LAMM: Language-Assisted Multi-Modal Instruction-Tuning Dataset, Framework, and Benchmark

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models have become a potential pathway toward achieving artificial general intelligence. Recent works on multi-modal large language models have demonstrated their effectiveness in handling visual modalities. In this work, we extend the research of MLLMs to point clouds and present the LAMM-Dataset and LAMM-Benchmark for 2D image and 3D point cloud understanding. We also establish an extensible framework to facilitate the extension of MLLMs to additional modalities. Our main contribution is three-fold: 1) We present the LAMM-Dataset and LAMM-Benchmark, which cover almost all high-level vision tasks for 2D and 3D vision. Extensive experiments validate the effectiveness of our dataset and benchmark. 2) We demonstrate the detailed methods of constructing instruction-tuning datasets and benchmarks for MLLMs, which will enable future research on MLLMs to scale up and extend to other domains, tasks, and modalities faster. 3) We provide a primary but potential MLLM training framework optimized for modalities' extension. We also provide baseline models, comprehensive experimental observations, and analysis to accelerate future research.

1 Introduction

Large language models(LLM), notably the GPT series, have made remarkable progress toward achieving general artificial intelligence. The initial GPT-3 [1] model gains its generation ability, world knowledge, and in-context learning through pre-training and further acquires the ability to follow instructions and generalize to unseen tasks through instruction tuning [2]. The latest GPT-3.5 or GPT-4 model has all these powerful abilities and can directly comprehend user intents and generalize to unknown real-world tasks [3, 4]. LLM has become a ubiquitous model for natural language processing tasks. Almost all natural language understanding and generation tasks can be transformed into instruction inputs, enabling a single LLM to perform zero-shot generalization on various downstream applications [5].

Recent works on multi-modal large language models(MLLM), including KOSMOS [6], LLaVA [7], and PaLM-E [8] demonstrate that by introducing other modalities, currently focusing on images, into LLMs through instruction tuning, MLLMs can effectively handle visual modalities and have preliminary abilities to interact with visual content through question-answering dialogue. In these works, LLMs serve as the universal task interface, with inputs from vision tokens provided by pre-

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.

trained multi-modal encoders and language instructions. The powerful modeling capability of LLMs, combined with a unified optimization objective, can help align the model to various modalities [6].

While previous works mainly focused on the image modality, we aim to extend the research of MLLMs to additional modalities, such as point clouds, in this work. We present **LAMM-Dataset**, which emphasizes fine-grained and dense information, and factual knowledge. Additionally, we introduce the **LAMM-Benchmark**, which is the first attempt of a benchmark for MLLMs that offers a comprehensive evaluation of existing models on various computer vision tasks, with two new evaluation strategies designed explicitly for multi-modal language models. We conduct over 200 experiments to provide extensive results and valuable observations on the capabilities and limitations of existing MLLMs. Also, we establish an extensible framework to facilitate the extension of multi-modal language models to additional modalities. Our baseline model surpasses existing multi-modal language models in downstream tasks related to images, demonstrating the effectiveness of our framework and dataset. Finally, we will open-source our codebase, baseline model, instruction tuning dataset, and multi-modal language model benchmark as soon as possible to promote the development of an open research community for MLLMs.

LAMM-Dataset includes an image instruction-tuning dataset containing 186,098 image-language instruction-response pairs and a point cloud instruction-tuning dataset with 10,262 point cloud-language instruction-response pairs. We collect images and point clouds from publicly available datasets and use the GPT API and self-instruction [9] methods to generate instructions and responses based on the original labels from these datasets. The resulting LAMM-Dataset has three appealing properties: 1) Existing multi-modal instruction tuning datasets mainly focus on holistic and rough information. To emphasize fine-grained and dense information, we add more visual information, such as visual relationships and fine-grained categories as input for the GPT API. 2) We observe that existing MLLMs may struggle to understand vision task instructions. To address this, we designed a method to convert vision task annotations into instruction-response pairs, which enhances MLLMs' understanding and generalization of vision task instructions. 3) LAMM-Dataset also includes data pairs for commonsense knowledge question answering by incorporating a hierarchical knowledge graph label system from the Bamboo [10] dataset and the corresponding Wikipedia description.

