framework COFA-EVAL and human evaluation.

Model	InstructBLIP	LLaVA-1.5	GPT-4V
Evaluation data: randomly selected			
Faithfulness	76.5	84.5	64.1
Coverage	24.3	26.3	41.2
Evaluation data: co-occurrence selected (**Ours**)			
Faithfulness	74.5 (-2.0)	72.1 (-12.4)	61.6 (-2.5)
Coverage	24.8 (+0.5)	24.7 (-1.6)	38.8 (-2.4)

Table 5: Model performance comparison on our data selection method against random selection. Faithfulness and coverage scores are in percentage (%).

ally sacrifice coverage to improve the precision of their outputs.

Meanwhile, GPT-4V distinguishes itself with an unparalleled average coverage score of 28.0, showcasing its adeptness in encapsulating a wide array of features from the input image. This indicates that GPT-4V excels in recognizing and addressing diverse elements within images, although it does not necessarily always maintain the highest accuracy, as seen in its lower faithfulness score of 61.6. Particularly in evaluations concerning the existence of objects, GPT-4V leads with the highest coverage score of 38.8, underlining its comprehensive approach to object detection. This approach tends to favor inclusivity, which might lead to the occasional identification of objects that are not present in the image. Furthermore, in evaluations focused on attributes related to people, GPT-4V again achieves the highest coverage score. However, this comes with a trade-off, as it also exhibits a higher tendency towards hallucinations compared to other models, indicating a propensity to generate details or elements that may not be grounded in the actual content of the image.

Models such as LLaVA-1.5, CogVLM, and InternLM-XComposer2 showcase a more equitable performance, achieving respectable scores in both faithfulness and coverage metrics. This highlights their capability to provide responses that are not only precise but also encompassing. Notably, LLaVA-1.5 stands out for its remarkable outcomes, achieved through the efficient use of training data, underscoring the significance of leveraging high-quality instruction-tuning data to enhance model performance.

5.2 Effectiveness of Evaluation Framework

To demonstrate the effectiveness and reliability of our automatic evaluation pipeline, we conduct experiments to evaluate if our evaluation framework correlates with human evaluations in both faithfulness and coverage. Specifically, we have human and our GPT-4-based evaluation method evaluate InstructBLIP outputs and compute the Pearson correlation (ρ) score[2]. As shown in Table 4, for object evaluation, the findings reveal a significantly strong Pearson correlation of 0.91 for faithfulness and 0.89 for coverage, effectively rejecting the null hypothesis that posits no correlation between the two evaluation methodologies, with a compelling p-value of 0. Additionally, our study achieved a notably high correlation of 0.98 in both attribute recognition and comparative relations. When evaluating positional relations, which tend to involve longer and more complex descriptions, the correlation scores were not as high as those observed in the other categories but still indicated a very high level of correlation with 0.78 in faithfulness and 0.86 in coverage. These results affirm the comparability of our automatic evaluation metrics to human evaluation in terms of both *efficacy* and *reliability*.

5.3 Ablation Study

In this section, we serve to answer two questions and discuss our findings.

1. **How does our co-occurrence data selection method compare to other alternatives?**

To illustrate the effectiveness of the co-occurrence method, we set up a baseline of randomly selecting 50 images in the GQA val set and applied human annotation, the same as for our dataset. For the ablation study, we focus on the well-studied object hallucination. We evaluate three popular models representing query tokens-based image features, linear projection-based features, and advanced commercial LVLMs. As shown in Table 5, all models tend to produce more hallucinations and exhibit significantly lower faithfulness compared to our benchmark. Notably, LLaVA-1.5 scores 12.4 points lower in faithfulness

[2]We opt for Pearson correlation as our assessment metric due to its suitability for measuring *linear* relationships, as opposed to Spearman's rank correlation, which is more attuned to *monotonic* relationships.

when evaluated against our benchmark. Additionally, the coverage scores for both LLaVA-1.5 and GPT-4V decreased. Upon further analysis through human review, we discover that our benchmark, on average, contains 1.69 more objects than images selected at random. This finding helps explain the observed phenomenon.

