Evaluation and Enhancement of Semantic Grounding in Large Vision-Language Models

Jiaying Lu*,†, Jinmeng Rao*, Kezhen Chen*, Xiaoyuan Guo*, Yawen Zhang*, Baochen Sun*, Carl Yang*, Jie Yang*

*Mineral, †Emory University

{jinmengrao, kezhenchen, xiaoyuanguo, yawenz, baochens, yangjie}@mineral.ai
*{jiaying.lu, j.carlyang}@emory.edu

Abstract

Large Vision-Language Models (LVLMs) offer remarkable benefits for a variety of vision-language tasks. However, a challenge hindering their application in real-world scenarios, particularly regarding safety, robustness, and reliability, is their constrained semantic grounding ability, which pertains to connecting language to the physical-world entities or concepts referenced in images. Therefore, a crucial need arises for a comprehensive study to assess the semantic grounding ability of widely used LVLMs. Despite the significance, sufficient investigation in this direction is currently lacking. Our work bridges this gap by designing a pipeline for generating large-scale evaluation datasets covering fine-grained semantic information, such as color, number, material, etc., along with a thorough assessment of seven popular LVLMs' semantic grounding ability. Results highlight prevalent misgrounding across various aspects and degrees. To address this issue, we propose a data-centric enhancement method that aims to improve LVLMs' semantic grounding ability through multimodal instruction tuning on fine-grained conversations. Experiments on enhanced LVLMs demonstrate notable improvements in addressing misgrounding issues.

Introduction

Large Vision-Language Models (LVLMs) (Zhao et al. 2023b; Tang et al. 2023; OpenAI 2023; Yang et al. 2023) expand the powerful large language models (Ouyang et al. 2022; Touvron et al. 2023; Ling et al. 2023) to versatile general-purpose vision-language understanding and generation interfaces. This is achieved through the integration of the vision encoder and the autoregressive large language model (Liu et al. 2023a; Li et al. 2023a; Luo et al. 2023). While demonstrating promising performance in solving various vision-language benchmarks, comprehensive examination and analysis are desired before deploying LVLMs into real-world critical-sensitive applications. Recent studies reveal that many LVLMs still suffer from text-image misalignment (Yarom et al. 2023), adversary perturbed input (Zhao et al. 2023a), and object hallucination (Li et al. 2023c) even in some seemingly simple cases. In this paper, we specifically focus on the evaluation of the under-explored semantic grounding ability in LVLMs.

Semantic grounding (i.e., the capability to connect words to the physical-world entities or concepts they refer to) (Yun, Sun, and Pavlick 2021; Li et al. 2022) is critical to the safe, robust, and reliable development of LVLMs. Although identifying “a Siberian tiger” as “a tabby cat” in the user-shared image under a social chatbot setting seems harmless, stakes escalate significantly when an LVLM-aided disease diagnosis assistant interprets instruction to analyze the patient’s “left lung” as “right lung”. In this study, we comprehensively evaluate the semantic grounding ability of existing LVLMs through our proposed evaluation suite. Specifically, we automatically generate 9,000 vision-language test samples, exploring semantic grounding proficiency across four formats and addressing six types of grounding targets. Seven state-of-the-art LVLMs are evaluated using the 9,000 test samples, and the experimental results reveal that most of them exhibit semantic grounding deficiency across various aspects and degrees. To facilitate the scalability of test sample generation and the interpretability of evaluation metrics, we propose to adopt multiple-choice questions as the test format. Further technical details are presented in the Evaluation of Semantic Grounding Section.

Moreover, we introduce a data-centric enhancement method designed to enhance the semantic grounding capabilities of LVLMs. Diverging from classical model-centric approaches (Zha et al. 2023) that emphasize advancements in model architecture, our approach focuses on curating a substantial volume of diverse multimodal, multi-round conversation data that can be leveraged by any LVLM. In total, we curate 180,000 fine-grained instructional instances for the purpose of semantic grounding enhancement. Experimental results with LVLMs fine-tuned on our enhancement data consistently demonstrate improvements in multimodal semantic grounding.

Related Work

Trustworthiness Evaluation of LVLMs. Many works leverage existing vision-language datasets to derive trustworthiness evaluation benchmarks for LVLMs. MME (Fu et al. 2023) consists of 14 subtasks based on public images with manually constructed annotations, which measure both perception and cognition abilities of LVLMs in the...
form of Yes-or-No question answering. The LAMM benchmark (Yin et al. 2023) covers nine common 2D image tasks and three common point cloud tasks with specifically curated inference instruction. Other similar benchmarks include LVLM-eHub (Xu et al. 2023), MM-Vet (Yu et al. 2023), and MMBench (Liu et al. 2023b), etc. There also exist benchmarks focusing on evaluating specific properties of LVLMs. POPE (Li et al. 2023c) focuses on evaluating object hallucination as sentence-level classification, using human-annotated labels. (Zhao et al. 2023a) proposes evaluating the robustness of LVLMs by adding adversarial noise into input images. Our work also provides a valuable resource to serve as a comprehensive trustworthiness benchmark for LVLMs, from a novel perspective focusing on semantic grounding. Moreover, we provide a generic framework to enable researchers to conveniently create test samples for evaluating LVLMs.