LAMM-Benchmark evaluates 9 common image tasks, using a total of 11 datasets with over 62,439 samples, and 3 common point cloud tasks, by utilizing 3 datasets with over 12,788 data samples, while existing works only provide quantitative results on fine-tuning and evaluating specific datasets such as ScienceQA, and most works only conduct demonstration or user studies. 1) We are the very first attempt to establish a benchmark for MLLMs. We conducted a comprehensive benchmark to quantify the zero-shot and fine-tuning performance of existing multi-modal language models on various computer vision tasks and compare them against state-of-the-art methods of these tasks, including classification, object detection, pose estimation, visual question answering, facial classification, optical character recognition, object counting. 2) We also attempted two novel evaluation strategies designed explicitly for MLLMs. Specifically, as for text generation, we established a scoring logic based on the GPT API. As for tasks involving interactions between points and images, such as object detection and pose estimation, we proposed an object-locating evaluation method.

LAMM-Framework To validate the effectiveness of LAMM-Dataset and LAMM-Benchmark, we propose a primary but potential MLLM training framework. To avoid modality conflicts caused by introducing multiple modalities, we differentiate the encoder, projector, and LLM finetuning blocks for different modalities in the framework design. Meanwhile, by adding encoders and decoders for other modalities, our framework can flexibly extend to cover more modalities and tasks, such as video understanding, image synthesis, and so on. We provide the results of our baseline models trained using this framework and various observations to accelerate future research.

2 Related Work

Multimodal Large Language Model. With the rapid development of Large Language Models (LLM) such as ChatGPT, GPT-4 [3], many studies manage to explore incorporating other modalities

based on LLM and they can be categorized into two perspectives. **1) System Design Perspective:** Visual ChatGPT [11] and MMREACT [12] invoke various vision foundation models by processing user query to investigate the visual roles of ChatGPT with the help of Visual Foundation Models. ViperGPT [13] instructs LLM to parse visual queries into interpretable steps expressed by Python code. HuggingGPT [14] extends its framework to more modalities by integrating more expert models on Huggingface. **2) End-to-End Trainable Model Perspective:** The other methodology is to connect models for different modalities into an end-to-end trainable model, also known as multimodal large language model. Flamingo [15] proposes a unified architecture for language and vision modeling, while BLIP-2 [16] introduces a Querying Transformer to connect information from image to text modality. Kosmos [6] and PaLM-E [8] build an end-to-end trainable framework on web-scale multi-modal corpora. With the open-sourced LLaMA [17], Mini-GPT4 [18] optimizes a trainable projection matrix only, which connects pre-trained BLIP-2 style vision encoder and large language model, while LLaVA [7] and mPLUG-OwL [19] also finetune LLM. Besides feeding visual info to LLM as input only, LLaMA-Adapter [20], Multi-modal GPT [21] and Otter [22] also integrate multi modal information with intermediate features in LLM.

Instruction Tuning. Instruction tuning [23] is a method proposed to improve the ability of large language models to follow instructions and enhance downstream task performance. Instruction-tuned models like InstructGPT [2], OPT-IML [24], Alpaca [25], have shown promising improvement compared to their based model. The existing instruction tuning datasets are primarily derived from collections of academic datasets like FLAN [23], chatbot data collected from ChatGPT usage such as ShareGPT, or constructed using self-instruction [9] methods like Alpaca. Apart from pure text instruction tuning datasets, Multi-Instruct [26] covers 47 multi-modal tasks. Mini-GPT4 [18] constructs instruction following dataset by composing image-text datasets and handwritten instruction templates. Moreover, LLaVA [7] feeds captions and bounding boxes as the context of COCO images to GPT-4 and therefore get 150K instruction data. Otter [22] builds such instruction tuning datasets from multi-modal MMC4 dataset [27] and incorporates in-contextual examples into instruction tuning by grouping similar instructions together.

In this work, LAMM follows the design of end-to-end trainable MLLM and employs a simple projection layer to connect vision encoder and LLM, which can be easily extended to more modalities. We extend existing vision datasets in 2D instruction datasets and build the first instruction tuning datasets for point cloud, to the best of our knowledge. Furthermore, we evaluated MLLMs on computer vision tasks with quantitative results rather than providing demonstrations only.

3 LAMM-Dataset

We introduce a comprehensive multi-modal instruction tuning dataset, namely LAMM-Dataset. It involves images and point clouds from publicly available datasets for diverse vision tasks, as well as high-quality instructions and responses based on the GPT-4 API and self-instruction methods [9]. To be specific, LAMM-Dataset contains 186K language-image instruction-response pairs, and 10K lanuage-3D instruction-response pairs. Figure 1 provides an overview of its construction process.

