PROVISION: Programmatically Scaling Vision-centric Instruction Data for Multimodal Language Models

Jieyu Zhang[1], Le Xue[2], Linxin Song[3], Jun Wang[2], Weikai Huang[1], Manli Shu[2], An Yan[2], Zixian Ma[1], Juan Carlos Niebles[2], Silvio Savarese[2], Caiming Xiong[2], Zeyuan Chen[2], Ranjay Krishna[1]*, Ran Xu[2]*

[1]University of Washington, [2]Salesforce Research, [3]University of Southern California

Code: https://github.com/JieyuZ2/ProVision

Dataset: https://huggingface.co/datasets/Salesforce/ProVision-10M

Abstract

With the rise of multimodal applications, instruction data has become critical for training multimodal language models capable of understanding complex image-based queries. Existing practices rely on powerful but costly large language models (LLMs) or multimodal language models (MLMs) to produce instruction data. These are often prone to hallucinations, licensing issues and the generation process is often hard to scale and interpret. In this work, we present a programmatic approach that employs scene graphs as symbolic representations of images and human-written programs to systematically synthesize vision-centric instruction data. Our approach ensures the interpretability and controllability of the data generation process and scales efficiently while maintaining factual accuracy. By implementing a suite of 24 single-image, 14 multi-image instruction generators, and a scene graph generation pipeline, we build a scalable, cost-effective system: PROVISION *which produces diverse question-answer pairs concerning objects, attributes, relations, depth, etc., for any given image. Applied to Visual Genome and DataComp datasets, we generate over 10 million instruction data points,* PROVISION-10M, *and leverage them in both pertaining and instruction tuning stages of MLMs. When adopted in the instruction tuning stage, our single-image instruction data yields up to a 7% improvement on the 2D split and 8% on the 3D split of CVBench, along with a 3% increase in performance on QBench2, RealWorldQA, and MMMU. Our multi-image instruction data leads to an 8% improvement on Mantis-Eval. Incorporation of our data in both pre-training and fine-tuning stages of xGen-MM-4B leads to an averaged improvement of 1.6% across 11 benchmarks.*

*Corresponding authors.

1. Introduction

The success of Multimodal Language Models (MLMs) such as LLaVA and InstructBLIP has been largely built upon the availability of multimodal data [23, 50, 82], and visual instruction data [20, 54]. In particular, visual instruction data is key to enable MLMs to follow the instruction and respond to user questions about input images effectively. To gather visual instruction data, existing practice mainly relies on powerful Large Language Models (LLMs) or MLMs to generate such data samples [54, 55, 58, 94]. While effective, it does come with certain limitations. First, the generation process remains largely a black-box mechanism, making it difficult to interpret the process and control or customize outputs precisely. Second, even the most advanced LLMs or MLMs are still prone to hallucination [19, 27, 29, 80, 85, 92, 118], generating content that can be factually inaccurate, which undermines the reliability of the resulting visual data and is typically hard to detect and correct ex post. Third, the reliance on powerful LLMs or MLMs might hinder the scalability of the data generation process due to the potential cost (such as API usage costs) and entail license constraints that prevent the use of generated data for model training [72].

In this work, we explore a complementary approach for programmatically generating visual instruction data. To enable programmatic generation, we leverage *scene graphs* [36] as a structured representation of image semantics. We develop programs to systematically generate visual instruction data using automatically extracted scene graph representations from images. In a scene graph, each object is represented as a node, where the attributes of the object—such as color, size, or materials—are assigned directly to that node. The relationships or interactions between these objects are depicted as directed edges connecting the corresponding nodes. Given a scene graph, a program can generate questions like "How many red objects

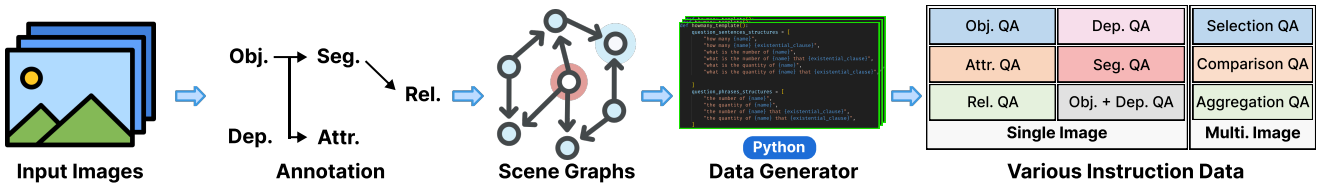

Figure 1. **Overview of PROVISION system**. It can generate a scene graph for any image, enabling programs to synthesize instruction data procedurally from the scene graph. If the scene graph exists, it directly generates instruction data based on it.

are there?" alongside the ground truth answer. Extending this to multiple scene graphs, another program can create comparative questions such as "Which image has the most red objects?", facilitating multi-image instruction data. Notably, this programmatic approach allows us to readily produce both multiple-choice and short-answer questions.

This approach helps to mitigate the limitations of existing LLM/MLM-driven approaches. First, assuming correct scene graphs, combined with human-written programs, introduces transparency and interpretability into the data creation process, and enables a more controlled, customizable output generation, eliminating much of the unpredictability that arises in end-to-end models. Second, programs produce instructions devoid of hallucinations as long as the underlying scene graphs are accurate. Rather than relying on probabilistic outputs from LLMs, our instructions are grounded in the explicit information captured within scene graphs. Furthermore, by shifting the workload from powerful LLMs to programmatic generation, this method enables a scalable and cost-effective solution for data creation. Finally, this approach avoids the licensing constraints associated with LLM- or MLM-generated data, as scene graphs and custom programs do not have to involve proprietary model outputs.

We develop PROVISION, a scalable programmatic system as shown in Figure 1 and introduce PROVISION-10M, a 10 million instructional dataset. PROVISION-10M is created from Visual Genome [36] and DataComp [23]. We implement a total of 24 single-image instruction data generator programs and 14 multi-image instruction data generator programs. These programs cover a diverse range of image-based queries, addressing aspects such as objects, attributes, relations, segmentation, and depth. For Visual Genome, we leverage its manually annotated scene graphs.

Not all datasets come with human-annotated scene graphs. For example, DataComp images do not have ground truth scene graphs like Visual Genome does. To scale PROVISION, we automatically generate scene graphs using a pipeline consisting of state-of-the-art models for object/relation detection [17, 76], image segmentation [79], depth estimation [106], *etc*. The scene graph pipeline automatically generates a scene graph for any image, which the instruction data generator programs can then utilize to produce both single-image and multi-image instructional data.

Using PROVISION-10M, we train various MLMs, experimenting with both pre-training and instruction tuning stages, varying data scales, and different data formats (multiple choice vs. short answer) to evaluate the effectiveness of our generated instruction data. Specifically, our single-image instruction data leads to at most 7% and 8% improvement of CVBench's 2D and 3D splits, around 3% improvement points for QBench2, RealWorldQA, and MMMU, and multi-image instruction data brings 8% point improvement on Mantis-Eval. Our experimental results indicate that programmatically generated instruction data can indeed enhance model performance, but data scale and format are critical factors in achieving optimal results. Moreover, this programmatic instruction data proves effective for both pre-training and instruction tuning stages, and incorporating the data in both stages yields better performance than using it in either stage alone.

2. PROVISION

In this section, we first introduce how we generate vision-centric instruction data programmatically with scene graphs. Then, we present our scene graph generation pipeline that automatically generates a scene graph for any given image.

2.1. Generating instructions programmatically

Augmented scene graph. We first describe the scene graph definition used throughout this work, which is an augmented version of the standard scene graph representation defined in Visual Genome [36], including additional depth and segmentation labels. Given an input image x with size (w, h), which have N objects $\{i_1, \cdots, i_N\}$ and each object i_j has a list of attribute a^j_{attr}. The augmented scene graph is $G = (V, E)$, where $V \subseteq \{i_1, \cdots, i_N\}$, $E = \{(i_j, i_k, a^{jk}_{\text{rel}}) \mid i_j, i_k \in V\}$ and a^{jk}_{rel} is the relation between objects i_j and i_k. Each object i_j has its corresponding bounding box and label pair a^j_{det}, segmentation a^j_{seg}, and a list of attribute a^j_{attr}. Additionally, we add depth annotation a^j_{dep} as an augmented feature.