**Methods for Building Safe, Robust, and Reliable LVLMs.** There are two streams of approaches to improve the safeness, robustness, and responsibilities of LLMs and LVLMs (Ji et al. 2023): model-centric approaches and data-centric approaches. Model-centric approaches often focus on the model advancements, which involve (1) designing robust training paradigms (Berg et al. 2022; Dong et al. 2023), (2) robust inference (Wang et al. 2022; Zhang et al. 2023), (3) refining generated response (Madaan et al. 2023), etc. Data-centric approaches (Zha et al. 2023; Bai et al. 2024), on the other hand, focus on ensuring data quality and reliability, which often involve (1) faithful training data development such as data collection (Liu et al. 2023a; Lu et al. 2023) and data cleaning (Northcutt, Jiang, and Chuang 2021; Monarch and Munro 2021), (2) inference stage data augmentation such as retrieving supporting knowledge (Chen et al. 2022; Cui et al. 2023). Our framework follows the data-centric idea. Instead of modifying the model structure, our work aims to improve the semantic grounding ability of LVLMs via multimodal instruction tuning, which has proved to be a generic and efficient approach for improving LVLMs.

**Preliminaries**

We illustrate one representative architecture of LVLMs in Figure 1, which typically consists of the following components:

1. a visual encoder that extracts features from the input image;
2. a projector that projects visual features to the language embedding space;
3. a tokenizer that tokenizes textual input into textual tokens and maps them into language embedding space;
4. a decoder-only LLM (e.g., LLaMA) that generates textual responses based on the multimodal inputs.

The overall output generation process of a LVLM \( F_\Theta \) can be formally described by

\[
Y = F_\Theta (X^v, X^q),
\]

where \( \Theta \) are parameters of the LVLM, \( X^v \) denotes the visual input, \( X^q \) denotes the textual input, and \( Y \) denotes the generated output sequence.

Specifically, given a visual input \( X^v \), a visual encoder \( f_\phi \) firstly extracts visual features by

\[
Z^v = f_\phi (X^v),
\]

where \( Z^v \in \mathbb{R}^{l_v \times d} \), and \( \phi \) are parameters of \( f_\phi \) typically frozen in LVLMs training. Here \( l_v \) denotes the length of visual tokens, and \( d \) is the dimension of visual tokens. Regarding the visual encoder \( f_\phi \), most existing LVLMs employ ViT-based structures (Dosovitskiy et al. 2020) and select certain layers outputs to construct a certain length of visual tokens. For instance, LLaVA (Liu et al. 2023a) utilizes grid features before and after the last Transformer layer of ViT, while LaVIN (Luo et al. 2023) utilizes the \([CLS]\) embedding from every fourth layer of ViT.

A trainable projector, denoted as \( g_\omega \), is then applied to convert \( Z^v \) into language embedding space, which can be defined by

\[
H^v = g_\omega (Z^v) = g_\omega (f_\phi (X^v)),
\]

where \( H^v \in \mathbb{R}^{l_v \times h} \), and \( \omega \) are trainable parameters of \( g_\omega \). \( h \) denotes the dimension of language embedding space. In practice, an efficient implementation of \( g_\omega \) can be a projection matrix \( W \in \mathbb{R}^{d \times h} \) (Liu et al. 2023a), thus \( H^v = Z^v W \).

More sophisticated projectors are also proposed, such as Q-former (Li et al. 2023b).

Given the input textual query \( X^q \), a tokenizer \( k_\psi \) is employed to tokenize and map them into “textual tokens” by

\[
H^q = k_\psi (X^q),
\]

where \( H^q \in \mathbb{R}^{l_q \times h} \), and \( \psi \) are trainable parameters of \( k_\psi \). It is worth noting that \( H^q \) and projected \( H^v \) have the same dimensionality \( \mathbb{R}^h \) as the input of the large language model \( p_\theta \). Similarly, the length of textual tokens varies depending on the choice of tokenizer, as words are chunked into subwords. Moreover, special tokens \([s], [/s]\) can be added to indicate the span of visual tokens and textual tokens. Therefore, we denote the process of LVLM to prepare the multimodal input of its LLM component as:

