Pink: Unveiling the Power of Referential Comprehension for Multi-modal LLMs

Shiyu Xuan[1]*, Qingpei Guo[2], Ming Yang[2], Shiliang Zhang[1]
[1]National Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University, Beijing, China.
[2]Ant Group
shiyu_xuan@stu.pku.edu.cn, {qingpei.gqp, m.yang}@antgroup.com, slzhang.jdl@pku.edu.cn

Abstract

Multi-modal Large Language Models (MLLMs) have shown remarkable capabilities in various multi-modal tasks. Nevertheless, their performance in fine-grained image understanding tasks is still limited. To address this issue, this paper proposes a new framework to enhance the fine-grained image understanding abilities of MLLMs. Specifically, we present a new method for constructing the instruction tuning dataset at a low cost by leveraging annotations in existing datasets. A self-consistent bootstrapping method is also introduced to extend existing dense object annotations into high-quality referring-expression-bounding-box pairs. These methods enable the generation of high-quality instruction data which includes a wide range of fundamental abilities essential for fine-grained image perception. Moreover, we argue that the visual encoder should be tuned during instruction tuning to mitigate the gap between full image perception and fine-grained image perception. Experimental results demonstrate the superior performance of our method. For instance, our model exhibits a 5.2% accuracy improvement over Qwen-VL on GQA and surpasses the accuracy of Kosmos-2 by 24.7% on RefCOCO_val. We have also attained the top rank on the leaderboard of MMBench. This promising performance is achieved by training on only publicly available data, making it easily reproducible. The models, datasets, and codes are publicly available at https://github.com/SY-Xuan/Pink.

1. Introduction

Large Language Models (LLMs) [2, 29, 30, 34] show impressive capabilities across a wide range of natural language tasks. These inspiring results have motivated researchers to extend LLMs to Multi-modal Large Language Models (MLLMs) by integrating additional modalities, *e.g.*, image, audio, or point cloud. Visual instruction tuning [6, 19, 43], using high-quality image-text instruction tuning data, allows the incorporation of visual comprehension ability into LLMs by projecting visual features into the natural language space of the LLMs [44]. Powered by those methods, existing MLLMs are capable of basic image-level comprehension. However, they are still confronted by fine-grained image understanding [13, 36, 37]. Limited fine-grained image understanding ability hinders the performance of MLLMs in multi-modal tasks and restricts their potential applications as reported in the GPT-4V(ision) test report [42].

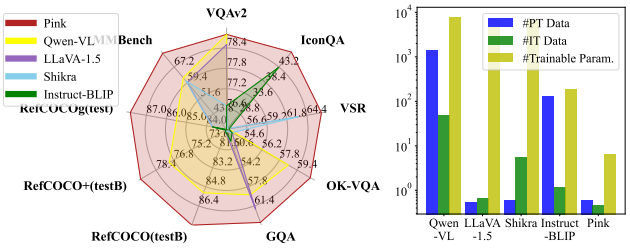

Figure 1. With fewer trainable parameters and less training data, Pink achieves the best performance on both conventional multi-modal tasks and RC tasks. "#Trainable Param.", "#PT Data", and "#IT Data" indicate the number of trainable parameters, the number of samples in pre-training and instruction tuning stage, respectively.

To address this issue, some methods [4, 26] incorporate some datasets related to Referential Comprehension (RC) such as RefCOCO [13], and PointQA [23] to enhance the fine-grained image perception ability of MLLMs. However, these datasets are insufficient to cover a wide range of abilities that MLLMs desire to have for fine-grained image perception. Limited RC tasks also make it hard for the model to generalize across various RC tasks through instruction tuning. For instance, as shown in Fig. 4, Shikra [4] and Qwen-VL [1] show limited instruction-following ability on RC tasks beyond its instruction tuning, failing to provide

*This work was done during the internship of the first author at Ant Group.

relevant responses to questions.

In addition to instruction tuning, the capability of the visual encoder is also important to the performance of MLLMs. MLLMs typically employ a visual encoder trained through contrastive language-image pre-training like the one in CLIP [28]. Simply performing global alignment is ineffective in exploring fine-grained relationships between image regions and text words [40, 47]. The visual encoder stands as a bottleneck for achieving fine-grained image perception in MLLMs.

This paper proposes a new framework to enhance the fine-grained image perception ability of MLLMs through RC tasks. We refer to the trained model as **Pink** [1]. Fine-grained image understanding is closely tied to some fundamental abilities such as instance identification and recognition of relative positions between different instances. Integrating tasks that demand these fundamental abilities during instruction tuning is crucial for enhancing the model's fine-grained image perception ability. To this end, we propose a new dataset construction pipeline that extends the annotations of existing datasets to various RC tasks about these fundamental abilities. Specifically, we design several RC tasks, such as visual relation reasoning and visual spatial reasoning, based on the annotations from Visual Genome [14]. To further incorporate more training data for these RC tasks, we introduce a novel self-consistent bootstrapping method to extend dense object annotations to referring-expression-bounding-box pairs. Compared to the expensive, and uncontrollable process of generating data using the GPT4 API [4, 46], our method leverages the existing annotations from datasets. This approach results in high-quality data and precise enhancement of the necessary capabilities required by the model. As shown in Fig. 1, the high-quality instruction tuning data generated by our method enables the model to achieve promising performance with a reduced number of training samples.