Single-image visual instruction data. We implement 24 single-image instruction data generators to transform an augmented scene graph into thousands of high-level percep-

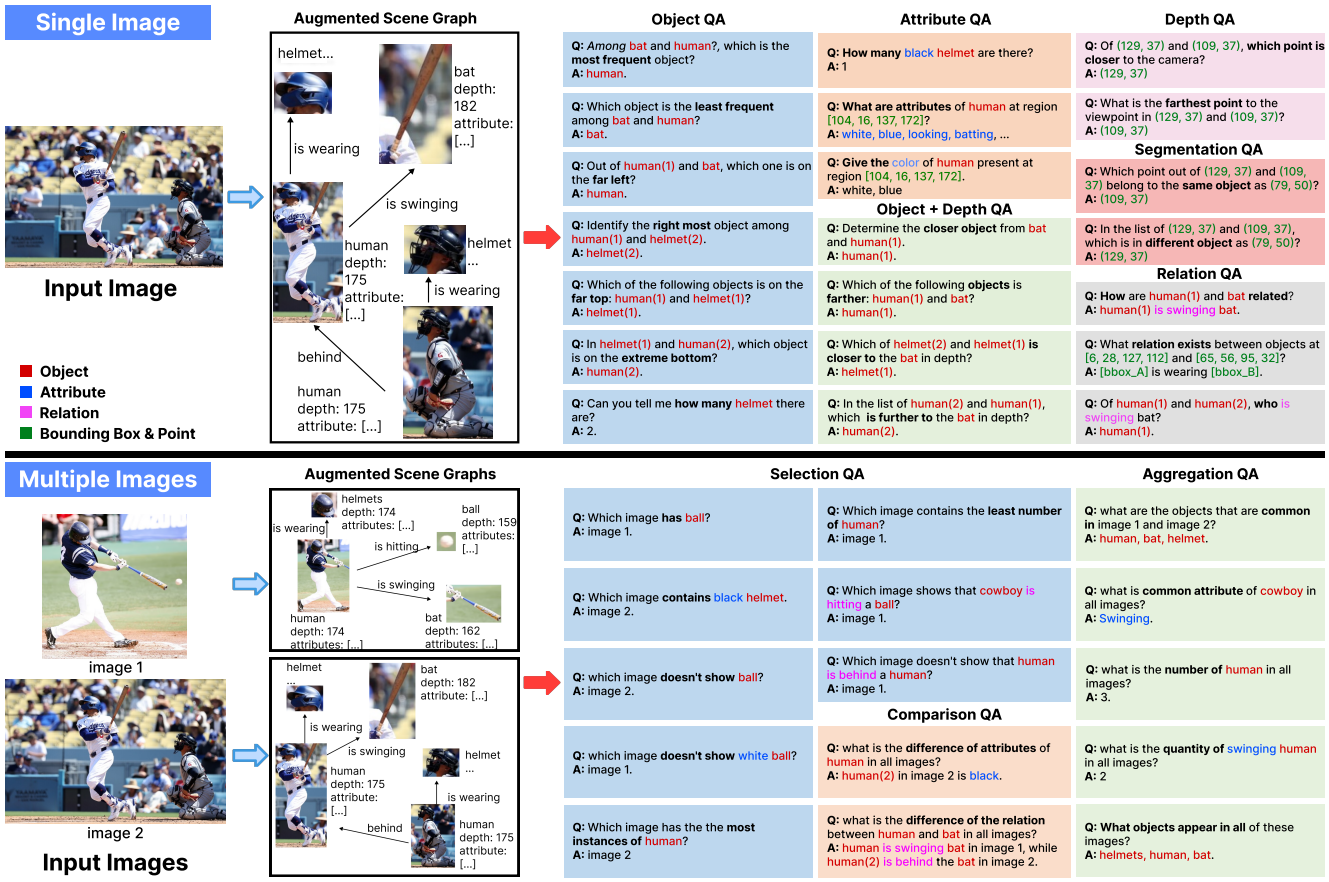

Figure 2. We visualize the instruction data generation process and generated examples for both single and multiple image scenarios. For single image, instruction data can be categorized into six dimensions, aiming to improve the model's ability in retrieving and understanding the annotations. We divide instruction data into three categories for multiple images to help the model understand the relation between different images from the perspective of the relation between their scene graph.

tual question-answer pairs for each image. Each generator utilizes hundreds of pre-defined templates, which systematically integrate these annotations to produce diverse instruction data. These generators are crafted to cover the model's ability to compare, retrieve, and reason about basic visual concepts of objects, attributes, and relations based on the detailed information encoded in each scene graph.

Multi-image visual instruction data. Beyond the singe-image scenarios, we also conduct 14 instruction data generators for multi-image scenarios. While single-image generators focus on producing instruction data from individual scene graphs, multi-image generators are capable of taking multiple scene graphs as input to generate question-answer pairs that span across images. These multi-image generators enable more complex queries, such as selection (e.g., "Which image contains more red objects?"), comparison (e.g., "What are the objects common in these images?"), and aggregation (e.g., "How many red objects in total in these images?") questions. By leveraging multiple scene graphs simultaneously, these generators produce instruction data that encourages models to develop advanced cross-image reasoning skills. We visualize some example instruction data that our data generators can produce in Fig. 2.

Note that while we provide a diverse set of instruction data generators in this work, our system is designed to be highly versatile and readily extendable. It allows users to program additional data generators, expanding the space of instruction data that can be generated. By adding new templates or customizing existing ones, users can introduce novel question-answer pairs, explore new types of visual reasoning tasks, and adapt the system to meet evolving needs. This flexibility makes our framework a scalable tool for creating diverse, high-quality instruction data across various multimodal applications.

2.2. Generating scene graph for any image

We generate a scene graph with object detection, image segmentation, attribute generation, relation generation, and depth estimation. While we utilize state-of-the-art, openly accessible models for each module, our approach is not lim-

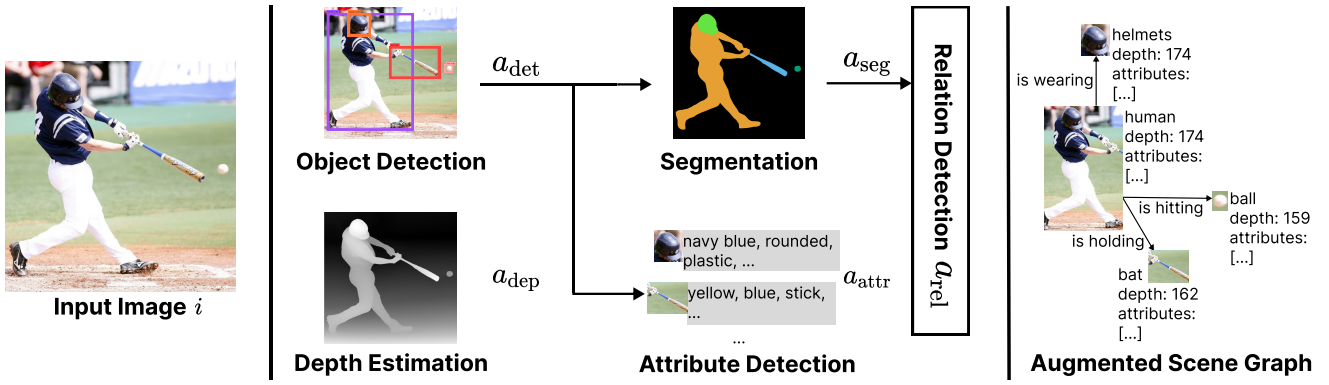

Figure 3. Our scene graph generation pipeline. For each input image, it will generate five annotations, including object, depth, segmentation, attribute, and relation, collectively forming an augmented scene graph.

ited to these specific models. We provide an overview of the scene graph generation pipeline in Fig. 3.

Object detection. We start with object detection to seek the bounding boxes and labels for further annotation methods. The object detection model $f_{\text{det}}(x)$ will annotate all bounding boxes and the corresponding labels of all objects. For example, for object j, the object detection method will output $a^j_{\text{det}} = ([x^j_{\text{min}}, y^j_{\text{min}}, x^j_{\text{max}}, y^j_{\text{max}}], l_j)$, where $(x^j_{\text{min}}, y^j_{\text{min}})$ denotes the left bottom point of the bounding box and l_j denotes the label for object j. In this work, we adopt YOLO-world [17] as our object detection model $f_{\text{det}}(x)$.

Image segmentation. We then adopt image segmentation for better object representations. The image segmentation model $f_{\text{seg}}(x, a_{\text{det}})$, SAM-2 [79] in this work, takes the image x and bounding boxes a_{det} from object detection as input. Specifically, the segmentation will draw the pixel-wise segmentation $a_{\text{seg}} \in \mathbb{R}^{w \times h}$ according to a_{det}.

Attribute detection. Inspired by prior work [122], we fine-tune vision-language models, *i.e.*, CoCa [110] and LLaVA-1.5 [54] as attribute detection models. We construct the training data from LSA, a large-scale attribute dataset [78]. We use its bounding box annotations to crop each object as a single image and use the corresponding attribute annotation as the target output. For LLaVA-1.5, we use "<image> {object_label}" as the prompt template for finetuning data construction. Based on our automatic and manual evaluations, LLaVA-1.5-13B is better than competitors with a precision of 90%, so we adopt it as our attribute detection model. More details of the evaluation and implementation can be found in Appendix.