\[
H = [H^v, H^q] \in \mathbb{R}^{(l_v+l_q) \times h}.
\]
Once the multimodal input features are processed, the LLM component of an LVLM is responsible for generating responses. The LLM component is essentially a probabilistic auto-regressive model $p_{\theta}$ with trainable parameters $\theta$ that can predict the next token $y_t$ step by step based on the input $H$ and tokens predicted so far $Y_{0:t-1}$. The process can be formulated by:

$$Pr(Y_{0:t} | H) = \prod_{t=1}^{T} p_{\theta}(y_t | H, Y_{0:t-1}).$$  \hspace{1cm} (6)

where $Pr$ denotes the probability of generating the output sequence $Y_{0:T}$.

**Evaluation of Semantic Grounding**

A wide range of forms can be used for the evaluation of semantic grounding in LVLMs, and each form has its own advantages and disadvantages. Free-form questions (Yarom et al. 2023) are easy to design but require resource-intensive human evaluations and are difficult to score consistently. Similarity-based evaluation uses less resources, but is heavily reliant on high-quality ground truth responses and the bias of similarity metrics. Yes-or-No questions (Fu et al. 2023) are non-ambiguous and easier to evaluate. However, they may be too easy and cannot capture all aspects of semantic grounding in LVLMs.

We propose **Msg-MCQ** for the novel Multimodal Semantic Grounding evaluation based on Multiple-Choice Questions (MCQs) (Lu et al. 2022a,b). This form presents a question along with a set of predetermined choices, allowing respondents to select the option they believe to be correct. The MCQs facilitate efficient grading and analysis of responses, and the difficulty level can be controlled by adjusting the number and the feasibility of distractor choices (Lu et al. 2022a,b). Moreover, the Yes-or-No form can be regarded as a special case in MCQs where the choices are “(A) Yes, (B) No.”. We use accuracy as the evaluation metric for MCQs. Following the notation in the Preliminaries section, we extend the textual input question $X^t$ into two parts: the question body $X^{qb}$ and the multiple choices $X^{qc}$. Therefore, we formally define the task of MSG-MCQ:

**Definition 1 (Msg-MCQ)** Given an input image $X^s$, input question body $X^{qb}$, and $K$ input choices $X^{qc} = \{C^1, C^2, \ldots, C^K\}$, **Msg-MCQ** expects a LVLM $F_\theta$ to select the correct choice $C^i$ where $1 \leq i \leq K$ from the input choices $X^{qc}$.

In the scope of LVLM, the selection action is typically determined by the generated sequence of LVLM. Depending on the specific implementation of LVLMs, the output can be as simple as a choice indicator like “C”, or a comprehensive response that includes both the selected choice and an explanation of selecting it such as “The correct choice is A, because . . .”.

**MSG-MCQ Generation Pipeline**

In this work, we propose an efficient automated MCQ generation method to derive our Msg-MCQ. Figure 2 provides several automatically generated evaluation examples. Specifically, we pose four specific kinds of MCQs: **Yes-or-No, Fill-in-the-Blank, What, Correction** in different columns in Figure 2. Orthogonal to the specific kinds of MCQs, the evaluation focuses on different targets of semantic grounding that indicate the particular deficiency of evaluated LVLMs. In this work, we include six targets of semantic grounding: **Entity, Number, Color, Material, Action, and Spatial**. These target categories are shown in different rows in Figure 2. It is worth noting that the evaluation module can be easily extended to more types and targets in the future.

All these MCQs are generated from ground-truth source data through a four-step generation pipeline. In **step 1**, we curate a pool of question templates with placeholders for each kind of MCQ focusing on one specific target of semantic grounding, according to the source data we have. In **step 2**, we randomly sample one question template from the pool. As an illustration, a question template of **What** question focusing on color grounding is:

**Example 1 (What question)** What color of [obj-attr] object is featured in [bbox-color] bounding box of the image? (A) [distractor#1] (B) [ground-truth] (C) [distractor#2] (D) [distractor#3].

As can be seen, the sampled question template contains several placeholders, and the ground-truth is randomly placed into the choice (A), (B), (C), or (D). In **step 3**, we use source data to fill in these placeholders. For example, “[obj-attr], [ground-truth]” can be filled using the gold annotation in the source data. Regarding the “[bbox-color]”, we first use gold bounding box coordinates referring to the querying object to draw a box in the original image using a random color (green or red), then replace “[bbox-color]” with the name of that color. In **step 4**, we generate the distractors of ground truth based on the multimodal input information. This is one of the most critical steps in the evaluation pipeline, since distractors determine the difficulty level of MCQs. Distractors should not be semantically equivalent to ground truth, but they should be plausible enough to serve as an answer candidate to the question. Various distractor generation methods (Lu et al. 2022a) can be used, such as manual generation, thesaurus-based generation, and end-to-end generation. In this work, we utilize thesaurus-based generation with post-human verification. According to the type of ground truth (entity, number, color, etc.), we randomly sample 15 distractors. For entity grounding, we use the sentence transformer model (Reimers and Gurevych 2019) to filter out candidates that have overly high similarity scores to the gold answer. For other targets of grounding, the thesaurus is guaranteed to contain semantically different antonyms of the gold answer.