2. **How does our LLM-based evaluation framework compare with LLM-free evaluation?**

We compare our proposed LLM agent augmented framework against the original CHAIR metric which is adopted by all previous studies. Because the CHAIR metric is limited to evaluating only 80 objects from the MSCOCO dataset, for a fair comparison, we randomly select 20 COCO images and re-annotate them for analysis alongside the CHAIR metric. We have made these annotations publicly available, adhering to the same list of synonyms used in the original CHAIR metric. To conduct this comparison, we utilize two accuracy scores. For Acc (F), we assess the performance by comparing the number of hallucinated objects identified by the metric against the ground-truth hallucinated objects in the caption. If an object is incorrectly identified as hallucinated when it is not, the metric imposes a penalty of -1. This score aligns with the *matching* phase of our framework, ensuring a thorough evaluation of hallucination detection accuracy. For Acc (C), we calculate the number of objects detected by metric over the unique objects mentioned in the caption, assessing our *extraction* phase's efficiency. As shown in Table 6, our framework significantly outperforms in both faithfulness and coverage accuracy by a large amount. This improvement is due to our framework's *open vocabulary matching* ability, unlike the original chair approach that struggles with new expressions without pre-defined synonyms. Notably, with complex models like GPT-4V, CHAIR's faithfulness accuracy drops to 5.88, highlighting our method's strength in managing diverse object descriptions.

Moreover, the limitation of CHAIR's pre-defined object list extends to its inability to account for potential hallucinated objects, which are essential for differentiating between mere words and actual objects in captions. This leads to its failure in detecting hallucinated objects, resulting in performance degradation. In contrast, our method overcomes this by using an automatically extracted object list that dynamically matches objects, avoiding this limitation. Although approaches like Wang et al. (2023b) attempt to address this by including a selection of potential hallucinated objects, they cannot guarantee coverage of all possible hallucinated objects, particularly in complex outputs from advanced LVLMs that generate extensive captions.

Metric	F.$_\uparrow$	C.$_\uparrow$	Acc (F)$_\uparrow$	Acc (C)$_\uparrow$
Model: InstructBLIP				
CHAIR	75.0	34.3	11.11	80.66
CHAIR$_{\text{LLM}}$ **(Ours)**	76.9	30.4	88.89 (+77.78)	100.0 (+19.34)
Model: LLaVA-1.5				
CHAIR	74.3	34.1	30.00	83.52
CHAIR$_{\text{LLM}}$ **(Ours)**	81.5	27.0	90.00 (+60.00)	97.08 (+13.56)
Model: GPT4V				
CHAIR	79.3	54.8	5.88	82.35
CHAIR$_{\text{LLM}}$ **(Ours)**	69.7	57.9	82.35 (+76.47)	98.17 (+15.82)

Table 6: Comparison of LLM-augmented CHAIR with original CHAIR metric. Here, F. and C. denote faithfulness and coverage scores in percentage (%). Acc (F) represents the average percentage of hallucinated objects detected by the metric. Acc (C) denotes the average percentage of objects detected by metric.

6 Conclusion

We introduce a comprehensive multi-dimensional benchmark and a generalized evaluation framework tailored for LVLMs, focusing on addressing hallucination in generative tasks. Our newly developed benchmark, named CoFa, categorizes hallucinations into three distinct types, providing a nuanced approach to understanding model errors. Moreover, we propose a two-stage evaluation pipeline CoFa-Eval incorporating an LLM agent, offering a holistic framework that measures both the faithfulness and coverage of captions generated by LVLMs. This methodology significantly improves upon existing benchmarks by overcoming challenges associated with open vocabulary, semantic similarity, and the nuanced evaluation of attributes and relations, thereby offering a more granular and comprehensive assessment of image captioning tasks. Our experimental findings underscore the ongoing challenges within the domain, revealing that even state-of-the-art models such as GPT4-V(ision) still have high rates of hallucination. This research underscores the imperative for continuous advancements in LVLM evaluation techniques and establishes a new benchmark for future endeavors aimed at reducing hallucination and bolstering the reliability of content generated by LVLMs.

7 Ethical Considerations

Our work investigates the phenomenon of hallucination in outputs generated by LVLMs. Here, we outline the primary ethical considerations associated with our study. In developing our evaluation framework, we employed GPT-4 for feature extraction and matching tasks to evaluate the model's hallucination. Consequently, we recognize that any biases inherent to the GPT-4 model will likely influence the results observed in our benchmark. Furthermore, our data collection efforts encompassed datasets from GQA and images sourced from the internet (specifically Pixel[3]). We acknowledge and adhere to the pertinent policies and requirements governing data sharing and utilization within our benchmark.