We design four kinds of multi-modal instruction-response pairs: 1) *C1: n-round daily dialogue* focuses on multi-modal daily conversations. 2) *C2: n-round factual knowledge dialogue* aims at factual knowledge reasoning. 3) *C3: 1-round detailed description* aims to elaborate images and 3D scenes in texts. 4) *C4: 1-round visual task dialogue* transfers various vision tasks into instruction-response pairs, aiming at enhancing generalizability towards domain tasks in other modalities.

We include diverse 2D/3D vision tasks into LAMM-Dataset, such as captioning, scene graph recognition and VQA that are directly compatible with natural languages, as well as classification, detection, counting and OCR that output labels, bounding boxes, digits and a list of words instead. The former tasks are used in the first three categories, and the latter tasks are used in the fourth category. In LAMM-Dataset, the instruction-response pairs are gathered from 8 image datasets and 4 point cloud datasets, which are referred in Figure 1.

Figure 1: An overview of LAMM dataset. The above demonstrates the process of constructing our Instruction Tuning dataset using the GPT API. By designing different system messages, in-context learning pairs, and queries, we have created the LAMM dataset that covers almost all high-level vision tasks for both 2D and 3D vision. The dataset includes four distinct groups: n-round Daily Dialogue, n-round Factual Knowledge Dialogue, 1-round Detailed Description, and 1-round Visual Dialogue. It is worth noting that for the introduction of vision tasks, we only used the GPT API to generate instruction-response templates and did not directly generate dialogue data. Finally, some examples of the LAMM dataset are presented below, including 2D and 3D scenes and their corresponding instruction-response pairs.

The first three types of instruction-response pairs are generated by inputting several special designed signals to the GPT-4 API, namely system messages, in-context learning pairs and queries: (1) System messages are to inform the GPT-4 API about the task requirements. (2) Several in-context learning pairs are manually annotated to ensure that the rest instruction-response pairs can be generated by a similar fashion. (3) Queries include comprehensive annotations of captions, bounding boxes of objects, relations between objects, factual knowledges from the Bamboo's label system and their Wikipedia descriptions.

The last type of instruction-response pairs also apply the system messages and in-context learning pairs, but use GPT-4 API to generate a pool of templates of instruction-response pairs instead. In this way, ground-truth annotations of many vision tasks, such as object/keypoint detection, OCR, counting and *etc*., can be inserted into these templates, and thus are easier to be converted into reliable language responses, rather than aforementioned query-based conversion.

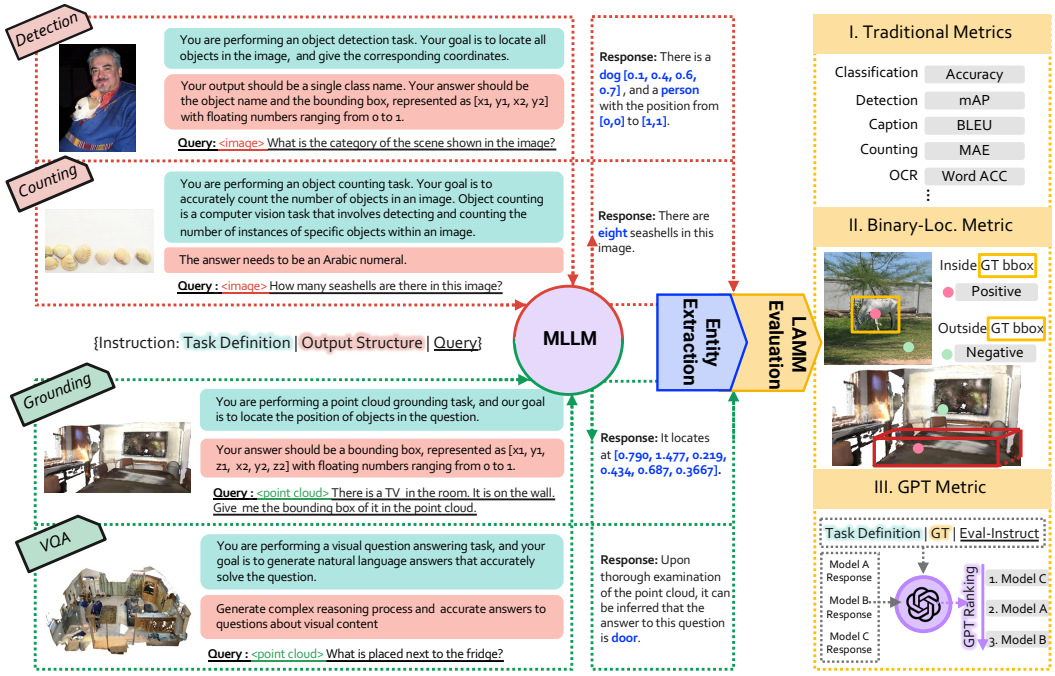

Figure 2: An overview of LAMM Benchmark. It includes both 2D and 3D pipelines, covering multiple computer vision tasks. For each task, we provide the task definition, output structure, and a set of questions as instructions to the MLLM model. Then the entity extraction is applied on the output to extract the key answer. The LAMM Evaluation is used to evaluate the model's performance, which includes traditional metrics, binary-location metric and the GPT Metric.