Improving the fine-grained image understanding ability of the visual encoder is not a trivial task. Most existing MLLMs [4, 6, 19, 43, 48] freeze the visual encoder during instruction tuning. Because directly tuning the visual encoder can result in a semantic loss due to the limited scale of the visual instruction dataset [38]. To address this issue, we tune the visual encoder by introducing several tunable components like Adapters [10] and LoRA [11]. Freezing the main parameters of the model avoids forgetting the learned knowledge. The introduced tunable components are trained to adapt the visual encoder.

We have conducted extensive experiments to test the performance of the model. Benefited by the designed tasks in the instruction tuning, our framework enhances MLLMs' performance in both conventional vision-language tasks and RC tasks. For instance, with only 6.7M tunable parameters, we achieve up to a 6.0% accuracy improvement on OK-VQA [25] compared to Shikra [4]. We also attain the top rank on the leaderboard of MMBench [20]. It should be noted that, our method also surpasses methods that rely on more training data, *e.g.*, Qwen-VL [1].

This work is an original effort to enhance MLLMs' fine-grained image perception ability by addressing two main bottlenecks: limited instruction tuning tasks and the lack of ability of the visual encoder. By designing tasks related to fundamental abilities, every dataset with corresponding annotations can be converted into the instruction tuning dataset. The self-consistent bootstrapping further increases the number of training data. This dataset construction pipeline significantly reduces the cost of obtaining high-quality data with diversified tasks and eliminates the dependency on GPT4 APIs. The whole training pipeline is reproducible in academia as it only relies on publicly available data and can be trained on consumer GPUs with 24GB memory. We will release the codes and datasets to facilitate further research and evaluation.

2. Related Works

Multi-modal large language model. Several approaches have been proposed to condition LLMs with additional modalities. Typically, these methods utilize two-stage training. The pre-training stage is performed to align two modalities with image-text pairs. The subsequent stage is adopted to improve the ability of MLLMs to follow instructions with high-quality instruction tuning dataset [17, 19, 41, 48]. Many methods freeze the visual encoder during the pre-training stage to reduce requirement of large-scale image-text pairs. For example, Mini-GPT4 [48] and LLaVA [19] only fine-tune a single fully connected layer to align the vision and language modalities. Other methods leverage millions of image-text pairs to achieve better alignment between two modalities. Instruct-BLIP [6] introduces an instruction-aware visual feature extraction method and fine-tunes the entire Q-Former, showing promising zero-shot generalization ability on various multi-modal tasks. mPlug-Owl [43] incorporates a visual abstractor module to align the two-modalities. Both the visual encoder and the visual abstractor are updated during the pre-training stage. All of above methods freeze the visual encoder during the multi-modal instruction tuning stage to prevent the potential semantic loss caused by the small-scale instruction tuning dataset. However, this strategy makes the visual encoder cannot benefit from the multi-modal instruction tuning.

Referential Comprehension of MLLMs. Referential comprehension is important to the fine-grained image perception of MLLMs. Therefore, enhancing MLLMs with the RC ability is highly valuable. Inspired by Pix2Seq [5], many works use discrete coordinate tokens to encode spatial in-

[1]This name is from the main character of the album *The Wall* by the great rock band *Pink Floyd*.

formation and unify RC tasks as sequence generation tasks, *e.g.*, OFA [39], Unified-io [21], and Kosmos-2 [26]. Another line of works, as seen in PVIT [3] and GPT4RoI [45], leverage the ROI operation [9] to extract features of referring objects. These works require extra modules and may lose context information. Another limitation of these works is that they cannot refer objects in their responses, limiting their applications, *e.g.*, in visual grounding.

In addition to the model design, the construction of RC instruction tuning data also plays a crucial role. Shikra [4] converts existing datasets of RC tasks including RefCOCO [13] and PointQA [23] into the instruction following format. Kosmos-2 uses the grounding model GLIP [15] to extract coordinates of noun chunks in image captions and constructs a large-scale dataset. Datasets constructed by the above methods only includes RC tasks, such as visual grounding, grounding caption and pointQA, which are still not diversified enough to cover various RC tasks. The trained models thus show a poor ability to generalize to new RC tasks beyond the instruction tuning. ChatSpot [46], PVIT [3], and Shikra [4] all prompt GPT4 to generate instruction tuning data for RC, which is expensive and uncontrollable.

Differences with previous works. Existing methods for enhancing MLLMs through RC tasks construct the instruction tuning dataset by either integrating existing RC datasets or relying on GPT4 APIs. However, these methods exhibit some major drawbacks: 1) the diversity of RC tasks cannot cover a wide range of fundamental abilities, and 2) data generation through GPT4 APIs is expensive, uncontrollable, and prone to noise. In contrast, our work effectively leverages existing datasets to cover a wide variety of RC tasks. The proposed self-consistent bootstrapping method extends dense object annotations to referring-expression-bounding-box pairs. This pipeline significantly reduces the cost of generating high-quality datasets. The high qualify of the data allows our model to be trained with fewer parameters on less training data, which is friendly to reproduce in academia, than large commercial MLLMs.