8 Limitations

Our humanly annotated benchmark, CoFa, provides a more comprehensive and detailed evaluation than previous works in objects, attributes, and relations. This dataset is humanly curated to cover a broad spectrum of hallucination phenomena, focusing on object existence, color and count attributes, and positional and comparative relations. Despite the extensive coverage, it is essential to acknowledge that we did not fully address the entire range of possible attributes and relations that could be subject to hallucination in LVLMs. Although not covered in our current benchmark, additional elements are equally crucial for a holistic understanding and assessment of LVLMs. Further, we employ a single prompt for evaluating LVLM performance. This approach raises the possibility that some models may not be adequately trained to follow these instructions as intended or require refined prompt engineering to achieve optimal performance.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. *ArXiv preprint*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *ArXiv preprint*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *ArXiv preprint*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90% chatgpt quality. *ArXiv preprint*.

Chenhang Cui, Yiyang Zhou, Xinyu Yang, Shirley Wu, Linjun Zhang, James Zou, and Huaxiu Yao. 2023. Holistic analysis of hallucination in gpt-4v(ision): Bias and interference challenges. *ArXiv preprint*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *ArXiv preprint*.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. 2024. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *ArXiv preprint*.

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. 2023. Hallusionbench: An advanced diagnostic suite for entangled language hallucination & visual illusion in large vision-language models. *ArXiv preprint*.

Anish Gunjal, Jihan Yin, and Erhan Bas. 2023. Detecting and preventing hallucinations in large vision language models. *ArXiv preprint*.

Wenbo Hu, Yifan Xu, Yi Li, Weiyue Li, Zeyuan Chen, and Zhuowen Tu. 2023. Bliva: A simple multimodal llm for better handling of text-rich visual questions. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: a new dataset for compositional question answering over real-world images. *ArXiv preprint*.

Liqiang Jing, Ruosen Li, Yunmo Chen, Mengzhao Jia, and Xinya Du. 2023. Faithscore: Evaluating hallucinations in large vision-language models. *ArXiv preprint*.

[3] https://www.pexels.com/

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proc. of ICML*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proc. of ICML*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Proc. of EMNLP*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *ArXiv preprint*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *ArXiv preprint*.

Holy Lovenia, Wenliang Dai, Samuel Cahyawijaya, Ziwei Ji, and Pascale Fung. 2023. Negative object presence evaluation (nope) to measure object hallucination in vision-language models. *ArXiv preprint*.

OpenAI. 2023. Gpt-4 technical report.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proc. of EMNLP*.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. 2023. Generative multimodal models are in-context learners. *ArXiv preprint*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv preprint*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*.

Andrés Villa, Juan Carlos Le'on Alc'azar, Alvaro Soto, and Bernard Ghanem. 2023. Behind the magic, merlim: Multi-modal evaluation benchmark for large image-language models. *ArXiv preprint*.

Junyan Wang, Yi Zhou, Guohai Xu, Pengcheng Shi, Chenlin Zhao, Haiyang Xu, Qinghao Ye, Mingshi Yan, Ji Zhang, Jihua Zhu, Jitao Sang, and Haoyu Tang. 2023a. Evaluation and analysis of hallucination in large vision-language models. *ArXiv preprint*.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Ming Yan, Ji Zhang, and Jitao Sang. 2023b. An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *ArXiv preprint*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. Cogvlm: Visual expert for pretrained language models. *ArXiv preprint*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chaoya Jiang, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023a. mplug-owl: Modularization empowers large language models with multimodality. *ArXiv preprint*.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Mingshi Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023b. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *ArXiv preprint*.

Bohan Zhai, Shijia Yang, Xiangchen Zhao, Chenfeng Xu, Sheng Shen, Dongdi Zhao, Kurt Keutzer, Manling Li, Tan Yan, and Xiangjun Fan. 2023. Halle-switch: Rethinking and controlling object existence hallucinations in large vision language models for detailed caption. *ArXiv preprint*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *ArXiv preprint*.