We introduce the necessary tweak of the overall pipeline for each specific kind of MCQ, and explain the purpose of why we need them as below:

- **Yes-or-No**: Can a model identify whether a textual description is appropriate for a given image? For each Yes-or-No MCQ, two choices are given, as shown in the first column of Figure 2. The correct textual descriptions are either directly taken from the source data (image captioning datasets), or obtained using some sentence templates
(object detection). The wrong descriptions are generated by negative replacement on correct descriptions, where we change specific textual spans for corresponding semantic grounding target (e.g. sampling “yellow” to replace the original “blue” in color grounding).

- **Fill-in-the-Blank.** Can a model infer the missing pieces of information regarding multimodal input? For each Fill-in-the-Blank MCQ, four choices are given where only one choice is correct. The second column of the first two rows of Figure 2 gives real examples of such type MCQs. A text span that contains the concept of interest (e.g. entity, number) is blanked, and the span is used as one choice. Three distracting choices are generated using negative sampling.

- **What.** Can a model recognize and identify an object or attribute that is specified in the multimodal input? For each What MCQ, four choices are given where only one choice is correct. The second column of the last four rows of Figure 2 gives real examples of such type MCQs. Similarly, three distracting choices are generated using negative sampling.

- **Correction** Can a model identify the inconsistency across different modalities and propose an appropriate correction? For each Correction MCQ, four choices are given where only one choice is correct. The last column of Figure 2 gives real examples of such type MCQs. The choice “(D) none of the above” is always included in each Correction MCQ, indicating the scenario where the
original description is grounded. Correction MCQs are challenging, as they require models to imagine whether the corrected text is consistent with the image.

Enhancement of Semantic Grounding

Our solution for enhancing the semantic grounding of LVLMs is a data-centric approach (Zha et al. 2023) to systematically and algorithmically generate the instruction-tuning dataset to feed LVLMs. The emerging success of transformer model architectures in both large language models (Ouyang et al. 2022; OpenAI 2023; Touvron et al. 2023) and large vision-language models (Zhu et al. 2023; Awadalla et al. 2023) advocates a fundamental shift from the model-centric AI to data-centric AI. Instead of focusing on designing specific model architectures for a particular downstream task, we curate a substantial (180K instances) of new multimodal multi-round conversational instruction data for LVLMs to be further improved upon.

Instruction Tuning Data Generation

<table>
<thead>
<tr>
<th>Image</th>
<th>User</th>
<th>LVLM</th>
<th>User</th>
<th>LVLM</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Image" /></td>
<td>What colors are the strawberries the man is holding?</td>
<td>The man is holding one white strawberry and one red strawberry.</td>
<td>Do you think the white strawberry is to the right of the red strawberry?</td>
<td>No, the white strawberry appears to be on the left of the red strawberry in the image.</td>
</tr>
</tbody>
</table>

Table 1: Toy Example of Multimodal Multi-Round Instruction-Tuning Data

For the instruction data generation, we first curate a set of fine-grained conversations of multimodal inputs that help LVLMs interpret the information in the image step by step and eventually answer the question. Table 1 provides a toy example of such instruction data. All the conversations are provided as multi-step chain-of-thought (Zhang et al. 2023) instructions to simulate the human reasoning process and guide LVLMs to pay attention to the right information at each step. Specifically, we design three types of instructions:

- **Multi-Round Conversation.** Given an image, we provide a multi-round conversation between users and models. The user asks a few questions about some fine-grained content in the image and the LVLM gives factual and precise answers based on the observation.

- **Vision-Prompted Recognition.** Given an image, we draw visual prompts (e.g., bounding boxes) on the image and ask an LVLM to tell the names, attributes, or relations of the objects indicated by visual prompts. The LVLM needs to learn how to follow the visual prompts to first localize the indicated objects and then recognize detailed attributes or relations of them.

- **Fact Checking.** Given an image and a statement of some facts in the image, we ask the LVLM if and why a given statement is factually consistent. In the instructions, we provide reference answers to guide the LVLM to localize main objects, recognize their attributes and relations, and determine if there are any factual misalignments. Eventually, the LVLM determines whether it is factual.