3. Methodology

3.1. Model Architecture and Training Pipeline

Model architecture. As shown in Fig. 2, Pink follows a similar architecture of LLaVA [19], which consists of a visual encoder Φ_V, a projection layer Φ_P, and a decoder-only LLM Φ_L. Given an image I and a sequence of word embeddings Q_T representing an instruction sentence, the visual encoder is employed to embed the image as a sequence of visual tokens $Z_V = \Phi_V(I)$. A linear layer is used as Φ_P to convert Z_V into the input space of the LLM $Z_T = \Phi_P(Z_V)$. Z_T and Q_T are concatenated and fed into Φ_L to generate the next word.

To enable the LLMs to take coordinates as input and output, similar to Shikra [4], the coordinates are converted into texts in a specific format. Specifically, for a bounding box represented by its coordinates of the top-left and bottom-right corners $[x_{min}, y_{min}, x_{max}, y_{max}]$, the coordinates are normalized to the range $[0, 1]$ with respect to the image size and retain 3 decimal places for each number, *e.g.*, [0.222,0.333,0.444,0.555]. This design allows the coordinates to be processed as regular text and can appear in both the input and output of the model.

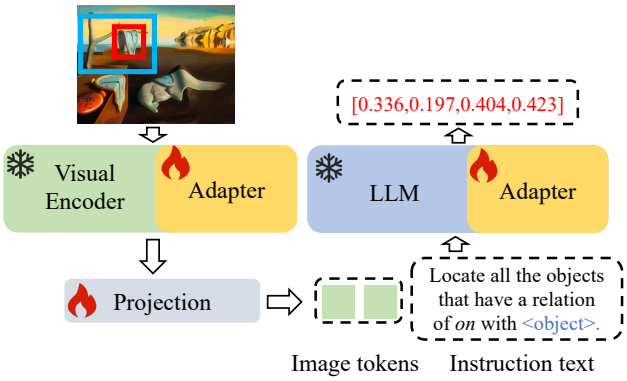

Figure 2. The illustration of our Pink model. Pink follows the architecture of LLaVA [19], which consists of three main components: a visual encoder, a projection layer, and a decoder-only LLM. The coordinates of a bounding box are converted into texts in a specific format. During instruction tuning, we freeze the visual encoder and LLM and only update the Adapters and the projection layer.

The visual encoder pre-trained by the contrastive loss [28] lacks region-level image comprehension. Directly fine-tuning the entire visual encoder during instruction tuning could lead a semantic loss due to the limited instruction tuning data [38]. To incorporate the fine-grained image perception ability of the visual encoder through the multi-modal instruction tuning with RC tasks, we freeze the visual encoder, meanwhile introducing tunable modules into it. This approach prevents the visual encoder from suffering semantic loss and provides an efficient way to adapt the model. In particular, we employ the Adapter [10] at both the visual encoder and LLM. Given an input token feature $Z \in \mathbb{R}^d$, the architecture of an Adapter is defined as follows,

$$\hat{Z} = \sigma\left(ZW_d\right)W_u + Z, \tag{1}$$

where $W_d \in \mathbb{R}^{d\times d_s}$ and $W_u \in \mathbb{R}^{d_s\times d}$ denote the weight matrices, d_s is the hidden dimension which is much smaller than d, and σ denotes the non-linear activation function. W_u is initialized to zero to ensure that at the beginning of the training, the Adapter does not change the original output.

Training pipeline. Both the image and coordinates are mapped into the input space of the LLM. Consequently, the model can be trained end-to-end using a language model-

ing task, which predicts the next word token based on the preceding context.

The model is trained in two stages. In the first stage, we exclusively fine-tune the projection layer with a small set of image-text pairs. In the second stage, we freeze both the visual encoder and LLM and fine-tune the newly added Adapters and the projection layer with the instruction tuning dataset. Therefore, both the visual and language modalities can benefit from the multi-modal instruction tuning.

3.2. Instruction tuning Dataset Construction

To create the instruction tuning dataset, we unify all the multi-modal tasks into a vision-language dialogue format:

> **Image:** {Image tokens}
> **User:** {Instruction template}
> **Assistant:** {Response}

where the placeholders {Image tokens}, {Instruction template}, and {Response} will be replaced with the image tokens extracted by Φ_V, task instruction template, and the response, respectively.

It is important to introduce diverse RC tasks for the instruction tuning to cover a wide range of fundamental abilities for the find-grained image perception of MLLMs. Existing datasets only offer limited RC tasks, *e.g.*, visual grounding, grounding caption [13], and pointQA [23]. Besides the RC tasks mentioned above, we design more diversified RC tasks by incorporating annotations from Visual Genome [14], which contain information about region descriptions, objects, and relations between different objects. Following parts proceed to introduce those tasks.

Visual relation reasoning. Visual Genome has annotated millions of relationship triplets (*subject-predicate-object*), *e.g.*, man-wearing-hat. We design two types of visual relation reasoning tasks by leveraging these annotations to help the model understand visual relationships between different objects: (1) We randomly select a relationship triplet. Given the coordinates of *subject* and *object*, the model is required to predict their relation. (2) We randomly select one *subject* and a relation from the annotations. The model is required to detect all objects that have selected relation with *subject* and output their coordinates and class names.