A Large Vision-Language Models

The recent advancements in large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023a,b; Chiang et al., 2023; Bai et al., 2023) have sparked a wave of research focused on enhancing vision-language pre-trained models (VLPMs) (Kim et al., 2021; Alayrac et al., 2022; Li et al., 2023a). By

incorporating the versatile capabilities of LLMs, these studies aim to improve the language understanding and generation abilities of VLPMs significantly. In this paper, we refer to the enhanced VLPMs with the integration of LLMs as *Large Vision-Language Models* (LVLMs) (Li et al., 2023b). LVLMs excel in comprehending both the visual semantics of objects in images and the linguistic semantics associated with these objects by leveraging the extensive parametric knowledge embedded in the LLMs. This dual understanding enables LVLMs to conduct intricate reasoning about the concepts related to these objects. Consequently, LVLMs demonstrate strong performance in various traditional multi-modal tasks, such as visual question answering, image captioning, and object detection, highlighting their versatility and robustness in these domains (Liu et al., 2023b; Zhu et al., 2023; Ye et al., 2023a; Dai et al., 2023; Liu et al., 2023a; Hu et al., 2023; OpenAI, 2023).

B Conditional Probabilities

1. $P(\text{feature}|\text{object})_{\text{max}}$: maximum conditional probability, highlighting the strongest feature-object associations.

2. $P(\text{feature}|\text{object})_{\text{avg}}$: average conditional probability, offering a broad view of how features tend to cluster around objects.

3. $P(\text{feature}|\text{object})_{\text{max}} - P(\text{feature}|\text{object})_{\text{avg}}$: the difference between the maximum and average conditional probabilities, revealing objects with outlier features.

4. $P(\text{feature}|\text{object})_{\text{avg}} - P(\text{feature}|\text{object})_{\text{min}}$: the spread between average and minimum conditional probabilities, indicating the range of commonality among features.

5. $P(\text{feature}|\text{object})_{\text{max}} - P(\text{feature}|\text{object})_{\text{min}}$: the range between maximum and minimum conditional probabilities, capturing the full spectrum of feature variability.

C Captions Generation Prompts

1. Object Existence: Write a detailed description of the image. Provide information about all objects in front and background.

2. Object attribute: Write a detailed description of the image. Provide information about the total number and colors of all objects from left to right and up to bottom.

3. People attribute: Write a detailed description of the image. Provide information about the total number of people and colors of clothes for each person from left to right.

4. Object positional relation: Describe the *positional relationship* between all the objects in the image in detail, using left, right, top, and bottom etc, from the view of the observer.

5. Object comparative relation: Rank the size of all the objects in the image in detail, from large to small.

D Features Extraction Prompts

The feature extraction prompts for objects, color and counting attributes, positional relation and comparative relation are illustrated in Table 8, Table 9, Table 10, Table 11, and Table 12, respectively.

E Features Matching Prompts

The features matching prompts for objects, color and counting attributes, positional relation and comparative relation are illustrated in Table 13, Table 15, Table 14,Table 16, and Table 17 separately.

F Comparative Relation Evaluation

For comparative relation matching, instead of directly matching the ranked relation, we employ the LLM agent to match the extracted objects names from generated captions to the name of annotated objects. Through this way, we obtained a rank orders of the annotated objects since the names are matched. For example if LVLMs' final results is "1.bed, 2.cup, 3.chair", where our annotation is "1.bed, 2.table, 3.chair 4.cup". After matching by LLM agent, the original results is now "1.bed, 4.cup, 3.chair" according to our annotation. Finally, we implemented a simple efficient algorithm with dynamic programming called find_longest_monotonic_subsequence to get the number of correct predictions. Here the answer can be either "[1,4]" or "[1,3]" which means only 2 objects' comparative relations are correctly predicted. This approach guaranteed 100% accuracy when LLM agents matched the correct objects.

Model	Visual Encoder	Alignment Network	Language Model
InstructBLIP	EVA CLIP ViT-G/14$_{1.1B}$	Q-Former	Vicuna$_{7B}$
LLaVA-1.5	CLIP ViT-L/14-336px$_{0.4B}$	MLP	Vicuna-v1.5$_{13B}$
MiniGPT-v2	EVA CLIP ViT-G/14$_{1.1B}$	Linear Projection	LLaMA-2$_{7B}$
mPLUG-Owl2	CLIP ViT-L/14$_{0.4B}$	Cross Attention	LLaMA-2$_{7B}$
BLIVA	EVA CLIP ViT-G/14$_{1.1B}$	Q-Former & Linear Projection	Vicuna$_{7B}$
CogVLM	EVA2-CLIP-E/14$_{4.7B}$	MLP	Vicuna-v1.5$_{7B}$
InternLM-Xcomposer2	CLIP ViT-L/14-336px$_{0.4B}$	Partial Low-Rank Adaptation	InternLM2$_{7B}$
Qwen-VL	CLIP ViT-G/14$_{1.9B}$	Cross Attention	QwenLM$_{13B}$
Emu2	EVA2-CLIP-E-plus/14$_{5.0B}$	Linear Projection	LLaMA$_{33B}$
GPT-4(V)	Unknown	Unknown	GPT-4

