Appendix

A. Computational Resources

The evaluations of open-source MLLMs were conducted on a single A100 GPU with 40GB of RAM, which required approximately 200 hours on our university-wide computing infrastructure. To evaluate GPT-4V/GPT-4o, which are closed-source, we used the paid ChatGPT API provided by OpenAI and spent $922 for GPT-4V and $451 for GPT-4o, including runs to tune prompts.

B. Additional Dataset Details

In this section, we present additional dataset details.

B.1. TallyQA

The counting questions in TallyQA are classified into complex and simple counting questions [2]. Simple counting questions were imported from existing datasets like VQA2 and Visual Genome. Complex questions were collected using Amazon Mechanical Turk (AMT) to gather 19,500 complex questions for 17,545 unique images. The images were sourced from both COCO and Visual Genome to ensure variety. The testing set of TallyQA contains 38,589 questions, which is a reasonable size. Therefore, we evaluated models on the entire original test set. The distribution of unique answers is given in Table 5. TallyQA is provided under the terms of the Apache License Version 2.0, January 2004: http://www.apache.org/licenses/

Table 5. The distribution of unique answers in TallyQA.

Answer	Complex	Simple
zero	4335	637
one	6853	12308
two	2479	5636
three	901	2034
four	453	1101
five	195	435
six	133	319
seven	70	152
eight	69	145
nine	31	84
ten	33	48
eleven	12	30
twelve	25	33
thirteen	7	13
fourteen	6	9
fifteen	6	7

B.2. VQDv1

VQDv1 [3] was created synthetically using annotations from Visual Genome, COCO, and COCO Panoptic. This synthetic generation approach helps combat certain biases. The queries are generated using multiple templates for each type, allowing for diverse queries. The annotations used to generate these questions are derived from a combination of COCO's object annotations and Visual Genome's attribute and relationship information.

For VQDv1, almost 90% of the queries have less than two ground truth bounding boxes. In our subset, we retained all queries with more than one ground truth bounding box, and we sampled 10% of the queries with zero or one ground truth bounding box. Table 6 provides the distribution of ground truth boxes across queries. The VQDv1 dataset is provided under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) license: https://creativecommons.org/licenses/by/4.0/legalcode

B.3. DVQA

The DVQA dataset was created by synthetically generating bar charts to test multiple aspects of bar chart understanding. This automatic generation process allows precise control over the visual elements' positions and appearances, and provides access to meta-data about the elements in the image, which is not available with real data [15].

The original version of DVQA had two test sets: Test-Familiar and Test-Novel. The critical difference between these two sets is that every bar chart in Test-Familiar has labels in DVQA's training set, whereas Test-Novel does not. Given that we are conducting zero-shot evaluations, these two sets can be treated equivalently. Therefore, we sample the same number of questions from both. Table 7 shows the question distributions of our subset version of DVQA. The DVQA dataset is provided under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0): https://creativecommons.org/licenses/by/4.0/legalcode

B.4. TDIUC

The TDIUC dataset was created by incorporating questions from three sources: existing datasets, questions generated based on image annotations, and human annotators. Questions were imported from COCO-VQA and Visual Genome datasets, with templates and regular expressions used to classify and generate questions[14]. Additionally, questions were generated using COCO's semantic segmentation annotations and Visual Genome's objects and attribute annotations[14]. For certain question types like sentiment understanding and object utility/affordance, trained volunteers performed manual annotation using a web-based tool[14]. We sample proportionately from 12 question types in TDIUC. Table 8 shows our subset of TDIUC question distributions. TDIUC is a public dataset but does not mention a particular

Table 6. Bounding box distribution for the original and modified versions of VQDv1.