All three types of instructions essentially consist of two fundamental components: the question and the response. We manually curate diverse templates for both questions and responses. In a similar fashion, the response templates include placeholders to be filled with ground-truth data. Once these placeholders are populated, we employ chatGPT to rephrase these instructions, enhancing their diversity.

Instruction Tuning

After the fine-grained conversations are generated, we conduct instruction tuning (Liu et al. 2023a; Ouyang et al. 2022) on LVLMs to enhance semantic grounding. Following the notation we introduce in the Preliminary section, the inference process of LVLM can be described as:

\[
\hat{Y} = F_{\Theta}(X^v, X^q),
\]

where \( F_{\Theta} \) denotes the LVLM with trainable parameters \( \Theta \), \( X^v \) denotes the visual inputs, \( X^q \) denotes the textual inputs, and \( \hat{Y} \) denotes the textual response. It is worth noting that the trainable parameters \( \Theta \) of \( F \) actually contains \( \Theta = \{ \phi, \omega, \psi, \theta \} \). We denote the \( n \)-th training sample \((X^v_{(n)}, X^q_{(n)}, Y_{(n)})\) from our generated instruction tuning data, where they represent the visual input, textual query input, and the ground truth response, respectively. The loss function of LVLM is

\[
\mathcal{L}(\Theta) = -\frac{1}{N} \sum_{n=1}^{N} \log Pr(Y_{(n)}|X^v_{(n)}, X^q_{(n)}),
\]

where \( Pr \) denotes the probability of generating the output sequence \( Y_{(n)} \) from input \( (X^v_{(n)}, X^q_{(n)}) \) using equation (6).

Experiments and Analysis

Evaluation Examples

Table 2 gives an overview of key statistics of test multiple-choice questions generated by MsG-MCQ. We collect six subsets for each semantic grounding type, together with 9K MCQs, 8.1K images, and 1.7K unique answers. These MCQs are built from four public datasets: Flickr30K (Young et al. 2014), PACO (Ramanathan et al. 2023), OpenImage-V7 (Krasin et al. 2017), SpatialSense (Yang, Russakovsky, and Deng 2019). Flickr30K is an image captioning dataset that includes images obtained from Flickr and each image is provided with five manually annotated captions.
We curate these testing MCQs by our proposed MSG-MCQ. Furthermore, OpenImage-V7 and SpatialSense provide relational annotations between two objects within one image. We curate these testing MCQs by our proposed MSG-MCQ. Also, we balance the answer distribution and the concept of semantic grounding target for promising evaluation.

**Evaluated Models**
We select seven SOTA LVLMs for evaluation and all use the 7B parameters variants for a fair comparison:
- mPLUG-Owl (Ye et al. 2023), MiniGPT4 (Zhu et al. 2023), LLaVA (Liu et al. 2023a), InstructBLIP (Dai et al. 2023): LVLMs based on the visual encoder and the pre-trained LLM that are similar to the architecture we introduce in the Preliminary Section. Both follow a two-stage training, where stage-1 is a pre-training stage to align visual and textual concepts when LLM is frozen, and stage-2 is a fine-tuning stage to feed into instruction-following data to train both visual encoder and LLM simultaneously.
- Otter (Li et al. 2023a), LLaMA-AdapterV2 (Gao et al. 2023), LaVIN (Luo et al. 2023): LVLMs with a slightly different architecture, where visual features after adaptation/fusion are attended with text embedding in each/some layers of the LLM component.

We also test one large language model (language-only): LLaMA-2-chat (Touvron et al. 2023) to verify the necessity of multimodal perception ability for answering the generated MCQs. To extract the choice indices (A, B, C, or D) from the free-form responses of LVLMs, we use regular expressions if they explicitly contain indices. Otherwise, we use ChatGPT to help extract indices.

**Zero-Shot Setting**
Figure 3 provides the radar charts of seven advanced LVLMs on each semantic grounding target or each question type. Among these LVLMs, Otter, LLaVA, and LaVIN are top competitors that consistently deliver better performance than other LVLMs. In terms of semantic grounding targets, LVLMs are more effective at perceiving and understanding Entity and Action, but extremely struggle in Spatial relations. We speculate the different performances on different kinds of semantic grounding come from the bias of training data used by LVLMs. mPLUG-owl, LLaMA-AdapterV2, MiniGPT-4, and InstructBLIP all mainly trained on coarse-grained text-image pairs corpus. On the other hand, Otter has been trained on their curated MIMIC-IT corpus that covers perception, reasoning, and planning-oriented text-image QA pairs. LLaVA has been trained on not only the captions, but also bounding boxes with detailed descriptions. The model checkpoint of LaVIN we used here is trained on ScienceQA (Lu et al. 2022b) corpus, which is in the format of MCQs. These additional training corpora provide more fine-grained information on the multimodal input, thus helping these LVLMs achieve better performance.