Table 7: Architectures of mainstream LVLMs evaluated in our benchmark. InstructBLIP (Dai et al., 2023), LLaVA-1.5 (Liu et al., 2023a), MiniGPT-v2 (Chen et al., 2023), mPLUG-Owl2 (Ye et al., 2023b), BLIVA (Hu et al., 2023), CogVLM (Wang et al., 2024), InternLM-XComposer2 (Dong et al., 2024), Qwen-VL (Bai et al., 2023), Emu2 (Sun et al., 2023) and GPT-4V (OpenAI, 2023).

System message
You are a language assistant that helps to extract information from given sentences.

Prompt
Given an image with a caption that is generated by a vision language model.
Please act as a linguistic master and extract all the objects from the captions and format your response in a JSON format with the key being "objects" and the value being a list of objects.
Please only extract objects without including attributes. For example, extract "field" instead of "grassy field". Also be mindful of plural forms. For example, extract "cow" instead of "cows". Please only extract the object that is a concrete entity in the real world instead of abstract concepts, actions, and moves.
It cannot be an abstract notion such as day, time, scene, moment, image, game, sport, setting, plot, atmosphere, surroundings, group etc.
It cannot be any words describing the emotions such as excitement, enthusiasm, etc.
It cannot be any words describing the positions in the image, such as foreground, background, left, right, etc.
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the objects based on the factual information in the caption.
Here is the caption: {Input Caption}

Table 8: Prompt template for extracting objects. {In-context examples} are in-context examples. {Input caption} are captions generated by evaluated models.

System message
You are a language assistant that helps to extract information from given sentences.

Prompt
Given an image with a caption that is generated by a vision language model.
Please act as a linguistic master and extract the total number and colors of all objects as mentioned in the captions. Your answer should be a dict of this format: {"total num of objects": "(NUM, OBJECT)", "objects": {"ORDER": "(ATTRIBUTE, OBJECT)"}}. Remember OBJECT should be in singular format.
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the objects and attributes based on the factual information in the caption.
Here is the caption: {Input Caption}

Table 9: Prompt template for extracting attributes (object). {In-context examples} are in-context examples. {Input caption} are captions generated by evaluated models.

System message
You are a language assistant that helps to extract information from given sentences.

Prompt
Given an image with a caption that is generated by a vision language model.
Please act as a linguistic master and extract the total number of people and colors of clothes for each person as mentioned in the captions. Your answer should be a dict of this format: {"total num of people": "(NUM, PERSON)", "clothes": {"ORDER": "person": "PERSON", "object": "(ATTRIBUTE, OBJECT)", "action": "ACTION"}}. OBJECT can be clothes or accessories (e.g., bags, socks).
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the objects and attributes based on the factual information in the caption.
Here is the caption: {Input Caption}

Table 10: Prompt template for extracting attributes (people). {In-context examples} are in-context examples. {Input caption} are captions generated by evaluated models.

System message
You are a language assistant that helps to extract information from given sentences.

Prompt
Given an image with a caption that is generated by a vision language model.
Please act as a linguistic master and extract a set of words describing the spatial or positional relations between all the visual objects from the captions. Your answer should be a list of values that are in format of object1 relation with object2 with the relation being left, right, top, bottom, middle etc. Do not extract the attribute along with the object and don't extract any relation that is an verb, replace it with simply which object is (on or to the left or etc) the other object or the image. Formulate your response into a JSON object with the key being "relations" and the value being a list of relations. If there are no relations found, please return an empty list.
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the relations based on the information in the caption.
Here is the caption: {Input Caption}

Table 11: Prompt template for extracting positional relations. {In-context examples} are in-context examples. {Input caption} are captions generated by evaluated models.

System message
You are a language assistant that helps to extract ranking from given sentences.

Prompt
Given an image with a caption that is generated by a vision language model.
Given an image with a caption that is generated by a vision language model. Please act as a linguistic master and extract the rank of all the objects from large to small as mentioned in the captions. Your answer should be a dict of values which the keys represent the ranks starting from 1 and values are the No.1 largest object to smallest. If the caption does not mention the order of the object, you can by default view the order of objects appearance as from largest to smallest. If there are no objects mentioned in the caption, you can return an empty dict.
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the relations based on the information in the caption.
Here is the caption: {Input Caption}

Table 12: Prompt template for extracting comparative relations. {In-context examples} are in-context examples. {Input caption} are captions generated by evaluated models.