4 LAMM-Benchmark

Different from LLaVA [7], MiniGPT4 [18] and mPLUG-owl [19] that only provide demos and user studies to qualitatively evaluate the performances of their MLLMs, we propose the LAMM-Benchmark, which instead evaluates the quantitative performance of MLLMs on various 2D/3D vision tasks. It includes an inference pipeline and a set of evaluation metrics. To be specific, the LAMM-Benchmark-2D evaluates 9 common image tasks, using a total of 11 datasets with over 62,439 samples. The LAMM-Benchmark-3D evaluates 3 common point cloud tasks, by utilizing 3 datasets with over 12,788 data samples.

Inference Pipeline. It ensures that the MLLMs can produce reasonable responses that can be fairly evaluated, which includes the way of processing input instructions and the extracting output entities. We construct the Inference Instruction to help the model better understand the task it is performing and the output structure that is required. Inference Instruction includes Task Definition, Output Structure and the usually employed Query Questions, as shown in Figure 2. Inspired by LLaVA [7], we also prompt the MLLM to perform complex reasoning followed by the final answer, so as to obtain a more reliable answer. Then, we employ the Natural Language Toolkit (NLTK) and regular expression matching to extract entities from the output text. These entities act as the results.

Evaluation Metrics. The set of evaluation metrics includes Traditional Metrics, Binary Locating Metric, and GPT Metric. The Traditional Metrics are task-specific metrics from the listed 2D/3D vision tasks, which are the most rigorous to evaluate how MLLMs handle vision tasks. In the Binary Locating Metric, the model needs to output an approximated location of a recognized object through the instruction "output the position of the object", whose result is considered true if it is within the object's groundtruth bounding box. It is a straightforward metric to compare the localization ability of an MLLM model. To evaluate the understanding and question-answering abilitis of MLLM models, we utilize the GPT metric to evaluate the answers' relevance and accuracy to the groundtruth. To be specific, we prompt GPT to rank the responses of multiple MLLM models with the groundtruth results through the instruction described in Figure 2.

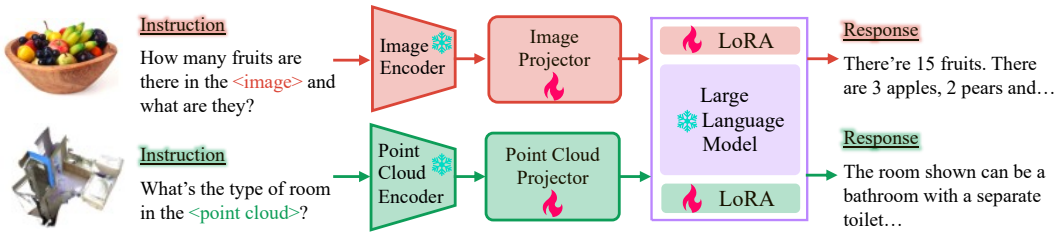

Figure 3: Framework of multi-modality language model. Each modality is encoded by corresponding pre-trained encoder and decoded by LLM. LLM is shared among modalities and trainable projection layers and LoRA parameters are modality-specific.

Evaluation Settings. All 2D/3D vision tasks can be evaluated in a zero-shot manner, where the testing data have no intersection with MLLM's training data. Moreover, we also evaluate the finetuning ability of MLLMs on the test dataset about several mainstream tasks, such as detection, classification and VQA in 2D tasks, as well as detection, grounding and VQA in 3D tasks.