Relation Detection. We finally retrieve the relations a^{jk}_{rel} for all pairs of objects i_j and i_k in image x according to their segmentation. To achieve this, we pick an finetuned Osprey model [76] as $f_{\text{rel}}(x, a^j_{\text{seg}}, a^k_{\text{seg}})$, which takes the whole image and segmentation of objects i_j and i_k as input, and generate a relation $\tilde{a}^{jk}_{\text{rel}}$. We then ground the generated relation by comparing the similarity between $\tilde{a}^{jk}_{\text{rel}}$ and our relation library and select the top-1 result as the a^{jk}_{rel}.

Depth estimation. Our augmented scene graph also included the pixel-wise depth annotation a_{dep} generated by a depth estimator $f_{\text{dep}}(x)$. In this work, we use Depth Anything V2 [106] as our depth estimation model. The pixel-wise depth annotation can be used to infer the depth of objects for comparing depth among objects.

3. Experiments

In this section, we first describe the instruction data we synthesize in this work, followed by the experimental setup, results, and analysis. We found that 1) Our synthesized instruction data can boost model performance and those from manually-annotated scene graphs are usually better than their counterpart from model-generated scene graphs; 2) The data format (short answer vs. multiple choice) and data scale are important factors to consider for the best performance; and 3) While our data helps when incorporated in either the pre-training or fine-tuning stage, incorporating them in both stages achieves the best performance.

3.1. ProVision-10M Dataset Construction

Leveraging manually-annotated scene graph dataset. We first utilize Visual Genome [36], one large-scale manually-annotated scene graph dataset to construct our instruction data. Specifically, we augment each scene graph with depth and segmentation annotation using Depth Anything V2 and SAM-2. Then, we generate 1.5 million single-image instruction data (**VG-S**) and 4.2 million multi-image instruction data (**VG-M**): For **VG-S**, we sample one instruction data per image and generator, while for **VG-M**, we generate 100,000 samples per generator.

Leveraging generated scene graph. Besides, we sample 120,000 high-resolution images with more than 5 objects from the DataComp dataset [23], and use our scene graph generation pipeline as described in Sec. 2.2 to generate the

	Data Ratio*	Data Format	CVB-2D	CVB-3D	SEED	MMB	MME	QBench2	MMMU	RealWorldQA	Avg.
LLaVA-1.5 instruction data [54]			58.0	61.0	66.8	66.7	63.2	46.4	36.2	54.2	56.6
Replacement	5%	Short Answer	55.0	66.0	67.1	66.3	64.2	48.5	36.7	52.7	57.1
		Multiple Choice	61.0	61.0	**67.5**	67.0	63.3	47.8	37.8	54.6	57.5
		Half-Half	60.0	66.0	67.4	66.5	62.5	46.6	37.4	55.6	57.8
	10%	Short Answer	58.0	67.0	67.0	67.4	64.2	49.2	37.8	56.1	58.3
		Multiple Choice	56.0	62.0	67.2	67.1	**64.4**	48.4	36.8	**58.7**	57.6
		Half-Half	56.0	64.0	67.0	67.3	63.4	46.5	36.7	56.0	57.1
	20%	Short Answer	59.0	66.0	67.3	66.8	63.4	47.7	36.1	54.5	57.6
		Multiple Choice	57.0	63.0	66.8	**68.0**	63.2	48.9	37.2	58.6	57.8
		Half-Half	63.0	66.0	67.5	66.7	62.5	46.7	**39.1**	57.9	**58.7**
	50%	Short Answer	54.0	**69.0**	65.9	64.9	60.5	**50.2**	35.2	55.7	56.9
		Multiple Choice	61.0	68.0	66.3	66.1	61.8	46.1	38.2	56.7	58.0
		Half-Half	**65.0**	**69.0**	66.6	65.0	62.3	47.2	38.1	55.3	58.6
Augmentation	5%	Short Answer	55.0	65.0	66.7	66.5	63.9	48.1	37.3	55.6	57.2
		Multiple Choice	60.0	63.0	66.5	67.7	64.1	47.7	37.8	55.7	57.8
		Half-Half	56.0	66.0	66.8	67.1	61.6	46.5	37.6	56.1	57.2
	10%	Short Answer	59.0	**69.0**	66.7	66.9	62.3	49.0	37.3	54.0	58.0
		Multiple Choice	60.0	64.0	66.4	66.7	63.3	47.6	37.1	56.0	57.6
		Half-Half	57.0	67.0	68.0	**68.1**	64.2	45.3	38.4	55.0	57.9
	20%	Short Answer	59.0	68.0	67.2	68.0	61.6	48.5	37.6	54.0	58.0
		Multiple Choice	58.0	63.0	67.0	67.7	63.3	46.2	37.6	56.6	57.4
		Half-Half	58.0	67.0	67.7	67.2	63.0	46.7	36.4	57.9	58.0
	50%	Short Answer	57.0	**69.0**	67.2	68.0	64.5	**49.4**	37.2	55.3	58.5
		Multiple Choice	61.0	68.0	67.4	67.8	63.3	48.5	36.3	56.6	58.6
		Half-Half	60.0	66.0	67.5	67.6	**66.0**	48.7	**38.9**	57.6	**59.0**

Table 1. Results of instruction tuning LLaVA-1.5-7B with **VG-S**, *i.e.*, single-image instruction data generated from the Visual Genome images and scene graphs. *: data ratio = number of our data added / size of LLaVA-1.5 instruction data.

augmented scene graph for each image. Based on these generated scene graphs, we follow the same process as above to generate 2.3 million single-image instruction data (**DC-S**) and 4.2 million multi-image instruction data (**DC-M**).

In total, these four splits add up to more than 10 million unique instruction data to form PROVISION-10M. For each generated instruction, we store both a multiple-choice and a short-answer version, ensuring flexibility in the types of questions available for model training.

3.2. Experimental Setup

Augmentation vs. replacement. To evaluate the utility of our generated dataset, we adopt two settings: augmentation and replacement. Given a *base dataset* which is an existing dataset used to train MLMs, the augmentation means augmenting the base dataset with our data, while the replacement is to replace a random subset of the base dataset with our data. In particular, we experiment with different augmentation/replacement *ratios*. For example, assume the base data contains 100K samples, an augmentation ratio of 5% indicates including an additional 5K of our data in the training set, while a replacement ratio of 5% means replacing 5K samples of the base data with our data.

Multiple choice vs. short answer. We explore both multiple choice and short answer formats for our generated instruction data, testing three distinct configurations: (1) all data in multiple choice format (multiple choice), (2) all data in short answer format (short answer), and (3) a balanced mix of formats, with half of the data in multiple choice and half in short answer (half-half). These settings allow us to assess the impact of each answer type on model performance, as well as the versatility of the generated data in supporting different response styles.

Model and training recipe. We use LLaVA-1.5 [54] instruction data as the base dataset and its training recipe for instruction tuning LLaVA-1.5-7B model with single-image instruction data; similarity, we follow Mantis [30] for LoRA [25] instruction tuning Mantis-SigLIP-8B with multi-image instruction data and adopt Mantis-Instruct (excluding video-related subsets) as the base dataset. In addition, we experiment with adding our data to both the pre-training and fine-tuning stages of xGen-MM-4B model [104].

Benchmarks. We evaluate models on several popular MLM benchmarks including the following single-image benchmarks: CV-Bench (CVB) [91], SEED-Bench [41, 42], MMBench (MMB) [57], MME [22], QBench2 [101], MMMU [113], RealWorldQA [90], MM-Star [13], MMVet [111]; and multi-image benchmarks: Mantis-Eval [30] and MMT-Bench (MMT) [108].