System message
You are given a task to match objects from two lists that have the same meaning.

Prompt
Input Lists:
1. "gt-objects": Ground truth objects in the image.
2. "generated-objects": Objects identified by a vision language model.
For each object in "generated-objects", find the object in the "gt-objects" that have the same meaning and add it to the "matched-objects" dictionary. By the same meaning, we mean the words can be synonyms, can be plural/singular forms of each other and can also have different length of words to express the same meaning of objects, etc. Note since we find the matched object for each object in "generated-objects", it's ok that multiple objects in "generated-objects" match one object in "gt-objects", list all matches. There is special scenario that when you can't find the matched object in "gt-objects" but you can find one or more object is a subset or a sub category of the generated object, which means that the generated object is a broader concept of the object in "gt-objects", add it to the "broader-concept" dictionary instead of the "matched-objects". If there are many objects are a subset or a sub category of the generated object, you can pick anyone of them. Note we are matching for each object in "generated-objects". If you can find the matched object in "gt-objects", you should not add it to the "broader-concept" dictionary.
Output:
- A "broader-concept" dictionary: only if an object from "generated-objects" denotes a broader category of a concept in "gt-objects". Key = word from "generated-objects", Value = word from "gt-objects".
- A "matched-objects" dictionary: Key = word from "generated-objects", Value = word from "gt-objects". It should not contain any words from the "broader-concept" dictionary.
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the broader-concept, and matched-objects from the following two objects lists.
- gt-objects: {Input Ground Truth Objects}
- generated-objects: {Input Generated Objects}
Please note that your answer should only be in a JSON format with the keys only being the broader-concept, and matched-objects.

Table 13: Prompt template for matching objects in image caption and reference caption. {In-context examples} are in-context examples. {Input Ground Truth Objects} are the ground truth objects list {Input Generated Objects} are the extracted objects list from the extraction step which are originally captions generated by evaluated models.

System message
You are given a task to match (attributes, objects) from two lists that have the same meaning.

Prompt
Inputs:
1. “gt-att-obj”: A dictionary with order being the key and the ground-truth (attribute, object) pair being the value. The order is the order of the object in the image. Sometimes one object can be, for example “(black, bag), (white, bag), (striped, bag)”, it means either “black” or “white” or "striped" is correct for an attribute related with the “bag” and should be matched.
2. “generated-att-obj”: A dictionary with order being the key and the generated (attribute, object) pair being the value. The order is the order of the object in the generated caption.
Matching Criteria:
- For each (attribute, object) in “generated-att-obj”, find the (attribute, object) in the “gt-att-obj” that have the same meaning and add it to the “matched-att-obj” dictionary.
- By the same meaning, we mean the words can be synonyms, can be plural/singular forms of each other and can also have different length of words to express the same meaning of attributes or objects, etc.
- If you find that the “generated-att-obj” can be matched with the “gt-att-obj” but the attribute or object in “generated-att-obj” is a broader concept of the attribute or object in “gt-att-obj”, for example, one object in “generated-att-obj” is “person”, but the “gt-att-obj” don’t have “person” but specifically have “man”, which is a subcategory of “person”, add it to the “broader-concept” dictionary instead of the “matched-att-obj”.
Output:
- A “broader-concept” dictionary: {“ORDER2”: {“(ATTRIBUTE1, OBJECT1)”: “(ATTRIBUTE2, OBJECT2)”}} only if an (ATTRIBUTE1, OBJECT1) with ORDER1 from “generated-att-obj” denotes a broader category of an (ATTRIBUTE2, OBJECT2) with ORDER2 in “gt-att-obj”. Notify that Key must be the (ATTRIBUTE1, OBJECT1)from “generated-att-obj”, Value must be (ATTRIBUTE2, OBJECT2) from “gt-att-obj”. If none, it should be an empty dictionary. ORDER1 should be the same as ORDER2.
- A “matched-att-obj” dictionary: {“ORDER2”: {“(ATTRIBUTE1, OBJECT1)”: “(ATTRIBUTE2, OBJECT2)”}} only if an (ATTRIBUTE1, OBJECT1) with ORDER1 from “generated-att-obj” can be mapped to an (ATTRIBUTE2, OBJECT2) with ORDER2 in “gt-att-obj” with the matching criteria. Key must be (ATTRIBUTE1, OBJECT1) from “generated-att-obj”, Value must be (ATTRIBUTE2, OBJECT2) from “gt-att-obj”. It should not contain any (ATTRIBUTE1, OBJECT1) or (ATTRIBUTE2, OBJECT2) from the “broader-concept” dictionary. ORDER1 should be the same as ORDER2.
- The keys in “broader-concept” and “matched-att-obj” must be the same as “gt-att-obj".
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the broader-concept, and matched-objects from the following two objects lists.
- gt-objects: {Input Ground Truth Attributes}
- generated-objects: {Input Generated Attributes}
Please note that your answer should only be in a JSON format with the keys only being the broader-concept, and matched-objects.