Bounding Box Count	Original Version	Our Version
0	80025 (42.08%)	8001 (21.59%)
1	90101 (47.38%)	9008 (24.31%)
2	10127 (5.33%)	10127 (27.33%)
3	3200 (1.68%)	3200 (8.64%)
4	1894 (1.00%)	1894 (5.11%)
5	1334 (0.70%)	1334 (3.60%)
6	700 (0.37%)	700 (1.89%)
7	533 (0.28%)	533 (1.44%)
8	366 (0.19%)	366 (0.99%)
9	305 (0.16%)	305 (0.82%)
10	276 (0.15%)	276 (0.74%)
11	193 (0.10%)	193 (0.52%)
12	194 (0.10%)	194 (0.52%)
13	255 (0.13%)	255 (0.69%)
14	618 (0.32%)	618 (1.67%)
15	26 (0.01%)	26 (0.07%)
16	5 (0.00%)	5 (0.01%)
17	7 (0.00%)	7 (0.02%)
18	7 (0.00%)	7 (0.02%)
19	3 (0.00%)	3 (0.01%)
20	1 (0.00%)	1 (0.00%)
23	2 (0.00%)	2 (0.01%)
25	1 (0.00%)	1 (0.00%)
26	1 (0.00%)	1 (0.00%)

Table 7. Distribution of question types in DVQA.

Question Type	Test-Familiar Version	Test-Novel Version	Our Version
Data	185356 (31.93%)	185452 (31.90%)	9269 (31.91%)
Reasoning	316923 (54.59%)	316881 (54.51%)	15844 (54.55%)
Structure	78278 (13.48%)	78988 (13.59%)	3930 (13.53%)

license: https://kushalkafle.com/projects/tdiuc.html

C. Creating "Slim" Evaluation Sets

We evaluate MLLMs on the entire validation set of TallyQA, which contains 38,589 questions. However, the other datasets are much larger, which makes it challenging to quickly and inexpensively evaluate MLLMs on them. To address this, we sample subsets from these datasets for evaluation. A uniform random sampling is suboptimal as these datasets have long-tailed distributions and sampling uniformly would result in discarding examples from the tail. Therefore, we adopt a stratified sampling approach for DVQA and TDIUC, where we also maintain as much answer variety as possible. Specifically, we first categorize the questions into fine-grained groups, defined by both the predefined types in the datasets (e.g., question types or difficulty levels) and their corresponding answers. We define r as the sampling ratio and k as the minimum number of samples from each group. For any large group, we uniformly sample an r proportion of the entries. For smaller groups, if the size m is such that $m \cdot r$ is less than k, we sample k entries. For groups even smaller than k, we use the entire group. The number of samples m' to be taken from group $|g_i| = m$ can be represented as follows:

$$m_i' = \begin{cases} m_i & \texttt{if } m_i \leq k \\ k & \texttt{if } m_i \cdot r < k \wedge m_i > k \\ \lceil m_i \cdot r \rceil & \texttt{if } m_i \cdot r \geq k \end{cases}$$

VQDv1 has a long-tail distribution regarding the number of bounding boxes per query, where queries with 0 or 1 box comprise almost 90% of the dataset. Our goal is to evaluate the MLLM's ability to generate a variable number

Table 8. Distribution of question types in TDIUC.

Question Type	Original Version	Our Version
Absurd	120411 (22.35%)	6844 (25.00%)
Activity Recognition	2682 (0.50%)	77 (0.28%)
Attribute	9200 (1.71%)	296 (1.08%)
Color	62490 (11.60%)	2142 (7.82%)
Counting	52905 (9.82%)	2262 (8.26%)
Object Presence	215324 (39.96%)	11884 (43.41%)
Object Recognition	30693 (5.70%)	1646 (6.01%)
Positional Reasoning	12284 (2.28%)	523 (1.91%)
Scene Recognition	22032 (4.09%)	1188 (4.34%)
Sentiment Understanding	634 (0.12%)	27 (0.10%)
Sport Recognition	10042 (1.86%)	478 (1.75%)
Utility Affordance	171 (0.03%)	12 (0.04%)

of bounding boxes – extending the evaluation scope beyond traditional referring expression comprehension datasets such as RefCOCO [27], where all referring expressions are associated with only one bounding box. Therefore, we retained all the questions with more than one bounding box and randomly sampled queries corresponding to 0 or 1 bounding box. As seen in Table 6, this method effectively increases the ratio of questions with multiple bounding boxes.