Coarse visual spatial reasoning. We introduce a coarse visual spatial reasoning task by utilizing the object annotations from Visual Genome. This task enhances MLLMs to identify relative spatial relation between different instances. We define four coarse spatial positions as *top-left, top-right, bottom-left, and bottom-right*. Given a randomly selected object and a coarse spatial position, the model is required to identify all objects located at this position relative to the selected object and predict their coordinates and class names.

Object counting. To endow the model with the concept of different instances and the capability of fine-grained object recognition, we design an object counting task. This task requires the model to count objects in the image that belong to the same category as the given object or class name.

Object detection. Object detection can empower the model to locate the position and boundaries of objects. This task is also important for the model to identify the existence or category of a certain object. Given a class name or a selected object, the model is asked to identify all objects that belong to the same category of the given object or class name, and provide their coordinates.

By incorporating these RC tasks into the instruction tuning, the model can learn a variety of RC abilities. To clarify the designed tasks, we list some instruction templates as follows,

> **Visual Relation Reasoning:**
> **User:** Assist me in finding the relation between <subject> and <object> in the photo.
> **Assistant:** <relation>.
>
> **User:** Please locate and categorize all the objects that have a relation of <relation> with <subject>.
> **Assistant:** <object> <category> <object> <category>.
>
> **Coarse Visual Spatial Reasoning:**
> **User:** Identify the objects located at <loc> of <object>.
> **Assistant:** <object> <category> <object> <category>.
>
> **Object Counting:**
> **User:** How many objects in the image are of the same category as <object>.
> **Assistant:** <number>.
>
> **Object Detection:**
> **User:** Identify all the objects that fit the same category as <object> and display their coordinates.
> **Assistant:** <object> <object>.

where the placeholders <object> and <category> will be replaced with the bounding box coordinates and the class name of a referring object, respectively. <subject> will be replaced with the bounding box coordinates of the selected subject. <relation>, <loc>, and <number> will be replaced with the relation between different objects, the selected relative spatial position, and the number of the objects, respectively. All instruction templates can be found in Supplementary Material.

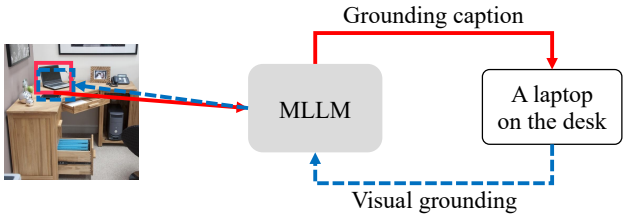

Figure 3. The illustration of self-consistent bootstrapping method. Given a bounding box, our method first generates its description by asking MLLM to perform grounding caption, then leverages the visual grounding to locate the generated description. The low-quality description will be filtered if the IOU between the predicted and ground-truth bounding box is below a threshold.

3.3. Self-consistent Bootstrapping Method

The constructed instruction-following datasets are adopted to reinforce the fine-grained image perception ability of MLLMs. We further acquire more high quality data. Existing datasets for object detection provide valuable bounding box annotations for objects appearing in the image, making them promising resources for instruction tuning. We propose a self-consistent bootstrapping method by leveraging those datasets. This method extends bounding box annotations to the referring-expression-bounding-box pairs. It comprises two key stages: bounding box description bootstrapping and self-consistent filtering as shown in Fig. 3.

At the bounding box description bootstrapping stage, given a bounding box B of an object, we prompt the model to generate a description D_B for that object by leveraging its ability of grounding caption. Due to the complexity of scenes or the presence of duplicate objects, the generated description may be noisy or fails to uniquely describe the corresponding object. Then, the self-consistent filtering stage is adopted to filter those low-quality descriptions. Specifically, with the generated description D_B, we locate this description in the image and predict the bounding box $\hat{B}$ by leveraging the visual grounding ability of our model. The generated description will be removed if the Intersection Over Union (IOU) between B and $\hat{B}$ is below a pre-defined threshold λ. This stage ensures that only high-quality descriptions are retained.

These two stages are performed to extend every annotated object in the dataset with textual description. This extended dataset is then well-suited for a wide range of RC tasks, *e.g.*, coarse visual spatial reasoning, object detection, object counting, visual grounding and grounding caption. Illustrations of generated data are in Supplementary Material. This self-consistent bootstrapping method serves as a powerful tool to harness the potential of object detection datasets for enhancing the RC ability.