Table 3 provides a detailed view of the zero-shot accuracy including human, random-guess, and LLaMA2 baselines. The accuracy of random-guess in each semantic grounding type is always 33.33, since each semantic grounding subset contains 500 two-way (50.00 accuracy by random guess) and 1000 four-way MCQs (25.00 accuracy by random guess). Human performance (five annotators carefully answer random 500 MCQs from MSG-MCQ) sets up the upper bound of the zero-shot experiments, which significantly outperforms all other LVLMs. Interestingly, humans do not perform well in color, material, and spatial. One possible reason for human deficiency in color and material is that human perception organs are a bit weaker at distinguishing attributes, as compared to recognizing objects and actions. For Spatial MCQs, we observe that the source data quality is not high. For example, sometimes the spatial relations are labeled according to the physical position, while sometimes they are labeled according to what the annotator perceived. Comparing the language only LLaMA2-chat and other LVLMs, the best LVLMs are consistently better in every semantic grounding type. This indicates the importance of the perception ability of input vision modality. Besides, LVLM models struggle more with correction questions, which is also more challenging for humans.

Moreover, Table 4 supplies LVLMs performance on different Q-Types with human, random-guess, and LLaMA2 baselines. As can be seen, Correction type MCQs are most challenging for both humans and LVLMs. Given the accuracy of the Random-Guess baseline as 25.00 (four-way MCQs with one correct answer), the best LVLM (Otter) only outperforms it by 11.63. While for other four-way What and Fill-in-the-Blank MCQs, the best LVLM achieves 17.70.

<table>
<thead>
<tr>
<th>Grounding Target</th>
<th>Q Type</th>
<th>#IMGs</th>
<th>#Qs</th>
<th>#As</th>
<th>Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>YoN, FiB, Corr</td>
<td>1,339</td>
<td>1,500</td>
<td>628</td>
<td>Flickr30K</td>
</tr>
<tr>
<td>Number</td>
<td>YoN, FiB, Corr</td>
<td>977</td>
<td>1,500</td>
<td>389</td>
<td>Flickr30K</td>
</tr>
<tr>
<td>Color</td>
<td>YoN, What, Corr</td>
<td>1,500</td>
<td>1,500</td>
<td>153</td>
<td>PACO</td>
</tr>
<tr>
<td>Material</td>
<td>YoN, What, Corr</td>
<td>1,500</td>
<td>1,500</td>
<td>68</td>
<td>PACO</td>
</tr>
<tr>
<td>Action</td>
<td>YoN, What, Corr</td>
<td>1,498</td>
<td>1,500</td>
<td>322</td>
<td>OpenImage-V7</td>
</tr>
<tr>
<td>Spatial</td>
<td>YoN, What, Corr</td>
<td>1,500</td>
<td>1,500</td>
<td>144</td>
<td>SpatialSense</td>
</tr>
<tr>
<td>Overall</td>
<td>YoN, FiB, What, Corr</td>
<td>8,119</td>
<td>9,000</td>
<td>1,683</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Overview of MCQs generated by MSG-MCQ. #IMGs denotes unique images, #Qs denotes unique questions, and #As denotes unique answers.
and 26.10 accuracy gains, separately. On the other hand, Yes-or-No type MCQs are quite challenging too. Given the Random-Guess accuracy as 50.00, the best LVLM (LLaMA-Adapter) only achieves 5.9 higher accuracy. The patterns of experimental outcomes indicate that LVLMs are better at responding to descriptive queries (What and Fill-in-the-Blank MCQs). One speculation for that is these LVLMs have been trained on a great amount of image captioning data and image description instruction data. With the pretraining on such training corpora, LVLMs may be relatively less prepared for judgemental and corrective queries.

We further conduct an in-depth analysis using critical difference (CD) analysis (Demšar 2006). Figure 4 shows the CD diagram, which is a powerful tool to compare outcomes of multiple compared models over multiple observations. The CD analysis involves several hypothesis tests, and models connected with each other in the diagram means that the performances of these models are not that different in the sense of statistical significance. As shown in Figure 4, the performances of LVLMs are relatively less prepared for judgemental and corrective queries.
to highlight the second-order performance gain for LA VIN is significantly higher than for other models. Interestingly, the performance gain for LA VIN is significantly higher than for other models. We believe this is primarily due to LA VIN's ability to automatically generate instruction data, enabling end-to-end optimization of the entire model. Moreover, we supply Table 6 that offers a detailed breakdown of accuracy improvements based on MCQ types. A closer investigation of the table reveals that while performance enhancements are evident across most MCQ types, there are instances of decreased accuracy in specific MCQ types. For instance, LA VIN exhibits a notable decrease of 15.20% in Fill-in-the-Blank MCQs, while LLaMA-AdapterV2 records slightly lower accuracies in Yes-or-No and Fill-in-the-Blank MCQs. Despite these isolated variations, consistent performance improvements are observed in other categories. In summary, LVLMs enhanced with our automatically generated instruction data deliver better semantic grounding performance. The instruction data is fundamentally different from the training data, since instruction data does not share the same data distribution and formats of the MSG-MCQ testing data. The instruction data also serves as a valuable resource for instruction tuning, readily available for any LVLMs to utilize.