Our sampling method preserves the most challenges samples present in the original dataset, ensuring a comprehensive evaluation while significantly reducing computational overhead. Summary statistics for the datasets are given in Table 9.

D. Prompt Engineering

To make the model performance comparison as fair as possible, we endeavored to keep the prompts consistent across different models. However, this was challenging due to variations in the models' ability to process the prompts. For example, BLIP2 and iBLIP failed when prompted to answer using a template such as "My answer is ¡answer¿." Inspired by Liu et al. [23], for TDIUC, DVQA, and TallyQA, we prompt the models to answer as concisely as possible instead of asking them to generate entire sentences. These prompts are given in Fig. 4.

Figure 4. Prompts used for TallyQA, DVQA, and TDIUC.

- TallyQA: Please answer the question in one word.
- DVQA: Please answer the question in one word
- TDIUC: Please answer in one word. Answer 'doesnotapply' if the question is not related to the image or cannot be answered.

Despite much effort, for VQDv1, we were unable to identify a universal prompt for generating multiple bounding boxes that worked well across models. For example, as shown in Table 10, LLaVA (7B) repeatedly generated the same bounding boxes until the maximum token limit was reached when this prompt was used. We believe this occurs because the model is confused by the instruction to generate multiple bounding boxes, even when only one object is detected. This may explain why it repeatedly generates the same bounding box. While we considered non-maximal suppression or eliminating redundant boxes, our goal is to fairly evaluate MLLMs without excessively post-processing their outputs. Therefore, we fine-tuned the prompts for different models. The results reported in the paper represent the best outcomes from our evaluations. The best-identified prompts for each model on VQDv1 are given in Table 11.

E. Model Details

In this paper, all the open source MLLMs are loaded directly from HuggingFace, the detail models are below:

GPT-4v/4o are not open sourced, therefore we are unable to identify the models. We utilize the API released by OpenAI to evaluate four datasets on GPT-4v/4o.

F. Additional Evaluation Details

Root Mean Squared Error (RMSE) Computation. For TallyQA, besides Micro and Macro Accuracy, we also compute RMSE. However, we observed that due to the unpredictability of the MLLMs, the models occasionally output unreasonably large numbers as their predicted object counts. For instance, LLaVA-NeXT predicts an unreasonably large object count of 150 for one of the questions. Such outliers significantly inflate the models' overall average RMSE across all questions. As shown in the distribution of TallyQA questions, all counting numbers are between 0 and 15. Therefore, we apply a simple cutoff technique: an upper bound of

Table 9. Summary statistics for the VQA and VQD datasets we study.

Dataset Name	# of Categories	# of Unique Answers	Original Size	Sampled Size
TDIUC [14]	12	562	538,868	27,336
TallyQA [2]	2	16	38,589	38,589
DVQA [15]	3	2113	580,557	29,025
VQDv1 [3]	5	24	190,174	37,057

Table 10. LLaVA (7B) struggled with some prompts for VQDv1.

Prompt with Query	Where is the motorcycle? Instruction: **Generate a list of bounding box coordinates around the objects that the prompt mentioned if they exist in the image. Even if the question uses a singular verb like 'is', you should still generate multiple bounding boxes if there are multiple objects that satisfy the prompt. The bounding box list should be in the following format: [[x_min, y_min, x_max, y_max], [x_min, y_min, x_max, y_max]]**
LLaVA (7B)	predicted_answer: "[[0.01, 0.0, 0.28, 0.24], [0.01,"

15 and a lower bound of 0 is applied to all predicted counts. This adjustment ensures that the RMSE remains meaningful and useful for analysis.