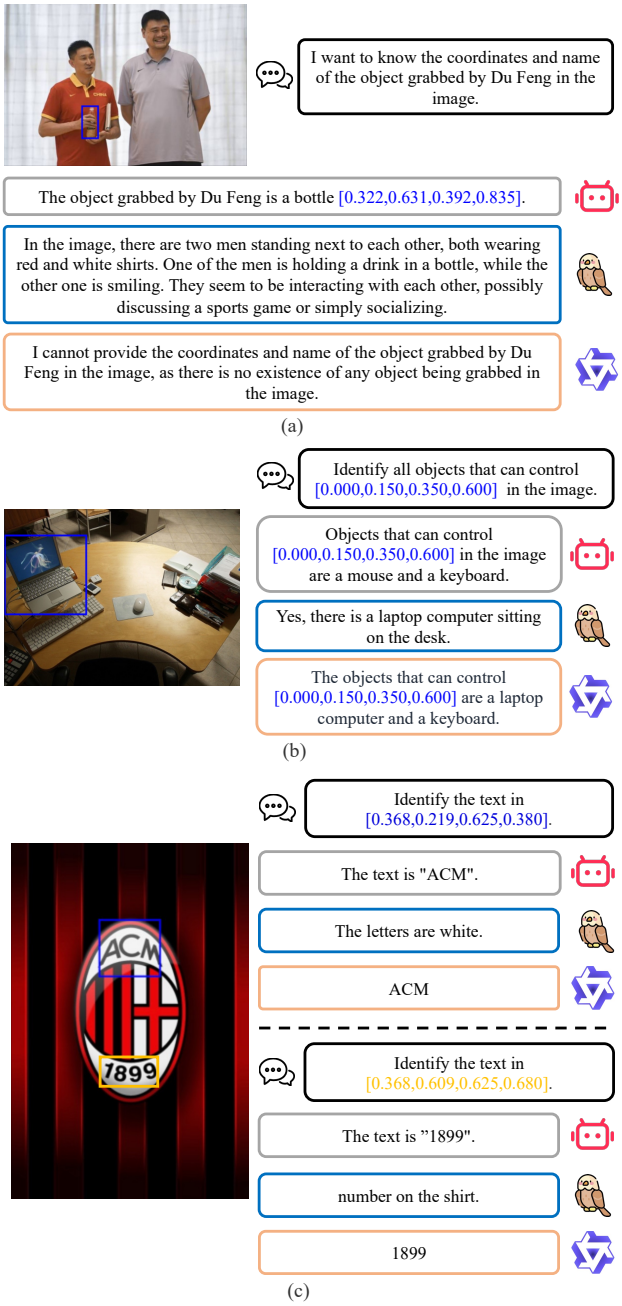

Figure 4. A comparison of Pink against the recent MLLMs Shikra and Qwen-VL on RC reasoning.

3.4. Qualitative Results on RC Reasoning

In Fig. 4, we compare some qualitative results on RC reasoning with Shikra [4] and Qwen-VL [1]. Pink exhibits substantially better capability on these RC tasks.

To answer the question as shown in Fig. 4 (a), the model needs to first identify Du Feng, a famous Chinese basket-

ball player, then understand the action of grabbing. Pink successfully provides the correct answer. Pink also exhibit better reasoning capability as shown in Fig. 4 (b). It successfully identifies the referred region as laptop, and inferred that, the mouse and keyboard in the image could control the labtop. Shikra fails to follow the instruction and provides an un-related answer. Similarly, Qwen-VL only outputs the category of the referred object.

As shown in Fig. 4 (c), trained with millions of OCR-based data in the instruction tuning, Qwen-VL can give correct responses to OCR instruction. It is interesting to observe that, without any OCR-based data, our model also accurately recognizes characters located at referred region. It indicates that, Pink exhibits promising generalization capability to different RC tasks. More qualitative results can be found in Supplementary Material. Extensive experiments are conducted in following section.

4. Experiments

4.1. Experimental Setting

Model architecture. We employ the ViT-L/14 [7] as the visual encoder, which is pre-trained with CLIP [28]. We choose an instruction-tuned model Vicuna-7B [35] based on LLaMA-1 [34] as the LLM. The projection layer is a single fully connected layer. The Adapters are inserted before each self-attention layer of both the visual encoder and the LLM, with a hidden dimension $d_s = 8$. The tunable parameter numbers of Adapter in the visual encoder and LLM are 393,216 and 2,097,152, respectively. The number of parameters in the projection layer is 4,194,304. Therefore, the total number of tunable parameters is about 6.7M.

Training data. The first stage utilizes 595K image-text pairs from CC3M [33], the same as LLaVA [19]. The second stage adopts VQAv2 [8], LLaVA-150K [19], A-OKVQA [31], Flickr30K [27], Visual Genome [14] and Object365 [32] with referring-expression-bounding-box pairs generated by our self-consistent bootstrapping method. At each training iteration, when using an image in Visual Genome or Object365, one designed RC task will be selected randomly. The model used to generate referring-expression-bounding-box pairs in Object365 is trained with the aforementioned datasets, excluding Object365 itself. Note that we reduce the probability of sampling Object365 in batch construction to avoid a large number of training samples in Object365 dominating the training.

Training details. AdamW is adopted as the optimizer. In the first stage, the model is trained for 1 epoch with a batch size of 128 and weight decay of 0.0. After a warm-up period of 200 steps, the learning rate starts at 0.03 and decays to 0 with the cosine schedule. In the second stage, the model is trained for 6 epochs with a batch size of 32 and weight decay of 0.05. The warm-up phase consists of 10k steps and the learning rate starts at 5e-4. The input image is resized to 224×224 without any additional data-augmentation. We set λ as 0.5 to filter out low-quality descriptions. The model is trained using 8 NVIDIA A100 GPUs. It takes about 1 and 30 hours for the first and second stage, respectively.