Table 14: Prompt template for matching attributes (object) in image caption and reference caption. {In-context examples} are in-context examples. {Input Ground Truth Attributes} are the ground truth attribute list {Input Generated Attributes} are the extracted attributes list from the extraction step which are originally captions generated by evaluated models.

System message
You are given a task to match (attributes, objects) from two lists that have the same meaning.

Prompt
Inputs:
1. “gt-att-obj”: A dictionary with order being the key and the ground-truth (attribute, object) pair being the value. The order is the order of the object in the image. Sometimes one object can be, for example “(black, bag), (white, bag), (striped, bag)”, it means either “black” or “white” or "striped" is correct for an attribute related with the “bag” and should be matched.
2. “generated-att-obj”: A dictionary with order being the key and the generated (attribute, object) pair being the value. The order is the order of the object in the generated caption.
Matching Criteria:
- For each (attribute, object) in “generated-att-obj”, find the (attribute, object) in the “gt-att-obj” that have the same meaning and add it to the “matched-att-obj” dictionary.
- By the same meaning, we mean the words can be synonyms, can be plural/singular forms of each other and can also have different length of words to express the same meaning of attributes or objects, etc.
- If you find that the “generated-att-obj” can be matched with the “gt-att-obj” but the attribute or object in “generated-att-obj” is a broader concept of the attribute or object in “gt-att-obj”, for example, one object in “generated-att-obj” is “person”, but the “gt-att-obj” don’t have “person” but specifically have “man”, which is a subcategory of “person”, add it to the “broader-concept” dictionary instead of the “matched-att-obj”.
Output:
- A “broader-concept” dictionary: {“ORDER2”: {“(ATTRIBUTE1, OBJECT1)”: “(ATTRIBUTE2, OBJECT2)”}} only if an (ATTRIBUTE1, OBJECT1) with ORDER1 from “generated-att-obj” denotes a broader category of an (ATTRIBUTE2, OBJECT2) with ORDER2 in “gt-att-obj”. Notify that Key must be the (ATTRIBUTE1, OBJECT1)from “generated-att-obj”, Value must be (ATTRIBUTE2, OBJECT2) from “gt-att-obj”. If none, it should be an empty dictionary. ORDER1 should be the same as ORDER2.
- A “matched-att-obj” dictionary: {“ORDER2”: {“(ATTRIBUTE1, OBJECT1)”: “(ATTRIBUTE2, OBJECT2)”}} only if an (ATTRIBUTE1, OBJECT1) with ORDER1 from “generated-att-obj” can be mapped to an (ATTRIBUTE2, OBJECT2) with ORDER2 in “gt-att-obj” with the matching criteria. Key must be (ATTRIBUTE1, OBJECT1) from “generated-att-obj”, Value must be (ATTRIBUTE2, OBJECT2) from “gt-att-obj”. It should not contain any (ATTRIBUTE1, OBJECT1) or (ATTRIBUTE2, OBJECT2) from the “broader-concept” dictionary. ORDER1 should be the same as ORDER2.
- The keys in “broader-concept” and “matched-att-obj” must be the same as “gt-att-obj".
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the broader-concept, and matched-objects from the following two objects lists.
- gt-objects: {Input Ground Truth Attributes}
- generated-objects: {Input Generated Attributes}
Please note that your answer should only be in a JSON format with the keys only being the broader-concept, and matched-objects.

Table 15: Prompt template for matching attributes (people) in image caption and reference caption. {In-context examples} are in-context examples. {Input Ground Truth Attributes} are the ground truth attribute list {Input Generated Attributes} are the extracted attributes list from the extraction step which are originally captions generated by evaluated models.