Match Answer with Ground Truth. For TallyQA, the model is tasked with generating object counts. If the model correspondingly generates a number enclosed within a string, such as "2", we directly convert it to int type by type conversion. For the case where the model generates a word, we map the word to its corresponding number using the mappings shown in table 5. Occasionally, the model generates answers that, while not numerical, still make sense. For example, the model might generate 'none' or 'no,' which we interpret as zero. We manually account for these cases and add additional mappings accordingly. While we acknowledge that even with these steps, we may still miss some unpredictable answers from the models, such as when the model responds with 'a few,' which is completely uninterpretable, we map these to None.

In datasets like TallyQA, DVQA, and VQDv1, synonymous answers are rarely an issue due to the specific nature of each task. For example, TallyQA typically expects numerical answers that are definitive and unambiguous (numbers seldom have synonyms). The main exception is when 'none,' 'no,' and 'zero' are all interpreted as 0. In DVQA, which focuses on chart understanding, questions such as 'Which bar has the highest number?' require the model to read and provide the exact text from the graph, minimizing the possibility of synonymous answers. Similarly, VQDv1 involves generating bounding boxes and computing the Intersection over Union (IoU) to determine if the ground truth is correctly matched. The evaluation uses Recall and Precision metrics, which are not binary and therefore do not penalize synonymous answers.

Word	Number	Word	Number	Word	Number
zero	0	four	4	eight	8
none	0	five	5	nine	9
no	0	six	6	ten	10
one	1	seven	7	eleven	11
two	2	twelve	12	fourteen	14
three	3	thirteen	13	fifteen	15

Figure 5. Mapping of words to numbers in TallyQA

In contrast, tasks in TDIUC are more likely to involve more interpretative answers. For example, the answers 'phone' and 'telephone' should be considered semantically similar and should both be acceptable if the ground truth is one of them. To minimize penalizing synonymous answers like the case above, we leverage WordNet[32], a lexical database for the English language that is specifically designed for natural language processing. Specifically, we retrieve the sets of synonyms for each word from WordNet (using the synsets function) and compare these sets. If there

Table 11. The best identified VQDv1 prompts for each model.

Model	Prompt
GPT-4o/GPT-4V	Generate a list of bounding box coordinates around the objects mentioned in the prompt if they exist in the image. Even if the prompt uses a singular verb like 'is', generate multiple bounding boxes if multiple objects satisfy the query. The bounding box list should be formatted as: [[x_min, y_min, x_max, y_max]], and it can contain zero or more bounding boxes. Only provide the bounding box list, without any additional descriptions.
LLaVA-NeXT	Please generate a list of bounding boxes coordinates of the region this query describes. Use the format [[x_min,y_min,x_max,y_max]....]. Do not respond in sentences, and only generate the bounding boxes. Respond with an empty list [[]], if no such region exists in the image.
LLaVA (7B)/(13B)	Please answer the question by generating a list of bounding box coordinates around the objects the question is asking, and if no such object exists in the image, answer: [[]]

Table 12. MLLM Model Repository Paths

Model	Repository Path
LLaVA-NeXT	llava-hf/llava-v1.6-mistral-7b-hf
InstructBlip	Salesforce/instructblip-flan-t5-xxl
BLIP2	Salesforce/blip2-flan-t5-xl
LLaVA1.5-7b	llava-hf/llava-1.5-7b-hf
LLaVA1.5-13b	llava-hf/llava-1.5-13b-hf

is any overlap in the synsets, two words are considered synonyms, and we use this to evaluate if the predicted word(s) matches the ground truth(s).

G. Other Related Efforts and Limitations

Related Efforts to Improve MLLM Evaluation. Recent works highlight challenges in evaluating MLLMs. In [36], the ARO benchmark was introduced to assess models' understanding of complex compositional elements, and models evaluated on it performed poorly for like "the grass is eating the horse" versus "the horse is eating grass." Similarly, the Winoground datasets [31] require models to match images with captions that use identical words in different orders to assess their comprehension of linguistic composition concerning visual information. In [29], a cycle-consistency framework is proposed, evaluating models' ability to understand semantically similar questions. These studies complement ours and reveal other biases and limitations in MLLMs.