5 Experiments and Results

5.1 LAMM-Framework

The overall framework of our baseline MLLM is depicted in Figure 3. Each modality, image or point cloud, is processed by corresponding encoder, whose features are then projected to the same feature space as the text embeddings by a trainable projection layer. Instructions are directly tokenized by SentencePiece tokenizer [28], then the vision and text tokens are concatenated to feed into the LLM model. To finetune LLM efficiently, we add LoRA [29] parameters to all projection layers in the self-attention layers. LoRA parameters for different vision modalities are not shared. Multimodal tokens are decoded by a shared LLM model and the corresponding LoRA parameters. As shown in Figure 3, only feature projectors and LoRA parameters are optimized during training. We use Vicuna-13B [30], as our LLM. Rank of LoRA modules are set to 32. We train all parameters including projection layers and LoRA modules in a one-stage end-to-end fashion with 4 A100 GPUs.

Input images are resized to be 224×224 and split into 256 patches. We use CLIP [31] pre-trained ViT-L/14 and use image patch features output from transformer layers as image representations. We follow the design of FrozenCLIP [32] to encode point clouds, in which point cloud is tokenized to be 256 tokens by PointNet++ [33] and further encoded by CLIP pretrained ViT-L/14.

5.2 Results on Traditional Metrics

Table 1: Comparison of Multimodal Large Language Models on 2D computer vision tasks.

Task	Dataset	Metric	SOTA	LLaVA[7]	MiniGPT4[18]	mPLUG-owl[19]	LAMM
Classification	CIFAR10 [34]	Acc ↑	99.5[35]	**60.83**	46.22	42.5	34.5
Detection	VOC2012 [36]	mAP ↑	97.2[37]	1.42	0.92	0.158	**4.82**
VQA	SQAimage [38]	Acc ↑	92.53 [7]	40.5	43.43	36.39	**47.15**
	AI2D [39]		N/A	18.13	Failed	19.31	**19.5**
Image Caption	flickr30k [40]	BLEU4 ↑	30.1 [41]	**6.65**	5.1	2.74	0.70
F-g classification	UCMerced [42]	Acc ↑	100[43]	**47**	33.6	32.5	13
Counting	FSC147 [44]	MAE ↓	10.79[45]	56.2	Failed	60.67	**53.97**
OCR	SVT [46]	Word Acc ↑	97.9 [47]	**37.78**	16.97	30.39	4.2
Facial Classification	CelebA(Smile) [48]	Acc ↑	N/A	Failed	**66.36**	Failed	51.3
	CelebA(Hair) [48]		N/A	**46.42**	43.47	40.93	30.48
Keypoints Detection	LSP [49]	PCK ↑	99.5 [50]	Failed	Failed	Failed	Failed

Zero-shot Setting on Image Tasks. Table 1 shows the results of MLLM on 2D vision tasks by the Traditional Metrics. The "SOTA" column lists the results obtained by the best task-specific models. In contrast, the MLLM models were tested in a zero-shot setting. Although MLLM models demonstrated certain abilities of recognizing open-vocabulary classes, understanding images, and answering questions, they performed poorly on tasks involving object localization, including object detection, counting and keypoints detection. *Localization-aware Tasks:* In detection tasks, our LAMM baseline model demonstrated stronger localization ability, but there is still a significant gap between the predicted and the groundtruth bounding boxes. In counting tasks, the MLLM models showed a significant gap between the predicted and ground truth number of objects. MiniGPT4 failed in this task as it is unable to provide a specific number for most of the data. As for the keypoints detection task, we asked the MLLM models to predict the position of each human keypoint in turn. However, all the predicted positions were not in an acceptable range. The MLLM models showed a significant gap in this task, indicating that they have difficulty in accurately predicting keypoint locations. *VQA Tasks:* Our LAMM model demonstrated certain advantages in image understanding and multiple-choice question answering compared to other models. Note that the SOTA is the LLaVA trained on the ScienceQA dataset and evaluated with the GPT-4 answer generation component. But the LLaVA model we compared to was evaluated in the zero-shot setting. Additionally, we removed the random choice process from the LLaVA evaluation to obtain a more straightforward evaluation. *Captioning Tasks:* All MLLM models performed poorly on image captioning. We guess that BLEU4 is not an appropriate metric since longer captions may lead to lower scores. *Classification Tasks:* In fine-grained classification tasks and face classification tasks, all MLLMs performed poorly. Specifically, on the CelebA (Smile) dataset, the LLaVA model gave all "yes" answers, while the mPLUG model randomly gave predictions. However, in the CelebA (hair) dataset, the MLLM models can recognize hair color. These results suggest that the MLLM models may have difficulty in tasks that require fine-grained distinctions. *OCR Tasks:* In OCR tasks, LLaVA can recognize and extract text from images. However, the LAMM model performed poorly on this task.