Evaluation settings. We evaluate our model on various datasets under the zero-shot and fine-tuning settings to validate the instruction-following ability of the trained model. These datasets encompass conventional multi-modal reasoning tasks, including conventional VQA (VQAv2 [8], MM-Bench [20]), abstract diagram understanding (IconQA [22]), visual spatial reasoning (VSR [16]), knowledge-intensive VQA (OK-VQA [25]), scene understanding (GQA [12]), and RC tasks such as RefCOCO/+ [13], RefCOCOg [24], Visual-7W [49], PointQA-Local/LookTwice [23].

4.2. Ablation Study

Instruction tuning dataset construction. To investigate the impact of instruction tuning dataset construction, we conduct ablation studies by excluding Visual Genome, designed RC tasks, and Object365 with referring-expression-bounding-box pairs. Results are summarized in Table 1.

As shown in Table 1, enhancing the MLLM with RC task can benefit the conventional multi-modal reasoning tasks, *e.g.*, removing visual grounding and grounding caption tasks leads to a performance degradation of 2.8% on VSR. When the model is trained solely with visual grounding and grounding caption, it fails to provide correct responses to the questions in PointQA-Local/LookTwice, indicating limited instruction-following ability for RC tasks. As more RC tasks are included, the model begins to exhibit better instruction-following ability for these tasks. The performance on PointQA-Local/LookTwice increases from 0.0%/0.2% to 54.6%/63.1%. The combination of all designed RC tasks yields the best performance on both conventional multi-modal reasoning tasks and RC tasks, thus validating the effectiveness of our instruction tuning dataset construction method. We also observe that incorporating Object365 further enhances the performance of our method on RC tasks. For example, on RefCOCO_val, the zero-shot accuracy increases from 54.1% to 77.0%. Notably, the exclusion of the self-consistent method results in the degradation of performance from 77.0% to 73.8% on RefCOCO_val due to low-quality referring-expression-bounding-box pairs generated by the model. These results demonstrate the importance of the generated data and underscore the importance of the proposed self-consistent method.

Training settings of visual encoder. We further assess the effectiveness of different settings for training the visual encoder, and summarize results in Table 1.

Full-tuning the visual encoder results in a significant performance degradation. The performance on RefCOCO_val drops from 54.1% to 0.05%. This result aligns with the

Settings	IconQA	VSR	OK-VQA	RefCOCO_val	Local	LookTwice
Baseline	44.6	65.6	58.5	55.0	0.0	0.2
w/o VG	43.1	62.8	58.3	-	-	-
+ R	44.4	65.7	58.5	52.1	17.1	12.8
+ R,S	46.2	65.8	58.5	52.7	50.9	60.0
+ R,S,C	47.4	65.7	58.9	53.1	53.4	60.7
+ R,S,C,D	**47.8**	66.3	59.5	54.1	54.6	63.1
+ R,S,C,D + Object365†	44.6	65.9	58.7	73.8	52.1	69.2
+ R,S,C,D + Object365	47.7	**67.1**	**59.5**	**77.0**	**57.2**	**70.3**
Freezing	42.9	61.5	58.3	37.2	44.9	57.5
Full-tuning	36.9	48.6	33.1	0.05	26.1	54.1
LoRA	44.3	65.4	58.9	**54.7**	**56.7**	62.2
Our	**47.8**	**66.3**	**59.5**	54.1	54.6	**63.1**

Table 1. Ablation study on instruction tuning dataset construction and training settings of visual encoder under a zero-shot setting. "Baseline" denotes leveraging Visual Genome by only performing visual grounding and grounding caption tasks. "VG" denotes Visual Genome. "R", "S", "C", and "D" denote the visual relation reasoning, coarse visual spatial reasoning, object counting and object detection tasks, respectively. † denotes generated referring-expression-bounding-box pairs in Object365 are not filtered with the self-consistent method. "Freezing" and "Full-tuning" denotes freezing the visual encoder and training the entire visual encoder, respectively. "LoRA" denotes using LoRA instead of the Adapter to perform parameter-efficient tuning.

conclusion in [38] that fine-tuning the visual encoder using a small-scale instruction tuning dataset can lead to a subsequent drop in performance. Freezing the visual encoder also leads to performance degradation on various datasets. It can be attributed to the limited ability of the visual encoder to fine-grained image understanding. Our design allows for the optimization of both modalities and leverages the benefits of multi-modal instruction tuning, resulting in improved performance. Moreover, using LoRA [11] instead of the Adapter to perform parameter-efficient tuning can also achieve improved performance compared with Full-tuning or Freezing, demonstrating the effectiveness of adapting the visual encoder during multi-modal instruction tuning.

4.3. Comparison with Recent Works

This section proceeds to validate the effectiveness of our method through comparison with other recent works.