Limitations. One significant challenge we encountered was effectively prompting the models (see Appendices D and H.2). The performance of MLLMs is susceptible to the phrasing and structure of prompts, with small changes leading to significant variations in outputs. Crafting prompts that balance complexity and clarity is difficult, especially given the diversity of tasks and datasets. Additionally, no standardized approach to prompt engineering across different models complicates fair comparisons. We experimented with various formulations to find effective prompts, but our approach may still have limitations. Future work should focus on developing systematic and standardized methods for prompt engineering to ensure consistent and fair evaluations.

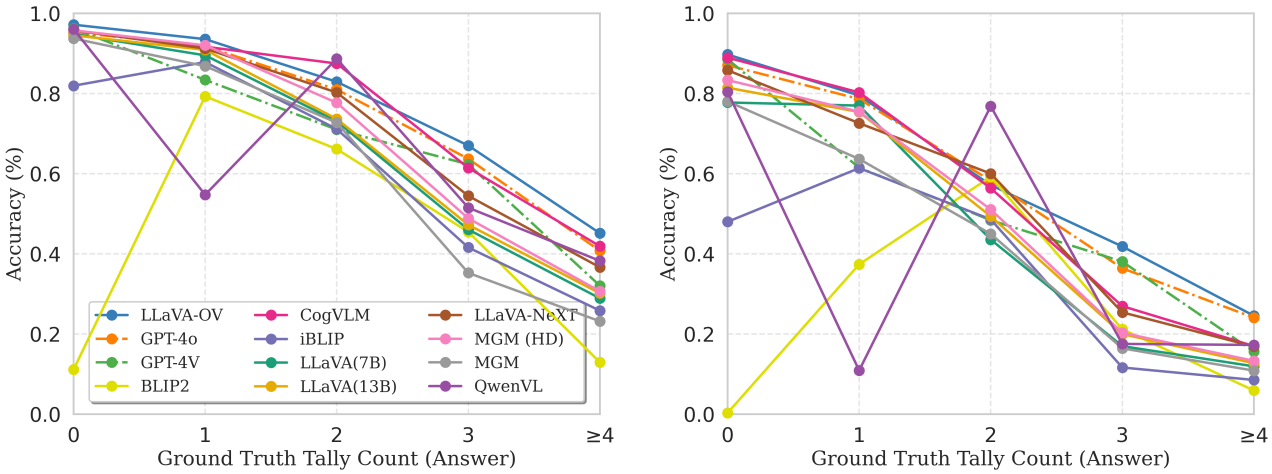
H. Additional Results

H.1. TallyQA

For TallyQA, we found that the performance of most models decreases as the correct number to output increases, as shown in Figs. 6a and 6b. Across counts, models perform much better at answering simple questions than complex questions.

H.2. VQDv1

Alternative Prompts. All models performed poorly on VQDv1. As mentioned earlier, it was challenging to identify the best prompt for each model. We hypothesized that given the verbosity of GPT-4o, it would benefit from being allowed to provide more extended responses where it reasons 'aloud.'



(a) Simple counting questions in TallyQA.

(b) Complex counting questions in TallyQA.

Figure 6. Accuracy as a function of the correct answer for simple and complex counting questions in TallyQA.

However, this performed worse than the prompts used in our main results. In Table 13, we provide alternative prompts that we tried, where the results are given in Table 14.

Qualitative Examples. Among all the datasets we evaluated, all models consistently performed poorly on VQDv1. Consequently, we provide qualitative examples from VQDv1 in the figures below, using the prompts employed in our main results. These visualizations demonstrate the challenges models face when required to detect multiple objects.

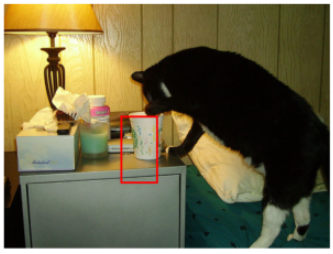
Table 13. Alternative prompts studied for VQDv1.