Fine-tuning Setting on Image Tasks. We also fine-tuned the LAMM baseline model on several vision datasets, including CIFAR10, VOC2012, and SQAimage. The results are shown in Table 2. The fine-tuned baseline achieved an accuracy of 91% on CIFAR10. It also achieved an mAP of 13% on VOC2012, in comparison with 4.8% in the zero-shot setting. These results indicate that our MLLM baseline model can receive the ability of localizing objects after being fine-tuned on detection data.

Zero-shot Setting on Point Cloud Tasks. Table 3 shows the result of LAMM model on 3D scene understanding tasks, under the zero-shot and fine-tuning settings, respectively. The colume “SOTA” presents the best performance on each task, and that for ScanQA is not available as we reformatted it to multiple-choice problem. The results after finetuning are significantly better than the zero-shot setting, in all test tasks. But these results are still inferior to the SOTA models. It is interesting to see that the LAMM finetuned on ScanQA multiple choice data almost achieves 100% accuracy, which may have an overfitting issue due to the narrow training/test gap and small scale of 3D dataset.

Table 2: Results of LAMM model on selected 2D vision tasks. Both zero-shot test result and finetuned results reported. Metrics for classification and VQA is **accuracy**, and that for object detection is **mAP@0.5**.

Task	Dataset	LAMM (Zero-Shot)	LAMM (Finetune)
Classification	CIFAR10 [34]	34.5	91.2
Object Detection	VOC2012 [36]	4.82	13.48
VQA	SQAimage [38]	47.15	74.27

5.3 Results of Binary Locating Metric and GPT Metric

Binary Locating Metric. Table 4 shows the zero-shot results of the MLLM models on the proposed Binary Locating Metric and GPT Metric. The Binary Locating Metric covers the data from VOC2012, FSC147, and LSP. Since the LAMM baseline model has been trained on a small amount of data with detection instructions, it significant improvement in localizing accuracy.

Table 3: Results of 3D tasks by LAMM. Metrics for 3D object detection and visual grounding is **mAP@0.5**, and that for 3D VQA is **accuracy** of multiple choice problem.

Task	Dataset	SOTA	LAMM (Zero-Shot)	LAMM (Finetune)
3D Object Detection	ScanNet[51]	63.2[52]	9.3	11.89
Visual Grounding	ScanRefer[53]	54.59[54]	Failed	3.38
3D VQA	ScanQA[55]	N/A	26.54	99.89

Table 4: Comparison of results of Binary Locating Metric and GPT Metric of existing MLLMs. The Binary-Locating Metric is the accuracy of the predicted position, and the GPT Metric is the score based on the given rank list from GPT.

	LLaVA	MiniGPT4	mPLUG-owl	LAMM
Binary-Loc Metric	14.73	13.12	4.42	**31.2**

	LLaVA	LAMM
GPT Metric	11	**89**

GPT Metric. We calculated GPT scores using a variety of tasks, including VQA, classification, captioning, as well as a small number of detection and counting tasks. As shown in Table 4, the LAMM model outperformed the LLAVA model overall.

5.4 Observation and Analysis

We conducted dozens of experiments and observations on the MLLM model across various tasks to summarize its current capabilities and limitations.

Better Performance in Counting Tasks with Small Number of Objects. As shown in the Table 1, the recent MLLM models are not performing well on counting tasks. In the FSC147 dataset, there are data samples with dozens or even hundreds of objects, and these MLLM models would reply with "I cannot accurately count the number" for such data samples. Therefore, we conducted tests on the subset of the FSC147 dataset with less than 10 objects to evaluate the performance of the models on simple data, as shown in Figure 5 (b). The results show that the language model is able to roughly estimate the number of specified objects in the image, but it is still unable to provide an exact numerical value.

GPT Metric is More Appropriate Than BLEU. Text with too much irrelevant information can lead to low BLEU4 scores, as shown in Table 1. Thus, we adopt the GPT Metric to assess the relevance and accuracy of the model's output captions to the ground truth captions. The results demonstrate that the LAMM outperforms LLaVA, which means the LAMM model is more able to generate high-quality, image-relevant text outputs.

Capable of Object Localization but Struggles with Precise Bounding Box Prediction. We visualize the results of LLaVA on VOC2012 dataset. Figure 4 (a) shows that the LAMM model was able to roughly point out a horse in the image, but was unable to accurately locate the entire object.

