OMG-LLaVA : Bridging Image-level, Object-level, Pixel-level Reasoning and Understanding

Tao Zhang[1], **Xiangtai Li**[2,4] †, **Hao Fei**[2], **Haobo Yuan**[3], **Shengqiong Wu**[2], **Shunping Ji**[1], **Chen Change Loy**[3], **Shuicheng Yan**[2]
[1]Wuhan University [2]Skywork AI [3]S-Lab, NTU [4]Bytedance
Project page: https://lxtgh.github.io/project/omg_llava
Email: xiangtai94@gmail.com and zhang_tao@whu.edu.cn

Abstract

Current universal segmentation methods demonstrate strong capabilities in pixel-level image and video understanding. However, they lack reasoning abilities and cannot be controlled via text instructions. In contrast, large vision-language multimodal models exhibit powerful vision-based conversation and reasoning capabilities but lack pixel-level understanding and have difficulty accepting visual prompts for flexible user interaction. This paper proposes OMG-LLaVA, a new and elegant framework combining powerful pixel-level vision understanding with reasoning abilities. It can accept various visual and text prompts for flexible user interaction. Specifically, we use a universal segmentation method as the visual encoder, integrating image information, perception priors, and visual prompts into visual tokens provided to the LLM. The LLM is responsible for understanding the user's text instructions and providing text responses and pixel-level segmentation results based on the visual information. We propose perception prior embedding to better integrate perception priors with image features. OMG-LLaVA achieves image-level, object-level, and pixel-level reasoning and understanding in a single model, matching or surpassing the performance of specialized methods on multiple benchmarks. Rather than using LLM to connect each specialist, our work aims at end-to-end training on one encoder, one decoder, and one LLM. The code and model have been released for further research.

1 Introduction

With the development of transformer models [94; 6; 93; 40; 71; 92; 64; 126; 49; 87; 10; 19; 58], recent works in both natural language processing (NLP) and computer vision raise one common trend: adopting one unified model to solve multiple tasks. For example, large language models (LLMs) [93; 40; 92] adopt scale-up models to solve multiple NLP tasks and achieve better results than previous expert models. In vision, we have also seen a similar trend [19; 58; 100; 99; 47; 112], adopting one model to solve multiple tasks or sub-tasks, including detection, segmentation, video analysis, low-level vision, pose estimations, and more tasks. Different methods adopt different transformer designs, including visual-in-context learning [99; 100], unified decoder [19; 58], and unified tokenizer [86; 16; 58]. In summary, benefiting from the *scalability* and *flexibility* of the transformer, adopting one model for all tasks has made a great progress [19; 71; 72; 70; 126; 88; 87].

Work done when Tao Zhang is an intern at Skywork AI. †: Project leader. Corresponding author: Xiangtai Li and Shunping Ji.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