Evaluation on the conventional multi-modal reasoning tasks. To evaluate the instruction-following ability of our method, we conduct experiments of five multi-modal reasoning tasks on public benchmarks. These benchmarks and tasks assess various aspects of multi-modal comprehension ability of the model. As shown in Table 2a, our model consistently achieves the best performance using fewer trainable parameters, a smaller training set, and lower input image resolution. This demonstrates that the constructed instruction tuning dataset can effectively enhance the fine-grained perception capability.

Zero-shot evaluation on the visual grounding task. Visual grounding is a fundamental RC task that requires the model to predict the coordinates of a bounding box based on a given textual description. We evaluate our model on three well-established datasets under the zero-shot setting in Table 2b. Our model significantly outperforms Kosmos-2 [26], which is trained with a generated dataset GRIT containing 91M images and 115M referring expressions. This result validates the effectiveness of the proposed self-consistent bootstrapping method.

Comparison with other models under fine-tuning setting on RC tasks. To further validate the RC ability of our method, we compare it with other models [1, 4, 39] that can perform RC tasks. The comparison incorporates models that have the capability to handle various vision-language tasks. Models specifically designed for the visual grounding task are not included. The instruction tuning dataset of compared models includes the training sets of datasets listed in Table 2c. For a fair comparison, we also leverage those training sets in the instruction tuning of Pink. Note that, the training set size of Pink is still substantially smaller than those in compared methods, *e.g.*, 50M of Qwen-VL [1] *vs.* 519K of Pink.

The results in Table 2c show that our model obtains promising performance under the fine-tuning setting. This can be attributed to the diversity of RC tasks in the instruction tuning. Object365 further improves the performance, even training sets of these datasets are already included. This demonstrates the effectiveness of self-consistent bootstrapping method in converting existing dataset into more valuable RC training set. Qwen-VL utilizes a stronger visual encoder ViT-G trained by CLIP. To make a fair comparison, we change the visual encoder of Pink to ViT-G. Pink-G outperforms Qwen-VL by large margins, even with a lower input image resolution and less training samples. This result further validates the effectiveness of our training pipeline.

Comparison with other models on MMBench test set.

Models	Res.	#PT Data	#IT Data	#Trainable Param.	VQAv2	IconQA	VSR	OK-VQA	GQA
Instruct-BLIP [6]	224	129M	1.2M	188M	-	43.1	54.3	-	49.2
Shikra-7B [4]	224	595K	5.5M	7B	76.7†	24.3	63.3	53.5	47.4
Pink	224	595K	**396K**	**6.7M**	**78.7**†	**47.8**	**66.3**	**59.5**	**52.6**
Qwen-VL [1]	448	1.4B	50M	8B	78.8†	-	-	58.6†	59.3†
LLaVA-1.5 [18]	336	**558K**	665K	7B	78.5†	-	-	-	62.0†
Pink+	224	595K	**477K**	**6.7M**	**78.8**†	48.8	67.4	**60.6**†	**64.5**†

(a) Results on the conventional multi-modal reasoning tasks. † denotes the training set of corresponding dataset is included. + denotes adding the training set of OK-VQA and GQA during instruction tuning. "Res.", "#Trainable Param.", "#PT Data", and "#IT Data" indicate input image resolution, the number of trainable parameters, the number of samples in pre-training and instruction tuning stage, respectively.

Models	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
Kosmos-2 [26]	52.3	57.4	47.3	45.5	50.7	42.2	60.6	61.7
Pink	54.1	61.2	44.2	43.9	50.7	35.0	59.1	60.1
Pink*	**77.0**	**82.4**	**68.2**	**65.6**	**75.2**	**53.4**	**72.4**	**74.0**

(b) Zero-shot results on the visual grounding task.

Models	Visual Encoder	Res.	RefCOCO			RefCOCO+			RefCOCOg		Visual-7W	LookTwice
			val	testA	testB	val	testA	testB	val	test		
OFA-L [39]	ResNet152	480	80.0	83.7	76.4	68.3	76.0	61.8	67.6	67.6	-	-
Shikra-7B [4]	ViT-L	224	87.0	90.6	80.2	81.6	87.4	72.1	82.3	82.2	84.3	72.1
Pink	ViT-L	224	88.3	91.7	84.0	81.4	87.5	73.7	83.7	83.7	85.1	73.5
Pink*	ViT-L	224	**88.7**	**92.1**	**84.0**	**81.8**	**88.2**	**73.9**	**83.9**	**84.3**	**85.3**	**73.6**
Qwen-VL [1]	ViT-G	448	89.4	92.3	85.3	83.1	88.3	77.2	85.6	85.5	-	-
Pink-G	ViT-G	224	**91.5**	**93.4**	**88.0**	**86.0**	**89.5**	**79.8**	**86.8**	**87.8**	**86.8**	**76.6**

(c) Fine-tuning results on the RC tasks. Pink-G indicates the ViT-G is used as the visual encoder for a fair comparison.

Models	Overall	LR	AR	RR	FP-S	FP-C	CP
Kosmos-2 [26]	58.2	48.6	59.9	34.7	65.6	47.9	70.4
LLaVA-1.5 [18]	59.5	32.4	72.6	49.3	62.3	52.2	67.7
Qwen-VL [1]	61.8	40.5	74.3	47.9	66.3	46.2	72.8
mPlug-Owl [43]	68.5	56.8	77.9	62.0	72.0	58.4	72.6
Pink	**74.1**	**58.5**	**78.2**	**73.2**	**77.3**	**67.2**	**78.7**

(d) CircularEval results on MMBench test set [20].