Model	**Prompt**
GPT-4o	Please generate a list of bounding boxes coordinates for regions that match what is described in the query. Bounding boxes should use the format: [[x_min,y_min,x_max,y_max], ..], where (x_min,y_min) is top left coordinate,(x_max,y_max) is bottom right coordinate. If there are no objects in the image that the query describes, please respond with an empty list. You can explain your answers if necessary, but end your response with the format: The bounding boxes coordinates are ¡box¿[[x_min,y_min,x_max,y_max],..,..]¡box¿. Please keep the special token ¡box¿ in your response.
LLaVA-NeXT	Generate a list of bounding box coordinates around the objects mentioned in the query, if they exist in the image. Even if the query uses a singular verb like 'is', generate multiple bounding boxes if multiple objects satisfy the query. The bounding box list should be formatted as: [[x_min, y_min, x_max, y_max]], and it can contain zero or more bounding boxes. Only provide the bounding box list, without any additional descriptions.
LLaVA (7B)/(13B)	Generate a list of bounding box coordinates around the objects that the prompt mentioned if they exist in the image. Even if the query uses a singular verb like 'is', you should still generate multiple bounding boxes if multiple objects satisfy the prompt. The bounding box list should be in the following format: [[x_min, y_min, x_max, y_max], [x_min,y_min, x_max, y_max]].

Table 14. MLLM performance on VQDv1 using the alternative prompts from Table 13.

Model	LLaVA (7B)	LLaVA (13B)	LLaVA-NeXT	GPT-4o
Micro F_1	4.27%	8.90%	14.66%	23.81%

Question: Show the cup in the picture.

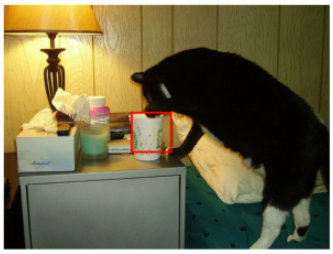


model: GPT-4o

model: LLaVA-NeXT

model: LLaVA-v1.5-7b

model: GPT-4V



model: LLaVA-v1.5-13b

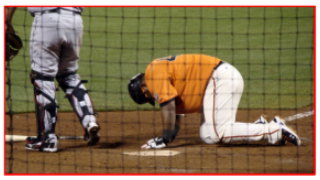
Ground truth

Question: Where is the person in the picture?

model: GPT-4o

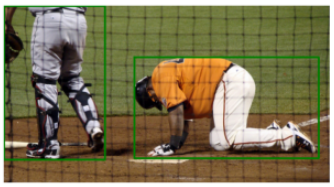
model: LLaVA-NeXT

model: LLaVA-v1.5-7b



model: GPT-4V

model: LLaVA-v1.5-13b

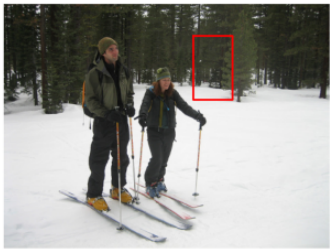


Ground truth

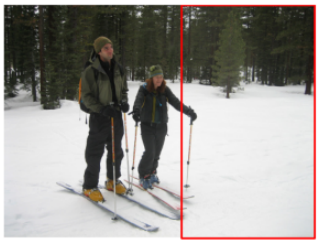
Question: Which tree is green in color?

model: GPT-4o

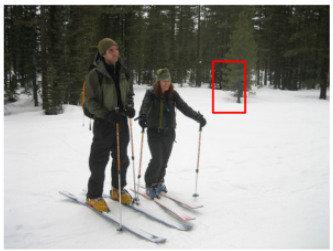
model: LLaVA-NeXT



model: LLaVA-v1.5-7b



model: GPT-4V



model: LLaVA-v1.5-13b

Ground truth