	Data Ratio*	Data Format	Multi-image benchmark		Single-image benchmark						Avg.
			Mantis-Eval	MMT	SEED	MMB	MME	QBench2	MMMU	RealWorldQA	
Mantis instruction data [30]			54.4	52.9	68.1	72.8	58.5	70.1	44.3	51.5	59.1
Replacement	5%	Short Answer	59.9	52.5	68.4	73.3	**60.0**	70.3	41.4	50.3	59.5
		Multiple Choice	57.6	52.8	68.0	73.0	58.4	70.5	43.7	52.3	59.5
		Half-Half	57.6	53.4	68.1	71.7	57.8	**71.5**	44.4	50.9	59.4
	10%	Short Answer	69.0	**54.8**	68.6	**73.7**	57.4	68.4	41.3	50.7	59.2
		Multiple Choice	59.9	53.1	68.3	71.7	57.5	68.6	45.3	51.9	59.5
		Half-Half	59.0	53.7	68.2	72.6	58.6	68.9	43.1	51.5	59.4
	20%	Short Answer	**62.7**	52.9	68.2	72.9	56.6	70.2	**45.4**	49.7	**59.8**
		Multiple Choice	57.6	52.2	68.0	72.9	57.7	67.4	42.0	51.4	58.6
		Half-Half	58.5	58.6	**68.7**	72.2	59.6	69.4	44.1	51.8	59.7
	50%	Short Answer	57.1	53.2	67.4	70.4	58.6	65.2	42.2	51.2	58.2
		Multiple Choice	55.8	52.5	67.5	69.8	57.5	67.9	42.6	**53.5**	58.4
		Half-Half	54.8	54.0	67.9	72.1	58.2	66.8	43.7	51.5	58.6
Augmentation	5%	Short Answer	60.4	53.8	68.3	71.2	58.9	70.6	**44.3**	48.9	59.5
		Multiple Choice	58.1	54.0	68.1	71.7	58.8	70.3	42.4	50.2	59.2
		Half-Half	58.1	52.5	68.0	71.8	58.4	70.1	41.3	**52.9**	59.1
	10%	Short Answer	60.4	53.0	68.1	72.8	59.2	71.4	44.2	50.6	60.0
		Multiple Choice	60.4	52.7	68.0	72.2	59.2	71.1	43.1	50.2	59.6
		Half-Half	**61.3**	53.0	68.0	72.9	60.0	67.7	42.0	51.0	59.5
	20%	Short Answer	57.1	53.2	68.5	72.3	60.3	**71.6**	43.8	50.3	59.6
		Multiple Choice	58.5	52.9	**68.6**	72.1	**60.6**	**71.6**	43.0	51.5	60.0
		Half-Half	60.4	52.8	68.5	72.4	**60.6**	68.4	43.7	52.7	59.9
	50%	Short Answer	57.6	53.0	68.1	71.6	59.1	70.4	**44.3**	51.2	59.4
		Multiple Choice	58.5	**54.1**	68.4	72.4	58.8	70.1	41.4	51.6	59.4
		Half-Half	60.4	53.7	68.1	**73.4**	60.4	69.7	43.8	51.6	**60.1**

Table 2. Results of instruction tuning Mantis-SigLIP-8B with **VG-M**, *i.e.*, multi-image instruction data generated from the Visual Genome images and scene graphs. *: data ratio = number of our data added / size of Mantis instruction data.

3.3. Instruction Tuning

We exhibit our experiment results by answering the following questions: (1) do scene graphs help produce applicable instructions, and (2) do they need to be real, or can they be automatically generated?

Do scene graphs help produce applicable instructions? This question can be answered affirmatively by Table 1 and Table 2. For single image instructions (Table 1), we compare the model trained with the base dataset and the models trained on four dataset augmentation/replacement ratios and three data formats across eight benchmarking datasets. The results illustrate that, for replacement, (1) instruction tuning the LLaVA-1.5-7B model with **VG-S** data yields improvements over the base dataset (LLaVA-1.5 instruction data) in averaged performance across all settings and achieves the best performance when the replacement ratio at 20%, and (2) on average, model performance is positively related to the amount of replaced multiple choice questions while negatively related to the replacement of short answers. For augmentation, we can see that (1) the model performance on all data formats increases with more data samples from **VG-S** and (2) compared with replacement, augmentation achieves better performance at the same level of data ratio. Overall, results on single image tasks suggest that mixing original data with scene graph-generated short answer and multiple choice format instruction yields competitive results when a substantial portion of the original data is replaced.

For multi-image instruction (Table 2), we test the models on two multi-image benchmarks and six single-image benchmarks. We can observe that for the 20% replacement ratio, the half-half format achieves the highest performance for both multi-image and single-image benchmarks, with an average score of 59.7, showing the benefit of mixing data formats. In contrast, at a 50% replacement ratio, the model's performance generally decreases in both benchmarks, suggesting that excessive replacement with new data may reduce the model's ability to generalize across tasks. For augmentation, multiple choice format stands out with a score of 60.0 on average at 20% of augmentation, while half-half at 50% augmentation achieves the highest average score of 60.1. This suggests that augmentation, especially with mixed data formats, can effectively enhance the model's robustness. Interestingly, augmentation generally provides higher performance stability across both multi-image and single-image benchmarks compared to replacement. On the other hand, half-half format with 10% augmentation and multiple choice format with 20% augmentation show strong performance across multi-image benchmarks (Mantis-Eval and MMT), showing the superiority

of scene graph-generated instruction in helping the model learn how to select, compare, and aggregate features on different images.

Do the scene graphs need to be manually annotated, or can they be model generated? We compare models trained on instruction data from manually annotated (**VG-S**, **VG-M**) and model-generated (**DC-S**, **DC-M**) scene graphs. According to Figure 4 (**VG-S** vs. **DC-S**), **DC-S** underperforms **VG-S** at lower data scales. Interestingly, as the data scale increases to a 50% ratio, **DC-S** achieves comparable performance to **VG-S**, suggesting that larger data scales help mitigate initial performance gaps between data from model-generated and human-curated scene graphs. Moreover, from Figure 5 (**VG-M** vs. **DC-M**), we observe that as the ratio increases, the model performance on the replacement setup grows first and decays later. With the increase of replacement or augmentation ratio, **DC-M** underperforms with **VG-M**, suggesting that on multi-image settings, a larger scale on generated scene graphs may trigger edge effects and not always provide stable performance gain to the model training. In conclusion, instruction data from manually annotated scene graphs is in general better than that from model-generated scene graph, yet both data are able to boost model performance in most cases.

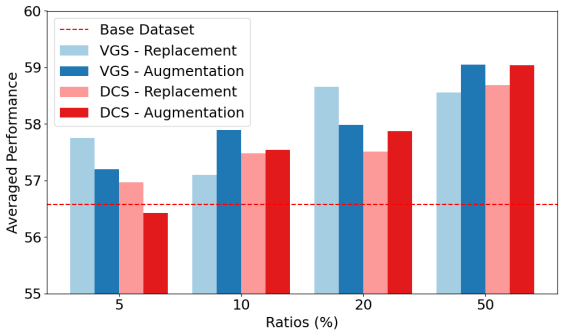

Figure 4. Results of instruction tuning LLaVA-1.5-7B with **DC-S**, *i.e.*, single-image instruction data generated from the DataComp images and model-generated scene graphs.

3.4. Pre-training vs. Instruction Tuning

To assess the benefits of incorporating our data at scale during the pre-training stage, and to compare the effects of adding our data in the pre-training versus fine-tuning stages, we adopt the xGen-MM (BLIP-3) [104] training methodology and use its pre-training data recipe as a foundation. We establish a baseline by pre-training a model on approximately 10 billion tokens using the xGen-MM (BLIP-3) [104] pre-training recipe without our data. Similarly, we apply a baseline fine-tuning recipe of 1 million samples that excludes our data. Details for both recipes are provided in the Appendix.

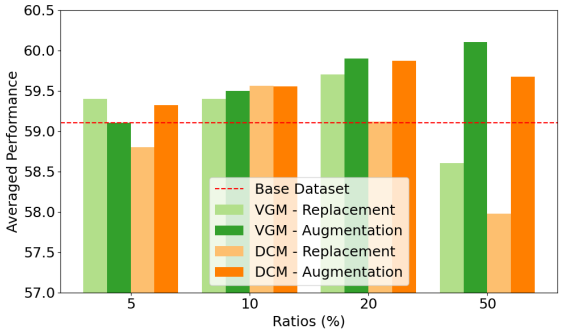

Figure 5. Results of instruction tuning Mantis-SigLIP-8B with **DC-M**, *i.e.*, multi-image instruction data generated from the DataComp images and model-generated scene graphs.

In the augmentation setup mentioned previously, we create recipes that incorporate our data into both the pre-training and fine-tuning stages. **At each stage, we ensure that our additional data accounts for around 5% of the final dataset size**. To examine the impact of different types of scene graph data input on our single-image instruction data generation pipeline, we conduct two sets of experiments: one using human-annotated scene graphs (**VG-S**) as the data source, and another using synthesized scene graphs generated by our pipeline (**DC-S**). Results in table 3 reveal several insights:

1. **Performance Gains with Augmentating Our Data in the Base Recipes**: Both **DC-S** and **VG-S** augmentations consistently improve model performance across the 11 evaluated benchmarks compared to the baseline dataset, which does not include our data. This validates the efficacy of incorporating additional vision-centric knowledge into multimodal foundation model training.
2. **Synergistic Effect of Dual-Stage Augmentation**: The results suggest that augmentation during both pre-training and fine-tuning stages synergistically enhances the performance. For instance, models trained on datasets augmented in both stages achieve the highest average scores (e.g., **DC-S** with +1.2% and **VG-S** with +1.6%), indicating that the synergized effect of dual-stage augmentation outperforms augmentation at either stage alone.
3. **Comparative Effectiveness of VG-S and DC-S**: While both types of data yield improvements, **VG-S**-based augmentation generally provides a marginally higher average score (60.1%) compared to **DC-S** (59.7%) when augmented in both stages. This suggests that cleaner scene graph data source may enable our pipeline to yield even higher performance.