Enhanced LVLMs Setting

In order to enhance semantic grounding in LVLMs, we conduct instruction tuning on LVLMs by generating 180K fine-grained multimodal instruction data covering multimodal conversations, vision-prompted recognition, and fact-checking. Table 5 shows the performance gains on several enhanced LVLMs by instruction tuning. Following the same format, we use **bold font** to highlight the best accuracy or accuracy gain, and *underline font* to highlight the second-best accuracy or accuracy gain. For those LVLMs not included, they typically do not release well-established instruction tuning scripts (or scripts are specifically for image captioning tasks). As can be seen, we observe consistent improvements over all tuned LVLMs, which indicates the effectiveness of our enhancement method. Interestingly, the performance gain for LA VIN is significantly higher than for other models. We believe this is primarily due to LA VIN's ability to automatically generate instruction data, enabling end-to-end optimization of the entire model. Moreover, we supply Table 6 that offers a detailed breakdown of accuracy improvements based on MCQ types. A closer investigation of the table reveals that while performance enhancements are evident across most MCQ types, there are instances of decreased accuracy in specific MCQ types. For instance, LA VIN exhibits a notable decrease of 15.20% in Fill-in-the-Blank MCQs, while LLaMA-AdapterV2 records slightly lower accuracies in Yes-or-No and Fill-in-the-Blank MCQs. Despite these isolated variations, consistent performance improvements are observed in other categories. In summary, LVLMs enhanced with our automatically generated instruction data deliver better semantic grounding performance. The instruction data is fundamentally different from the training data, since instruction data does not share the same data distribution and formats of the MSG-MCQ testing data. The instruction data also serves as a valuable resource for instruction tuning, readily available for any LVLMs to utilize.

Conclusion and Future Work

In this work, we evaluate and enhance the ability of LVLMs to ground fine-grained vision-language inputs. We propose an evaluation method MSG-MCQ to automatically generate testing samples focusing on specific grounding targets in various question formats, along with comprehensive experiments to understand how well the current state-of-the-art LVLMs perform in semantic grounding. We further propose a data-centric approach to enhance LVLMs. Its effectiveness is validated by observing consistent performance improvement of these LVLMs, after instruction tuning on multimodal multi-modal conversations generated by the enhancement method. In the future, we aim to extend the scope of semantic grounding in LVLMs into (1) more modalities such as audio, time-series, and tabular; (2) more semantic grounding targets such as social-emotion, terrains, and human organs.
References


Appendices

Implementations of Compared LVLMs
Implementations of the compared baselines are from codebases of original authors, with their open-sourced URLs as follows:

• LLaMA2-chat (Touvron et al. 2023): https://github.com/facebookresearch/llama, with the LLaMA2-7B-chat checkpoint.

• mPLUG-owl (Ye et al. 2023): https://github.com/X-PLUG/mPLUG-Owl, with the mPLUG-Owl-7B checkpoint.

• LLaMA-AdapterV2 (Gao et al. 2023): https://github.com/OpenGVLab/LLaMA-Adapter, with the LLaMA-AdapterV2-multimodal checkpoint.

• LaVIN (Luo et al. 2023): https://github.com/luogen1996/LaVIN, with the LaVIN-7B checkpoint.

• MiniGPT-4 (Zhu et al. 2023): https://github.com/Vision-CAIR/MiniGPT-4, with the MiniGPT-4-Vicuna7B-V0 checkpoint.

• LLaVA (Liu et al. 2023a): https://github.com/haotian-liu/LLaVA, with the LLaVA-Vicuna7B-v1.1 checkpoint.

• Otter (Li et al. 2023a): https://github.com/Luodian/Otter, with the OTTER-Image-MPT7B checkpoint.


As can be seen, we use 7B parameter size checkpoints for all compared LVLMs and the text-only LLaMA2-chat. All LVLMs utilize LLaMA1 as the large language model component for a fair comparison. The hyper-parameters for inference (e.g. temperature, number of beams, etc.) for each LVLM are directly taken from the default ones recommended by their authors in the original codebases, respectively.