Table 2. Comparison with other methods. * denotes Object365 with generated referring-expression-bounding-box pairs is used.

MMBench [20] has been proposed as a new benchmark to evaluate various abilities of MLLMs, including logical reasoning (LR), attribute reasoning (AR), relation reasoning (RR), fine-grained perception single instance (FP-S), fine-grained perception cross instance (FP-C), and coarse perception (CP). We hence conduct experiments on MMBench to validate the capabilities of our model in all aspects.

The results summarized in Table 2d show that our method achieves the best overall performance among compared methods. The main improvement comes from RR, FP-S, and FP-C. For example, compared with mPlug-Owl [43], the accuracy boosting of Pink on LR and FP-C is 1.7% and 8.8%, respectively. These results demonstrate the strong fine-grained perception ability of Pink, which can be attributed to the incorporation of various RC tasks during the multi-modal instruction tuning.

5. Conclusion

This paper presents a novel framework for enhancing fine-grained image perception ability of MLLMs. The framework includes a method for constructing an instruction tuning dataset by converting annotations from existing datasets into diverse RC tasks. A self-consistent bootstrapping method is proposed to extend object annotations to referring-expression-bounding-box pairs, enabling the acquisition of more instruction tuning data at a low cost. The visual encoder is tuned in a parameter-efficient way to gain fine-grained image understanding ability. With fewer trainable parameters and less training data, our method achieves superior performance on both multi-modal tasks and RC tasks.

Acknowledgement This work is supported in part by Natural Science Foundation of China under Grant No. U20B2052, 61936011, in part by the Okawa Foundation Research Award, and in part by Ant Group Research Intern Program.

References

[1] 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. 1, 2, 5, 7, 8

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 1

[3] Chi Chen, Ruoyu Qin, Fuwen Luo, Xiaoyue Mi, Peng Li, Maosong Sun, and Yang Liu. Position-enhanced visual instruction tuning for multimodal large language models, 2023. 3

[4] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 1, 2, 3, 5, 7, 8

[5] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021. 2

[6] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 1, 2, 8

[7] 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. In *ICLR*. OpenReview.net, 2021. 6

[8] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 6

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[10] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2, 3

[11] 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 *ICLR*. OpenReview.net, 2022. 2, 7

[12] 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. 6

[13] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 1, 3, 4, 6

[14] 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. 2, 4, 6

[15] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 3

[16] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023. 6

[17] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023. 2

[18] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 8

[19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *CoRR*, abs/2304.08485, 2023. 1, 2, 3, 6

[20] 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. 2, 6, 8

[21] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 3

[22] 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. *arXiv preprint arXiv:2110.13214*, 2021. 6

[23] Arjun Mani, Nobline Yoo, Will Hinthorn, and Olga Russakovsky. Point and ask: Incorporating pointing into visual question answering. *arXiv preprint arXiv:2011.13681*, 2020. 1, 3, 4, 6

[24] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 6

[25] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering

benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019. 2, 6

[26] 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. *arXiv preprint arXiv:2306.14824*, 2023. 1, 3, 7, 8

[27] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 6

[28] 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. 2, 3, 6

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67, 2020. 1

[30] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022. 1

[31] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022. 6

[32] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 6

[33] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL (1)*, pages 2556–2565. Association for Computational Linguistics, 2018. 6

[34] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. 1, 6

[35] Vicuna. Vicuna: An open chatbot impressing gpt-4. https://github.com/lm-sys/FastChat, 2023. 6

[36] Dongkai Wang and Shiliang Zhang. Contextual instance decoupling for instance-level human analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[37] Dongkai Wang, Shiliang Zhang, Yaowei Wang, Yonghong Tian, Tiejun Huang, and Wen Gao. Humvis: Human-centric visual analysis system. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9396–9398, 2023. 1

[38] Guangzhi Wang, Yixiao Ge, Xiaohan Ding, Mohan Kankanhalli, and Ying Shan. What makes for good visual tokenizers for large language models?, 2023. 2, 3, 7

[39] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 3, 7, 8

[40] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. Clipself: Vision transformer distills itself for open-vocabulary dense prediction. *arXiv preprint arXiv:2310.01403*, 2023. 2

[41] Canwen Xu, Daya Guo, Nan Duan, and Julian J. McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *CoRR*, abs/2304.01196, 2023. 2

[42] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9, 2023. 1

[43] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 1, 2, 8

[44] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023. 1

[45] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023. 3

[46] Liang Zhao, En Yu, Zheng Ge, Jinrong Yang, Haoran Wei, Hongyu Zhou, Jianjian Sun, Yuang Peng, Runpei Dong, Chunrui Han, et al. Chatspot: Bootstrapping multimodal llms via precise referring instruction tuning. *arXiv preprint arXiv:2307.09474*, 2023. 2, 3

[47] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022. 2

[48] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023. 2

[49] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004, 2016. 6