A key advantage of our pipeline is its scalability, enabling its application in large-scale multimodal language model training. We conduct experiments comparing the ad-

Augment in Pre-train	Augment in Fine-tune	CVB-2D	CVB-3D	SEED	MMB	MMStar	MME	QBench2	MMVet	MMMU	RealWorldQA	TextVQA	Avg.
✗	✗	62.9	73.5	70.1	73.9	44.1	62.1	53.9	38.2	41.6	56.2	66.5	58.5
DC-S: DataComp images and model-generated synthetic scene graphs													
✗	✓	64.5	71.9	69.1	73.3	44.8	60.9	**57.9**	35.2	44.1	59.7	67.0	58.9
✓	✗	67.9	70.8	**70.4**	73.5	**45.5**	**64.4**	53.6	37.7	**46.1**	58.8	**67.1**	59.6
✓	✓	**68.3**	**75.5**	69.6	**74.5**	44.4	62.4	54.0	**38.5**	41.9	**61.6**	66.5	**59.7**
VG-S: Visual Genome images and scene graphs													
✗	✓	67.7	72.2	70.0	73.6	**45.5**	64.2	54.4	36.7	42.9	**60.4**	**67.3**	59.5
✓	✗	65.9	73.3	**70.5**	**75.9**	44.8	**64.9**	**56.3**	36.8	42.2	59.2	**67.3**	59.7
✓	✓	**70.2**	**73.7**	69.9	73.9	44.3	64.0	55.7	**40.4**	**44.8**	57.4	66.8	**60.1**

Table 3. Comparison of augmenting the base data recipe with our data (both **DC-S** and **VG-S**) in pre-training, fine-tuning, or both. The first row represents the baseline model's performance, which uses only the base recipes for both stages without our data. The results demonstrate that augmenting the base data recipes with our data in either stage (pre-training or fine-tuning) improves the model's average performance across 11 benchmarks, with augmentation in both stages generally achieving the highest average performance. When augmented with 1.5 million samples from **VG-S**, our model shows an average improvement of approximately 1.4% over the baseline across all benchmarks. Bold values indicate the highest score for each benchmark.

Dataset Augmentation Scale	Avg.
No Augmentation	58.5
0.75 Million	59.1
1.5 Million	**60.1**

Table 4. Impact of dataset augmentation scale on model performance. Under the setup of augmenting data in both pre-training and fine-tuning stages, this table compares the model's performance across the same 11 benchmarks as Table 4, with varying scales of data augmentation during pre-training: no augmentation, 0.75 million samples, and 1.5 million samples from **VG-S**.

dition of 0.75 million versus 1.5 million samples of **VG-S** during pre-training, while keeping the fine-tuning recipe consistent. As shown in Table 4, scaling the inclusion of **VG-S** in pre-training stage from 0.75 million to 1.5 million samples yields notable gains in average performance across 12 benchmarks, increasing from 61.4% to 62.3%. This result underscores the potential of our data to enhance model capabilities in large-scale multimodal foundation model training.

4. Related Work

We contextualize our work on the recent rise of MLMs and approaches of synthesizing data for MLMs.

Multimodal language models (MLMs). In recent years, MLMs, by integrating visual encoders within various pre-trained large languages models [2, 6, 10, 11, 14, 26, 43, 49, 49, 56, 63, 64, 69, 77, 83, 86–89, 93, 97, 98, 104], have progressively driven advancements in visual-language learning. With ubiquitous open-sourced LLM backbones and the increasing data for visual instruction tuning. Models like Blip series [20, 45, 46, 75, 104], QwenVL series [5, 96], LLaVA series [52–54], InternVL series [15, 16], etc. have achieved unprecedented visual understanding performance in nearly all kind of visual tasks. However, recent works like Task Me Anything [115], CVBench (Cambrian-1) [91] show that while MLMs are adept at high-level semantic understanding, they surprisingly underperform in vision-centric tasks (e.g. depth estimation, counting, localization, etc). Furthermore, the availability of instruction data for vision-centric tasks remains limited compared to other multimodal data such as image captions, due to the high cost of collection and annotation.

Synthetic data for MLMs. Synthetic data has increasingly been used for pretraining and finetuning [4, 8, 21, 40, 47, 55, 58, 70, 71, 75, 99, 103, 107, 109, 116, 120, 121] of large language models (LLMs), leading to notable improvements in reasoning, instruction following, and other tasks. Similarly, synthetic data has been integrated into multimodal language model (MLM) development, including approaches like model-generated instruction data [54, 55, 58, 94] and synthetic captions [47, 59, 81, 105]. However, current methods largely focus on synthetic data generation using LLM, MLM, and diffusion models. Programmatic/procedural methods have also been employed to generate multimodal data, such as in GQA [28], AGQA [24], and Task Me Anything [115], yet these are often designed primarily for evaluation or as contributions to a final dataset. In contrast, our approach centers on the data generation process itself, producing single- and multi-image instruction data adaptable to any image source for training purposes.

5. Conclusion

Our PROVISION system programmatically synthesizes vision-centric instruction data for training MLMs by leveraging scene graph representations and human-written programs. Applied to Visual Genome and DataComp, PROVISION produces PROVISION-10M, a dataset of over 10 million instruction data, which we leverage in both pretraining and instruction tuning stages of MLMs, resulting in notable

performance improvements and demonstrating the potential of programmatically scaling vision-centric instruction data in advancing MLM capabilities.

Limitations and future directions. Limitations of PROVISION include its reliance on the quality and completeness of scene graphs, as well as its dependency on human-written programs. Future work could address these by enhancing the scene graph generation pipeline to enable more accurate data synthesis and by developing automated program synthesis, leveraging LLMs to further scale data generation.

References

[1] ajibawa 2023. Python-code-23k-sharegpt. https://huggingface.co/datasets/ajibawa-2023/Python-Code-23k-ShareGPT. 2

[2] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models, 2023. 8

[3] Anas Awadalla, Le Xue, Oscar Lo, Manli Shu, Hannah Lee, Etash Kumar Guha, Matt Jordan, Sheng Shen, Mohamed Awadalla, Silvio Savarese, et al. Mint-1t: Scaling open-source multimodal data by 10x: A multimodal dataset with one trillion tokens. *arXiv preprint arXiv:2406.11271*, 2024. 1

[4] Anas Awadalla, Le Xue, Manli Shu, An Yan, Jun Wang, Senthil Purushwalkam, Sheng Shen, Hannah Lee, Oscar Lo, Jae Sung Park, Etash Guha, Silvio Savarese, Ludwig Schmidt, Yejin Choi, Caiming Xiong, and Ran Xu. Blip3-kale: Knowledge augmented large-scale dense captions, 2024. 8, 1

[5] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 8

[6] Jing Bi, Nguyen Manh Nguyen, Ali Vosoughi, and Chenliang Xu. Misar: A multimodal instructional system with augmented reality, 2023. 8

[7] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusiñol, Ernest Valveny, C. V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. *arXiv preprint arXiv: 1905.13648*, 2019. 2

[8] Yung-Chieh Chan, George Pu, Apaar Shanker, Parth Suresh, Penn Jenks, John Heyer, and Sam Denton. Balancing cost and effectiveness of synthetic data generation strategies for llms. 2024. 8

[9] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021. 1

[10] Guo Chen, Yin-Dong Zheng, Jiahao Wang, Jilan Xu, Yifei Huang, Junting Pan, Yi Wang, Yali Wang, Yu Qiao, Tong Lu, and Limin Wang. Videollm: Modeling video sequence with large language models, 2023. 8

[11] Houlun Chen, Xin Wang, Hong Chen, Zihan Song, Jia Jia, and Wenwu Zhu. Grounding-prompter: Prompting llm with multimodal information for temporal sentence grounding in long videos, 2023. 8

[12] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023. 1

[13] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? *CoRR*, abs/2403.20330, 2024. 5

[14] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. Pali-3 vision language models: Smaller, faster, stronger, 2023. 8

[15] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. 8

[16] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 8

[17] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911, 2024. 2, 4

[18] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv: 2110.14168*, 2021. 2

[19] Chenhang Cui, Yiyang Zhou, Xinyu Yang, Shirley Wu, Linjun Zhang, James Zou, and Huaxiu Yao. Holistic analysis of hallucination in gpt-4v(ision): Bias and interference challenges, 2023. 1