System message
You are given a task to match (object-1 positional relation with object-2) from a ground truth dictionary and a list based on their meaning.

Prompt
Inputs:
1. "gt-relations": A dictionary of ground truth relations. Each key is a number with no meaning of order. Each key represents different relations. The values is a list of one or two relations, if there are two relations, they are synonyms. Sometimes in one relation it contains for example "image / table", it means either image or table in this phrase is correct.
2. "generated-relations": A list of generated relations from a model.
Matching Criteria:
- For each relation in "generated-relations", find the corresponding relation in "gt-relations" based on their meaning, if there is none, skip it.
- If you find a match, add it to the "matched-relations" dictionary. Note that if there are two relations in a item of "gt-relations", it means the same meaning of the relation, you can pick either one of them as the match to the relation in "generated-relations".
- If you find that the generated relation is a broader concept of a relation in "gt-relations" such as the generated relation is near each other, next to, in touch etc.
but the gt-relation specifically have their relation is specifically left, right, behind or front, etc, which is more than near, add it to the "broader-concept" dictionary.
Output:
- A "broader-concept" dictionary: only if an relation from "generated-relations" denotes a broader category of a concept in "gt-relations" Notify that Key must be the item from "generated-relations", Value must be item from "gt-relations". If none, it should be an empty dictionary.
- A "matched-relations" dictionary: only if an relation from "generated-relations can be mapped to an relation in "gt-relations" with the matching criteria. Key must be word from "generated-relations", Value must be word from "gt-relations". It should not contain any words from the "broader-concept" dictionary.
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the broader-concept, and matched-relations from the following two inputs.
- gt-relations: {Input Ground Truth Relations}
- generated-relations: {Input Generated Relations}
Please note that your answer should only be in a JSON format with the keys only being the broader-concept, and matched-objects.

Table 16: Prompt template for matching positional relations in image caption and reference caption. {In-context examples} are in-context examples. {Input Ground Truth Relations} are the ground truth relation list {Input Generated Relations} are the extracted relation list from the extraction step which are originally captions generated by evaluated models.

System message
You are given a task to match the correct objects with the same meaning from a ground truth dictionary and a generated dictionary.

Prompt
Inputs:
Inputs: 1. "gt-objects": A dictionary of ground truth objects. Each key is a number starting rank No. 1 and increment each time by 1. Each value is the corresponding object with the rank. Sometimes one object can be, for example "ground / court, it means either ground or court is correct and should be matched.
2. "generated-objects": A dictionary with rank being the key and the object being the value. The rank is the rank of the object in the generated caption.
Instructions:
1. Matching Criteria: - For each object in "generated-objects", find the object in the "gt-objects" that have the same meaning and add it to the "matched-objects" dictionary.
- By the same meaning, we mean the words can be synonyms, can be plural/singular forms of each other and can also have different length of words to express the same meaning of objects, etc.
- Notice that the final matched-objects must follow the order of values in "generated-objects".
- If you find that the "generated-objects" can be matched with the "gt-objects" but the object in "generated-objects" is a broader concept of the objects in "gt-objects", for example, one object in "generated-objects" is "person", but the "gt-objects" don't have "person" but specifically have "man", which is a subcategory of "person", add it to the "broader-concept" dictionary instead of the "matched-objects".
2. Output:
- A "broader-concept" dictionary: only if an object from "generated-objects" denotes a broader category of a concept in "gt-objects" Notify that Key must be the item from "generated-objects", Value must be item from "gt-objects". If none, it should be an empty dictionary.
- A "matched-objects" dictionary: only if an object from "generated-objects" can be mapped to an object in "gt-objects" with the matching criteria. Key must be word from "generated-objects", Value must be word from "gt-objects". It should not contain any words from the "broader-concept" dictionary.
For clarity, consider these examples:
{In-context examples}
————————-
With these examples in mind, please help me extract the broader-concept, and matched-relations from the following two inputs.
- gt-relations: {Input Ground Truth Relations}
- generated-relations: {Input Generated Relations}
Please note that your answer should only be in a JSON format with the keys only being the broader-concept, and matched-objects.

Table 17: Prompt template for matching comparative relations in image caption and reference caption. {In-context examples} are in-context examples. {Input Ground Truth Relations} are the ground truth objects ranking list {Input Generated Relations} are the extracted objects list from the extraction step which are originally captions generated by evaluated models.