LAMM Exhibits Fine-Grained Classification Ability on CIFAR10. As shown in Figure 4, when presented with a 32x32 pixel image of a car, the model's prediction was a more granular category: "Fiat 500L 2012", which accurately identifies the car's brand and model. The left sub figure in 4 (b) shows the image of Fiat 500L 2012 on Autoevolution [56], revealing that it has very similar features to the input image from CIFAR10. These results demonstrate that LAMM has the ability to perform more fine-grained classification, and is capable of recognizing subtle differences in images and assigning them to more specific categories.

Difficulty in Comprehending Visual Information for Shifted Data. We conducted an analysis on several datasets that exhibit significant deviations from the training dataset, including UCMerced, CelebA, and LSP. The UCMerced dataset consists of top-down views of scenes, CelebA is a facial dataset that can describe the expressions and hair colors, and the LSP dataset involves 14 key points of the human body, they are significantly different from the COCO dataset during the training phase. These results suggest that the performance of the MLLM model may degrade significantly on datasets that exhibit significant deviations from the training dataset.

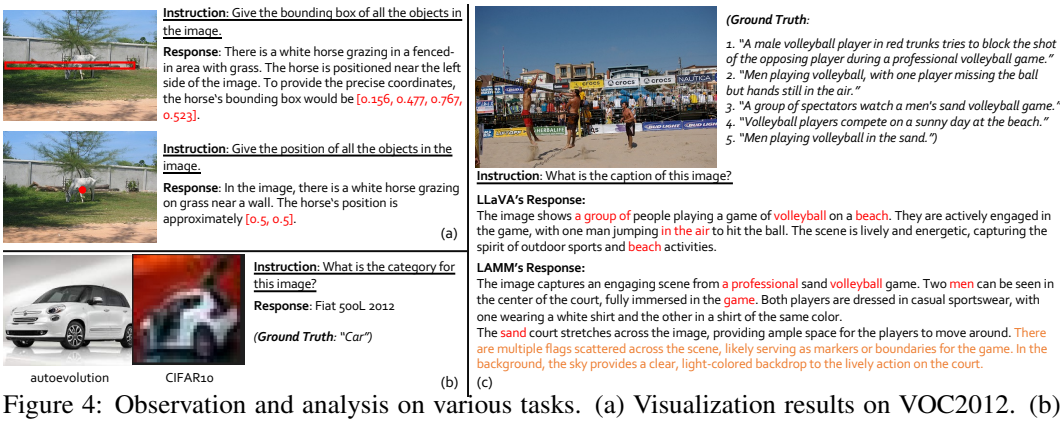

Figure 4: Observation and analysis on various tasks. (a) Visualization results on VOC2012. (b) Visualization results on CIFAR10. The left subfigure is from [56]. (c) Results on flickr30k.

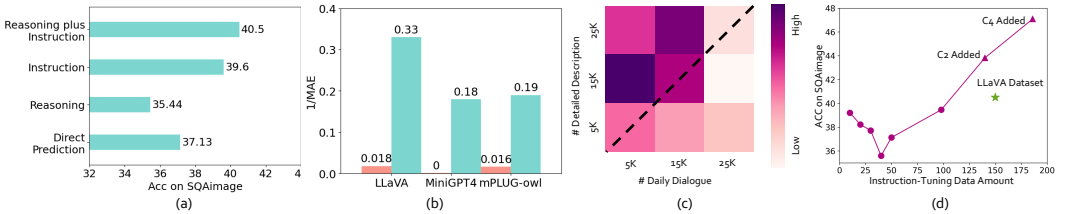

Figure 5: (a) Zero-shot Accuracy of LLaVA with different instructions in test. (b) Counting Performance on FSC147 of MLLMs. (c) Zero-shot accuracy of LAMM trained on various data combinations on SQAimage. (d) Zero-shot accuracy of LAMM model trained additional instruction data in LAMM-Dataset.

Difficulty in Reading Text on SVT data. We analyzed the performance of our LAMM model on the SVT dataset and observed unsatisfactory results in Table 1. A possible explanation is that we used the TextVQA [57] dataset to generate visual task dialogue, which is more geared towards conversational text rather than OCR-related vision tasks. This mismatch in dataset characteristics may have resulted in suboptimal generalization of our model to the SVT dataset. To address this issue, we intend to conduct further investigations and incorporate more appropriate OCR data during the training process to improve our model's performance on OCR-related vision tasks.