[20] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 8

[21] Qingxiu Dong, Li Dong, Xingxing Zhang, Zhifang Sui, and Furu Wei. Self-boosting large language models with synthetic preference data. 2024. 8

[22] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024. 5

[23] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 4

[24] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 8

[25] 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. In *International Conference on Learning Representations*, 2022. 5

[26] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. *arXiv preprint arXiv:2311.18445*, 2(3):9, 2023. 8

[27] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023. 1

[28] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 8

[29] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. 1

[30] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning, 2024. 5, 6, 4

[31] Kushal Kafle, Scott Cohen, Brian Price, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *CVPR*, 2018. 2

[32] Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. Geomverse: A systematic evaluation of large models for geometric reasoning. *arXiv preprint arXiv: 2312.12241*, 2023. 2

[33] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision – ECCV 2016*. Springer International Publishing, 2016. 2

[34] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384, 2017. 2

[35] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624, 2020. 2

[36] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 1, 2, 4

[37] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[38] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024. 1

[39] Hugo Laurençon, Léo Tronchon, and Victor Sanh. Unlocking the conversion of web screenshots into html code with the websight dataset, 2024. 1

[40] Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Krishna Anumanchipalli, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. Llm2llm: Boosting llms with novel iterative data enhancement. In *Annual Meeting of the Association for Computational Linguistics*, 2024. 8

[41] Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench-2: Benchmarking multimodal large language models. *arXiv preprint arXiv:2311.17092*, 2023. 5

[42] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 5

[43] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning, 2023. 8

[44] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1

[45] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 8

[46] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 8

[47] Xiaotong Li, Fan Zhang, Haiwen Diao, Yueze Wang, Xinlong Wang, and Ling-Yu Duan. Densefusion-1m: Merging

vision experts for comprehensive multimodal perception. *2407.08303*, 2024. 8

[48] Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". Openorca: An open dataset of gpt augmented flan reasoning traces. https://https://huggingface.co/Open-Orca/OpenOrca, 2023. 2

[49] Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, Ce Liu, and Lijuan Wang. Mm-vid: Advancing video understanding with gpt-4v(ision), 2023. 8

[50] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[51] Fangyu Liu, Guy Edward Toh Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 2023. 2

[52] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 8, 2

[53] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 1

[54] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1, 4, 5, 8, 3

[55] Jihao Liu, Xin Huang, Jinliang Zheng, Boxiao Liu, Jia Wang, Osamu Yoshie, Yu Liu, and Hongsheng Li. Mm-instruct: Generated visual instructions for large multimodal model alignment, 2024. 1, 8

[56] Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. Prismer: A vision-language model with multi-task experts, 2024. 8

[57] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 5

[58] Yangzhou Liu, Yue Cao, Zhangwei Gao, Weiyun Wang, Zhe Chen, Wenhai Wang, Hao Tian, Lewei Lu, Xizhou Zhu, Tong Lu, et al. Mminstruct: A high-quality multi-modal instruction tuning dataset with extensive diversity. *arXiv preprint arXiv:2407.15838*, 2024. 1, 8

[59] Zheng Liu, Hao Liang, Xijie Huang, Wentao Xiong, Qinhan Yu, Linzhuang Sun, Chong Chen, Conghui He, Bin Cui, and Wentao Zhang. Synthvlm: High-efficiency and high-quality synthetic data for vision language models. *ArXiv*, abs/2407.20756, 2024. 8

[60] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, 2021. 2

[61] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *NeurIPS Datasets and Benchmarks*, 2021. 2

[62] Pan Lu, Swaroop Mishra, Tanglin 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 *Advances in Neural Information Processing Systems*, pages 2507–2521. Curran Associates, Inc., 2022. 2

[63] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models, 2023. 8

[64] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration, 2023. 8

[65] Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, 2022. Association for Computational Linguistics. 2

[66] Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. Infographicvqa. *arXiv preprint arXiv: 2104.12756*, 2021. 2

[67] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2199–2208, 2021. 2

[68] Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-math: Unlocking the potential of slms in grade school math, 2024. 2

[69] Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Cyril Zakka, Yash Dalmia, Eduardo Pontes Reis, Pranav Rajpurkar, and Jure Leskovec. Med-flamingo: a multimodal medical few-shot learner, 2023. 8

[70] Avanika Narayan, Mayee F. Chen, Kush Bhatia, and Christopher R'e. Cookbook: A framework for improving llm generative abilities via programmatic data generating templates. 2024. 8

[71] Nihal V. Nayak, Yiyang Nan, Avi Trost, and Stephen H. Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation. *ArXiv*, abs/2402.18334, 2024. 8

[72] OpenAI. Terms of use. Accessed: 2024-11-01. 1

[73] OpenGVLab. Sharegpt-4o. = https://huggingface.co/datasets/OpenGVLab/ShareGPT-4o. 1

[74] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011. 1

[75] Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq R. Joty, Ran Xu, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *ArXiv*, abs/2311.18799, 2023. 8

[76] Jae Sung Park, Zixian Ma, Linjie Li, Chenhao Zheng, Cheng-Yu Hsieh, Ximing Lu, Khyathi Chandu, Norimasa Kobori Quan Kong, Ali Farhadi, and Ranjay Krishna Yejin Choi. Robin: Dense scene graph generations at scale with improved visual reasoning. 2024. 2, 4

[77] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world, 2023. 8

[78] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Improving closed and open-vocabulary attribute prediction using transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 4, 1

[79] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 4

[80] Mobashir Sadat, Zhengyu Zhou, Lukas Lange, Jun Araki, Arsalan Gundroo, Bingqing Wang, Rakesh R Menon, Md Rizwan Parvez, and Zhe Feng. Delucionqa: Detecting hallucinations in domain-specific question answering. *arXiv preprint arXiv:2312.05200*, 2023. 1

[81] Sahand Sharifzadeh, Christos Kaplanis, Shreya Pathak, Dharshan Kumaran, Anastasija Ilic, Jovana Mitrovic, Charles Blundell, and Andrea Banino. Synth2: Boosting visual-language models with synthetic captions and image embeddings. *ArXiv*, abs/2403.07750, 2024. 8

[82] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2556–2565. Association for Computational Linguistics, 2018. 1

[83] Mustafa Shukor, Corentin Dancette, Alexandre Rame, and Matthieu Cord. Unival: Unified model for image, video, audio and language tasks, 2023. 8

[84] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 2

[85] Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. On early detection of hallucinations in factual question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2721–2732, 2024. 1

[86] Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Fine-grained audio-visual joint representations for multimodal large language models, 2023. 8

[87] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners, 2024.

[88] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality, 2024.

[89] Yunlong Tang, Jinrui Zhang, Xiangchen Wang, Teng Wang, and Feng Zheng. Llmva-gebc: Large language model with video adapter for generic event boundary captioning, 2023. 8

[90] X.ai Team. Grok-1.5 vision preview. https://x.ai/blog/grok-1.5v, 2024. 5

[91] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms, 2024. 5, 8, 1, 2

[92] Priyesh Vakharia, Devavrat Joshi, Meenal Chavan, Dhananjay Sonawane, Bhrigu Garg, and Parsa Mazaheri. Don't believe everything you read: Enhancing summarization interpretability through automatic identification of hallucinations in large language models. *arXiv preprint arXiv:2312.14346*, 2023. 1

[93] Junke Wang, Dongdong Chen, Chong Luo, Xiyang Dai, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang. Chatvideo: A tracklet-centric multimodal and versatile video understanding system, 2023. 8

[94] Junke Wang, Lingchen Meng, Zejia Weng, Bo He, Zuxuan Wu, and Yu-Gang Jiang. To see is to believe: Prompting gpt-4v for better visual instruction tuning. *arXiv preprint arXiv:2311.07574*, 2023. 1, 8

[95] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset, 2024. 2

[96] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 8

[97] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling video foundation models for multimodal video understanding. *ArXiv*, abs/2403.15377, 2024. 8

[98] Zhanyu Wang, Longyue Wang, Zhen Zhao, Minghao Wu, Chenyang Lyu, Huayang Li, Deng Cai, Luping Zhou, Shuming Shi, and Zhaopeng Tu. Gpt4video: A unified multimodal large language model for lnstruction-followed understanding and safety-aware generation, 2023. 8

[99] Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T. Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. Codeclm: Aligning language models with tailored synthetic data. In *NAACL-HLT*, 2024. 8

[100] wendlerc. Renderedtext. https://huggingface.co/datasets/wendlerc/RenderedText. 2

[101] Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Chunyi Li, Wenxiu Sun, Qiong Yan, Guangtao Zhai, and Weisi Lin. Q-bench: A benchmark for general-purpose foundation models on low-level vision. In *ICLR*, 2024. 5