For instruction tuning, we also keep the default hyperparameters (e.g. batch size, optimizer, learning rate, etc.) except explicitly set the max tuning epoch as 3 for all LVLMs. Specifically, for instruction tuning of LLaMA-AdapterV2, the visual projector and language model adaptor are updated while leaving the pre-trained visual encoder and the pre-trained language model frozen. For mPLUG-Owl, only the pre-trained language model is updated. For LaVIN, the visual encoder adaptors, the visual projector, and the language model adaptors (namely “Mixture-of-Modality Adapter” in the original paper) are updated. For LLaVA, the visual projector and the language model are updated. For Otter, the Perceiver resampler of the visual encoder, input/output embeddings of the language model, and the cross-attention layers are updated.

Case Studies for Enhanced LVLMs
We investigate the differences between responses from original LVLMs and responses from treated LVLMs. Table 7 shows two real examples using the same data source with free-form questions. The compared LVLM is LLaVA, between the original 7B checkpoint version and the instruction-tuned version (denoted by \([\text{method+}]\)). In Case #1, the image depicts a well-furnished bedroom softly illuminated, creating a cozy and inviting ambiance. A red bounding box highlights a chair, while a green bounding box highlights a bed adorned with a blue comforter and pillows. Before instruction tuning, LLaVA struggled to provide accurate responses to questions about this image. For example, LLaVA mistakenly identified the chair as highlighted by the green bounding box, and the bed as highlighted by the red bounding box. In contrast, LLaVA+ produced more accurate responses to these questions. Furthermore, when asked about the spatial relationship between the chair and the bed, LLaVA responded incorrectly, stating that “the chair is in front of the bed.” In contrast, LLaVA+ correctly indicated that the bed is to the left of the chair. In Case #2, the image features a woman wearing a Santa hat while riding a surfboard in the water. This time, both LLaVA and LLaVA+ successfully identify the object within the green bounding box, which is the woman. When moving to the red bounding box, LLaVA fails to ground it and still recognizes it as the woman, while LLaVA+ adeptly recognizes it as the surfboard. LLaVA’s failure with the red bounding box results in an incorrect response to the subsequent question concerning the interaction between these two bounding boxes. These two illustrative examples support the effectiveness of our proposed data-centric enhancement method in aiding LVLMs in recognizing and interpreting nuanced multimodal information.
### Case#1

Source: SpatialSense (Yang, Russakovsky, and Deng 2019)

<table>
<thead>
<tr>
<th>User</th>
<th>What is in the green bounding box in the image?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLaVA</td>
<td>In the green bounding box in the image, there is a chair.</td>
</tr>
<tr>
<td>LLaVA+</td>
<td>The image shows the bed located in the green bounding box.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>What is in the red bounding box in the image?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLaVA</td>
<td>In the red bounding box in the image, there is a bed.</td>
</tr>
<tr>
<td>LLaVA+</td>
<td>The image shows the chair located in the green bounding box.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>What is the spatial relationship between them?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLaVA</td>
<td>The bed is located next to the chair, with the chair positioned in front of the bed.</td>
</tr>
<tr>
<td>LLaVA+</td>
<td>In the image, the bed in the green bounding box is to the left of the chair in the red bounding box.</td>
</tr>
</tbody>
</table>

### Case#2

Source: OpenImage-V7 (Krasin et al. 2017)

<table>
<thead>
<tr>
<th>User</th>
<th>What is in the green bounding box?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLaVA</td>
<td>In the green bounding box in the image, there is a woman wearing a Santa hat while riding a surfboard in the ocean.</td>
</tr>
<tr>
<td>LLaVA+</td>
<td>The image displays the woman located in the green bounding box.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>What is in the red bounding box?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLaVA</td>
<td>In the red bounding box, there is a close-up of a woman wearing a Santa hat while riding a surfboard in the ocean.</td>
</tr>
<tr>
<td>LLaVA+</td>
<td>The surfboard is situated within the red bounding box in the picture.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>What is the interaction between them?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLaVA</td>
<td>The interaction between the objects in the red bounding box and the green bounding box is that the woman in the red box is riding a surfboard in the ocean, while the woman in the green box is also riding a surfboard in the ocean.</td>
</tr>
<tr>
<td>LLaVA+</td>
<td>In the context of the image, the woman in the red bounding box surf the surfboard in the green bounding box.</td>
</tr>
</tbody>
</table>

Table 7: Case study regarding spatial misgrounding. LLaVA+ denotes the enhanced LLaVA using our proposed instruction tuning data. Text in cyan indicates the precise statement. Text in orange indicates the ambiguous statement.