Data volume validation on ScienceQA image-only data. As shown in Figure 5, our four types of image instruction tuning datasets outperform LLaVA[7] on all subsets, resulting in a 7% overall performance improvement for the complete LAMM-Dataset. Furthermore, we investigated the impact of sampling *Daily Dialogue* and *Detailed Description* data at different proportions. Notably, even with the small size of 10k examples, our LAMM-Dataset achieved comparable results to LLaVA-Dataset. As the dataset size increased, the overall performance of our model continuously improved, indicating that the LAMM-Dataset is scalable and can be further optimized by adding more data.

6 Conclusion

We present our research on extending multi-modal large language models (MLLMs) to point clouds and introduce the LAMM-Dataset and LAMM-Benchmark for image and point cloud understanding. Our contributions include presenting a large and comprehensive instruction-tuning dataset and benchmark, demonstrating the methods of constructing such datasets and benchmarks for MLLMs, and providing a primary but potential MLLM training framework optimized for modalities' extension. Our research shows that MLLMs can effectively handle visual modalities, including point clouds, and have the potential to generalize to various downstream applications through instruction tuning. By making our codebase, baseline model, instruction tuning dataset, and multi-modal language model benchmark publicly available, we hope to promote the development of an open research community for MLLMs. We believe that our research will facilitate future research on MLLMs and contribute to the development of artificial general intelligence.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[2] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 1, 3

[3] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 1, 2

[4] Hao Fu, Yao; Peng and Tushar Khot. How does gpt obtain its ability? tracing emergent abilities of language models to their sources. *Yao Fu's Notion*, Dec 2022. 1

[5] Yaru Hao, Haoyu Song, Li Dong, Shaohan Huang, Zewen Chi, Wenhui Wang, Shuming Ma, and Furu Wei. Language models are general-purpose interfaces. *arXiv preprint arXiv:2206.06336*, 2022. 1

[6] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023. 1, 2, 3

[7] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 3, 5, 6, 9

[8] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 1, 3

[9] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 2, 3

[10] Yuanhan Zhang, Qinghong Sun, Yichun Zhou, Zexin He, Zhenfei Yin, Kun Wang, Lu Sheng, Yu Qiao, Jing Shao, and Ziwei Liu. Bamboo: Building mega-scale vision dataset continually with human-machine synergy, 2022. 2

[11] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023. 3

[12] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023. 3

[13] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023. 3

[14] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*, 2023. 3

[15] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 3

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 3

[17] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[18] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 3, 5, 6

[19] 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. mplug-owl: Modularization empowers large language models with multimodality, 2023. 3, 5, 6

[20] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 3

[21] Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. Multimodal-gpt: A vision and language model for dialogue with humans. *arXiv preprint arXiv:2305.04790*, 2023. 3

[22] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 3

[23] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 3

[24] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. Opt-iml: Scaling language model instruction meta learning through the lens of generalization, 2023. 3

[25] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023. 3

[26] Zhiyang Xu, Ying Shen, and Lifu Huang. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning. *arXiv preprint arXiv:2212.10773*, 2022. 3

[27] Anas Awadalla, Irena Gao, Joshua Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo, March 2023. 3

[28] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. 6

[29] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6

[30] 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. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. 6

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[32] Xiaoshui Huang, Sheng Li, Wentao Qu, Tong He, Yifan Zuo, and Wanli Ouyang. Frozen clip model is efficient point cloud backbone. *arXiv preprint arXiv:2212.04098*, 2022. 6

[33] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 6

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6, 7

[35] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 6

[36] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 6, 7

[37] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions, 2023. 6

[38] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 6, 7

[39] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth A dozen images. *CoRR*, abs/1603.07396, 2016. 6

[40] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 6

[41] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa, 2019. 6

[42] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010. 6

[43] Andrea Gesmundo. A continual development methodology for large-scale multitask dynamic ml systems, 2022. 6

[44] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[45] Nikola Djukic, Alan Lukezic, Vitjan Zavrtanik, and Matej Kristan. A low-shot object counting network with iterative prototype adaptation, 2022. 6

[46] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International conference on computer vision*, pages 1457–1464. IEEE, 2011. 6

[47] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models, 2022. 6

[48] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018. 6

[49] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference*, 2010. 6

[50] Bruno Artacho and Andreas Savakis. Omnipose: A multi-scale framework for multi-person pose estimation, 2021. 6

[51] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 8

[52] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*, 2023. 8

[53] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, pages 202–221. Springer, 2020. 8

[54] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8

[55] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[56] https://www.autoevolution.com/cars/fiat-500l-2012.html#aeng_fiat-fiat-500l-2012-09l-105-hp-twinair. 8, 9

[57] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 9

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [Yes]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]