[102] Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, dingnan jin, Yu Cheng, Qifan Wang, and Lifu Huang. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. *Annual Meeting of the Association for Computational Linguistics*, 2024. 2

[103] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *ArXiv*, abs/2406.08464, 2024. 8

[104] Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, Shrikant B. Kendre, Jieyu Zhang, Can Qin, Shu Zhen Zhang, Chia-Chih Chen, Ning Yu, Juntao Tan, Tulika Awalgaonkar, Shelby Heinecke, Huan Wang, Yejin Choi, Ludwig Schmidt, Zeyuan Chen, Silvio Savarese, Juan Carlos Niebles, Caiming Xiong, and Ran Xu. xgen-mm (blip-3): A family of open large multimodal models. *ArXiv*, abs/2408.08872, 2024. 5, 7, 8, 1, 2

[105] An Yan, Zhengyuan Yang, Junda Wu, Wanrong Zhu, Jianwei Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Julian McAuley, Jianfeng Gao, et al. List items one by one: A new data source and learning paradigm for multimodal llms. *arXiv preprint arXiv:2404.16375*, 2024. 8, 1, 2

[106] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 2, 4

[107] Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. *arXiv preprint arXiv:2409.07431*, 2024. 8

[108] Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, Jiayi Lei, Quanfeng Lu, Runjian Chen, Peng Xu, Renrui Zhang, Haozhe Zhang, Peng Gao, Yali Wang, Yu Qiao, Ping Luo, Kaipeng Zhang, and Wenqi Shao. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi, 2024. 5

[109] Huimu Yu, Xing Wu, Weidong Yin, Debing Zhang, and Songlin Hu. Codepmp: Scalable preference model pretraining for large language model reasoning. 2024. 8

[110] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. 2022. 4

[111] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *Forty-first International Conference on Machine Learning*. 5

[112] Ye Yuan, Xiao Liu, Wondimu Dikubab, Hui Liu, Zhilong Ji, Zhongqin Wu, and Xiang Bai. Syntax-aware network for handwritten mathematical expression recognition. *CVPR*, 2022. 2

[113] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024. 5

[114] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[115] Jieyu Zhang, Weikai Huang, Zixian Ma, Oscar Michel, Dong He, Tanmay Gupta, Wei-Chiu Ma, Ali Farhadi, Aniruddha Kembhavi, and Ranjay Krishna. Task me anything. In *Thirty-Eighth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 8

[116] Chenyang Zhao, Xueying Jia, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. Self-guide: Better task-specific instruction following via self-synthetic finetuning. *ArXiv*, abs/2407.12874, 2024. 8

[117] Yilun Zhao, Chen Zhao, Linyong Nan, Zhenting Qi, Wenlin Zhang, Xiangru Tang, Boyu Mi, and Dragomir Radev. Robut: A systematic study of table qa robustness against human-annotated adversarial perturbations. *arXiv preprint arXiv: 2306.14321*, 2023. 2

[118] Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. Beyond hallucinations: Enhancing lvlms through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*, 2023. 1

[119] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment. *arXiv preprint arXiv: 2305.11206*, 2023. 2

[120] Fan Zhou, Zengzhi Wang, Qian Liu, Junlong Li, and Pengfei Liu. Programming every example: Lifting pre-training data quality like experts at scale. 2024. 8

[121] Kun Zhou, Beichen Zhang, Jiapeng Wang, Zhipeng Chen, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, and Ji-Rong Wen. Jiuzhang3.0: Efficiently improving mathematical reasoning by training small data synthesis models. *ArXiv*, abs/2405.14365, 2024. 8

[122] William Y Zhu, Keren Ye, Junjie Ke, Jiahui Yu, Leonidas Guibas, Peyman Milanfar, and Feng Yang. Artvlm: Attribute recognition through vision-based prefix language modeling. *arXiv preprint arXiv:2408.04102*, 2024. 4

PROVISION: Programmatically Scaling Vision-centric Instruction Data for Multimodal Language Models

Supplementary Material

6. Fine-tuning attribute detector

Dataset preparation. We adopt the LSA dataset [78] for training attribute detectors for our scene graph generation pipeline. We first filter out bounding boxes whose size is less than 25 pixels. Then we normalize the attributes by (1) removing non-attributes like "world" and (2) merging attributes like "gray" and "grey". We also remove objects with conflicting attributes at the same time like "big" and "small". Finally, for each object category with more than 10 instances in the dataset, we sample 5 instances to compose the test set (42,558 objects) and use the remaining as the training set (3,679,514 objects).

Fine-tuning and evaluation. For CoCa model, we use the OpenCLIP codebase to fine-tune a ViT-L-14 CoCa pretrained on LAION-2B. For LLaVA-1.5 model, we use the official codebase to fine-tune both LLaVA-1.5-7B and LLaVA-1.5-13B. For evaluation, we report the average number of predicted attributes of each model. Besides, we report the precision and recall of the model output against the provided labels. However, because the LSA dataset is noisy and incomplete, we sample 200 data from the test set and manually evaluate whether each predicted attribute is correct or not to calculate the human precision.

Results. The results are in Table 5. From the results, we can see that the CoCa model outputs more attributes (3.19) than LLaVA-1.5 models (1.13), because the LLaVA-1.5 models take the object label as input and is likely to focus on the object in the cropped image while CoCa only inputs the cropped image and may output attributes irrelevant to the centered object. In addition, we found that LLaVA-1.5-13B is better than LLaVA-1.5-7B and CoCa for all the evaluation metrics, so we adopt LLaVA-1.5-13B for our scene graph generation pipeline.

We will release both train/test dataset and the trained models.

7. Instruction Tuning Experiments

For fine-tuning LLaVA-1.5 models, we reuse the fine-tuning script in the official Github repository: https://github.com/haotian-liu/LLaVA/blob/main/scripts/v1_5/finetune.sh.

For fine-tuning Mantis-SigLIP-8B, we reuse the script from the official Github repository: https://github.com/TIGER-AI-Lab/Mantis/blob/main/mantis/train/scripts/train_mllava.sh.

8. xGen-MM (BLIP-3) Experiments Recipes

8.1. Pre-training Recipes

Base Recipe. Following xGen-MM(BLIP-3) [104], the base pre-training recipe includes the following – **Caption Datasets:** Datacomp [23] (10%), BLIP3-KALE [4] (60%), BLIP3-OCR [104] (10%), BLIP3-GROUNDING [104] (10%), CC12M [9] (2.5%), CC3M [9] (2.5%), VG [36] (2.5%), and SBU [74] (2.5%). **Interleaved Datasets:** OBELICS [37] (35%), MINT-1T-HTML [3] (35%), MINT-1T-PDF [3] (25%), and MINT-1T-ArXiv [3] (5%).

The baseline model is trained using 24 H100-80GB GPUs for 8,000 steps. At each step, data is sampled with equal probability from either the **Caption Datasets** or the **Interleaved Datasets** bucket (50% each). Within each bucket, datasets are sampled according to the probabilities listed above.

The batch sizes are configured as follows: **Caption Datasets:** Batch size of 300 per GPU. **Interleaved Datasets:** Batch size of 50 per GPU.

Augmented Recipe with PROVISION Data. To ensure a fair comparison, the composition of the **Caption Datasets** and **Interleaved Datasets** from the base recipe is preserved. Additionally, a new dataset bucket, **PROVISION**, is introduced. The sampling ratios are adjusted from 50%:50% (Caption Datasets vs Interleaved Datasets) to 47.5%:47.5%:5% (Caption Datasets vs Interleaved Datasets vs PROVISION).

The training duration is extended from 8,000 steps to 8,500 steps to ensure that the amount of caption and interleaved data remains consistent while incorporating the PROVISION data. All other training configurations are kept unchanged, allowing a fair assessment of the impact of including PROVISION data.

8.2. Fine-tuning Recipes

We use a mixture of open-source supervised fine-tuning datasets [38, 44, 53, 91, 105] as our base SFT data recipe. The base SFT data recipe contains around 1.2M single-image QA samples. We create the base SFT recipe to cover various visual tasks including:

- **General visual question answering (630K)**: sharegpt4v [12], sharegpt4o [73], websight [39],

Model	Avg. number of predicted attributes	Precision	Recall	Human precision
CoCa	3.19	14.5	36.0	53.1
LLaVA-1.5-7B	1.13	57.3	50.4	90.5
LLaVA-1.5-13B	1.13	58.4	51.5	91.9

Table 5. Results of fine-tuned attribute detector.

hateful_memes [35], vision_flan [102], llava-150K [52], SoM-llava [105], vsr [51]

- **OCR, Document and chart understanding (270K)**: DocVQA [67], ChartQA [65], AI2D [33], DVQA [31], stvqa [7], infographicVQA [66], rendered_text [100], TextVQA [84], RobuT Sqa [117], HiTab, RobuT wikisql [117], vistext, chart2text, arxivQA, hme100k [112]
- **Math, science (280K)**: iconqa [61], intergps [60], tqa [34], geomverse [32], raven [114], mathvision [95], scienceQA [62], cambrian-data-engine [91]
- **Text-only SFT data (45K)**: gsmk8k [18], slimorca [48], orca-math-word-problems [68], Python-Code-23k-ShareGPT [1], lima [119]

Fine-tuning details. We fine-tune the xGen-MM(BLIP-3) [104] using their official training code [1] with 8 H100 GPUs. We adopt the default training configuration as the original BLIP-3 model [2] and fine-tune our models for one epoch across all experiments. Please refer to the official BLIP-3 codebase for training details.

9. Instruction data example

In the current version of ProVision, we implement 24 single-image instruction data generators and 14 multi-image instruction data generators. We provide examples for both single-image instruction data (Table 6) and multi-image instruction data (Table 7).

10. Raw results of Figure 4 and Figure 5

We provide the raw results of Figure 4 and Figure 5, i.e., the evaluation results of models that were fine-tuned with DataComp images with our automatic annotation and scene graph generation pipeline (Table 8 and Table 9).

[1] https://github.com/salesforce/LAVIS/tree/xgen-mm

[2] https://github.com/salesforce/LAVIS/blob/xgen-mm/scripts/example_finetune_xgenmmv1-phi3_mini_4k_instruct.sh

Task generator	Example question	Example answer
ExistsObjectGenerator	Tell me what is the number of stop sign that you see?	1
MostObjectGenerator	Determine from boat and sail, which object is the most commonly found?	boat
LeastObjectGenerator	Among door and drawer, what is the least frequent object?	door
LeftMostObjectGenerator	Can you tell among jeans, pants, and horses, which object is located on the far left?	pants
RightMostObjectGenerator	Can you tell among girl, hat, and sign, which object is positioned on the far right?	hat
TopMostObjectGenerator	Identify which object is located on the most upward, among cloud, floor, and water?	cloud
BottomMostObjectGenerator	Provide the extreme bottom object among dome, flags, and crosswalk.	crosswalk
ExistsAttributeGenerator	Tell me the quantity of long kites.	two
AttributeBBoxGenerator	Can you tell what are the attributes for the kite positioned at region (0.13, 0.26, 0.24, 0.47)?	blue
TypedAttributeBBoxGenerator	Provide the shape of the hat found at region (0.89, 0.47, 0.96, 0.51).	round
ExistsRelationGenerator	Can you tell what is the specific relationship between car and bus?	behind and to the left of
RelationBBoxGenerator	What kind of relationship exists between objects at (0.0, 0.76, 0.83, 1.0) and (0.79, 0.77, 0.99, 1.0)?	to the left of
HeadRelationGenerator	Which of glass, kitchen, straw, and wine is to the left of post?	straw
SameObjectSegGenerator	Can you tell the point that is in the same object as (0.83, 0.52) among (0.85, 0.54) and (0.81, 0.54)?	(0.85, 0.54)
DiffObjectSegGenerator	Identify in the list of (0.4, 0.57) and (0.45, 0.54), which point is in the part of different objects as (0.4, 0.46)?	(0.45, 0.54)
CloserPointGenerator	Determine in (0.94, 0.9) and (0.9, 0.23), what is closer point to the camera?	(0.9, 0.23)
FartherPointGenerator	Identify from (0.38, 0.26) and (0.38, 0.83), which point is positioned farther away in depth?	(0.38, 0.26)
CloserObjectGenerator	Identify what is the nearer object in depth, out of plants and olives?	olives
FartherObjectGenerator	Tell me out of wheel and man, which object is located farther to the camera?	man
CloserToAnchorObjectGenerator	In the list of ceiling and shorts, which object is located nearer to the logo in depth?	shorts
FartherToAnchorObjectGenerator	Among nose and steps, which object is positioned farther away to the dirt in depth?	nose
SceneGraphObjectQAGenerator	What is the leafy and small object that the word is to the right of?	tree
SceneGraphRelationQAGenerator	What is the relation from the object, which is behind the empty and wood shelf, to the object, which the large and lying dog is sitting on?	to the left of
SceneGraphAttributeQAGenerator	What is the color of the stone object that the green arrow is to the left of?	brown

Table 6. Example QA for Single-Image Generator.

Task generator	Example question	Example answer
HasRelationMultiGenerator	Determine which image shows that door is to the left of man?	Image 1
HasNotRelationMultiGenerator	Tell me in which image tree isn't to the left of street light?	Image 1
HasObjectMultiGenerator	Which image shows buildings?	Image 0
HasNotObjectMultiGenerator	Tell me which image doesn't have mirror?	Image 0
HasAttributedObjectMultiGenerator	Tell me which image contains black tag?	Image 0
HasNotAttributedObjectMultiGenerator	Which image doesn't show striped mane?	Image 1
HasMostObjectMultiGenerator	Tell me which image shows the most window?	Image 1
HasLeastObjectMultiGenerator	Which image shows the least pole?	Image 0
CommonObjectMultiGenerator	Identify the objects that are seen in all of these images.	pot
CommonAttributeMultiGenerator	Identify what is common attribute of kite across these images?	flying
CountObjectMultiGenerator	Can you tell what is the number of coat in these images?	2
CountAttributeObjectMultiGenerator	Can you tell what is the number of black jacket in these images?	2
CompareRelationMultiGenerator	Determine the difference of the relation between window and windows across these images.	window is to the right of windows in Image 0, to the left of windows in Image 1.
CompareAttributeMultiGenerator	Determine what differences can be observed in the attributes of kite across these images?	kite is blue in Image 0, yellow and flying in Image 1.

Table 7. Example QA for Multi-Image Generator.

	Data Ratio	CVB-2D	CVB-3D	SEED	MMB	MME	QBench2	MMMU	RealWorldQA	Avg.
LLaVA-1.5 instruction data [54]		58.0	61.0	66.8	66.7	63.2	46.4	36.2	54.3	56.6
Replacement	5%	58.0	62.0	66.8	67.8	63.5	46.9	36.3	54.4	57.0
	10%	59.0	66.0	66.9	67.6	61.3	48.1	35.4	55.6	57.5
	20%	58.0	66.0	**67.0**	**68.2**	61.2	47.1	36.0	**56.6**	57.5
	50%	**60.0**	**70.0**	66.2	66.6	**64.0**	**49.9**	**37.4**	55.4	**58.7**
Augmentation	5%	54.0	62.0	66.7	66.8	63.4	46.0	36.9	55.7	56.4
	10%	58.0	66.0	**67.4**	66.8	**64.5**	45.7	35.6	56.3	57.5
	20%	57.0	68.0	66.7	66.1	62.9	**49.8**	37.3	55.0	57.9
	50%	**60.0**	**69.0**	67.2	**67.2**	64.3	48.6	**37.9**	**58.2**	**59.0**

Table 8. Raw results of Figure 4, Results of instruction tuning LLaVA-1.5-7B with **DC-S**, *i.e.*, single-image instruction data generated from DataComp images with our automatic scene graph generation pipeline.

		Multi-image benchmark		Single-image benchmark						
	Data Ratio	**Mantis-Eval**	**MMT**	**SEED**	**MMB**	**MME**	**QBench2**	**MMMU**	**RealWorldQA**	**Avg.**
Mantis instruction data [30]		54.4	52.9	68.1	72.8	58.5	70.1	44.3	51.5	59.1
Replacement	5%	52.5	**52.3**	**68.7**	73.1	58.7	**70.2**	**43.7**	51.2	58.8
	10%	**60.8**	52.0	68.0	73.5	58.5	69.9	42.3	51.4	**59.6**
	20%	55.8	51.8	**68.7**	**73.9**	**59.9**	68.0	42.9	52.0	59.1
	50%	54.4	51.8	67.2	71.1	59.5	65.9	41.4	**52.6**	58.0
Augmentation	5%	59.4	52.5	68.4	**73.2**	58.1	70.1	43.3	49.4	59.3
	10%	**59.9**	**52.7**	68.6	73.1	57.8	70.4	42.2	51.6	59.5
	20%	58.5	52.5	**68.8**	72.4	**60.5**	**70.5**	43.0	**52.8**	**59.9**
	50%	59.0	**52.7**	68.6	73.1	59.0	69.9	**43.7**	51.4	59.7

Table 9. Raw results of Figure 5, Results of instruction tuning Mantis-SigLIP-8B with **DC-M**, *i.e.*, multi-image instruction data generated from DataComp images with our automatic scene graph generation pipeline.