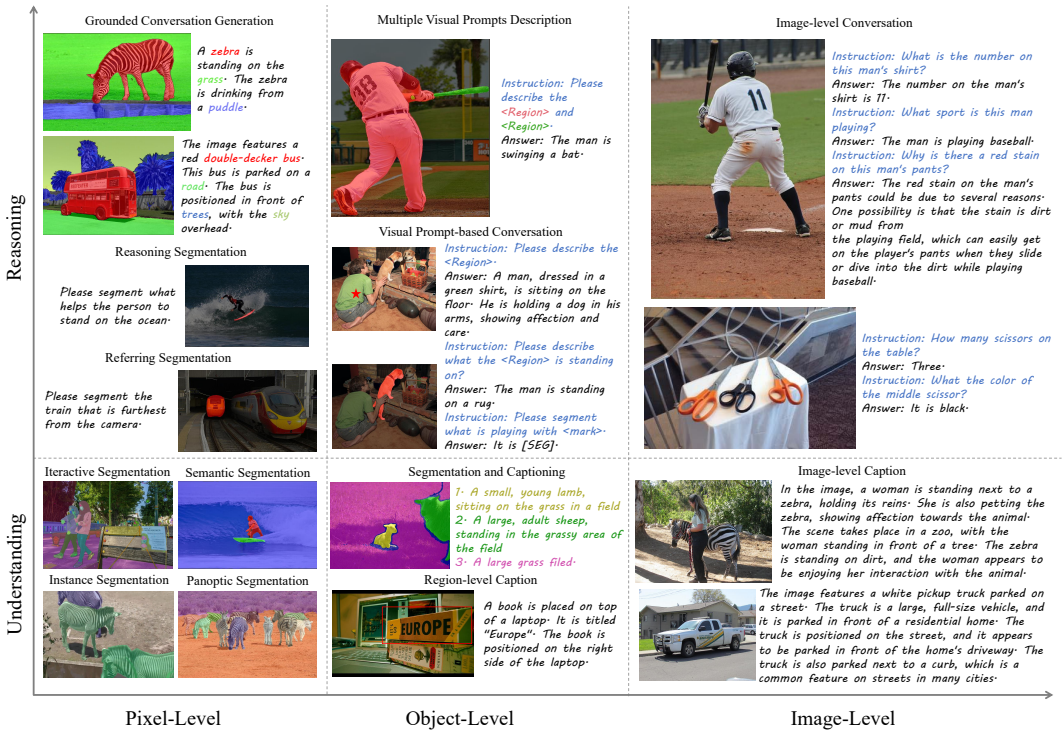


Figure 1: The comprehensive capabilities of OMG-LLaVA. OMG-LLaVA can handle a variety of pixel-level, object-level, and image-level understanding and reasoning tasks.

Meanwhile, by combining vision models and language models [71; 72; 70; 64; 65; 107], research on multi-modal models also adopts transformer-based design. One representative work, LLaVA [71; 72; 70], treats visual tokens as the inputs of LLMs and makes LLMs understand visual contents. Several works adopt similar designs [3; 13; 64; 18; 25], and all of them are termed Multi-modal Large Language Models (MLLMs). After that, most research focuses on improving MLLM benchmarks in various ways, including increasing data sizes [14; 18; 70] and enhancing the visual encoders [131; 24; 18] and visual resolutions [110; 18; 65; 25]. However, LLaVA-like models cannot output precise location information since they only carry out image-level analysis. Thus, recent works [126; 133; 11; 82; 13; 119; 128; 87; 67] try to fill this gaps by adding extra detection models for object level analysis, mask decoder for pixel-level analysis, visual prompts, and also propose task-specific instruction tuning with various datasets. By providing extra detection data and a decoder, the updated MLLMs can perform localization output. However, these models [135; 96; 49] are specifically tuned on specific tasks, losing the ability of LLaVA for image level analysis, such as caption and visual question answering. Meanwhile, several works [126; 49; 87; 78] adopt LLMs as agents to collaborate with various visual models or generation models. Despite the works being simple and effective, the inference and parameter costs are huge due to the multiple visual encoders and decoders. Moreover, there are no specific designs for task unification.

Motivated by the previous analysis, we ask one essential question: Can we bridge image-level, object-level, and pixel-level tasks into one MLLM model with only one LLM, one visual encoder, and one visual decoder? Back to the universal perception models, we can leverage these models to help us build a stronger MLLM to unify three-level inputs, including image, object, and pixel levels. In particular, we adopt OMG-Seg [58] as our universal perception model due to its simplicity and effectiveness in various segmentation tasks.

In this work, we present OMG-LLaVA, an elegant MLLM that bridges image-level, object-level, and pixel-level reasoning and understanding tasks in one model. We preserve the basic pixel-level segmentation ability of OMG-Seg by freezing the visual encoder and decoder, as shown in the bottom left of Fig. 1. Since the LLM processes text input, OMG-LLaVA can also perform referring segmentation, reasoning segmentation, and grounded conversation and generation, shown in the top left of Fig. 1. Moreover, as shown in Fig. 1, with the help of LLMs, OMG-LLaVA can also perform

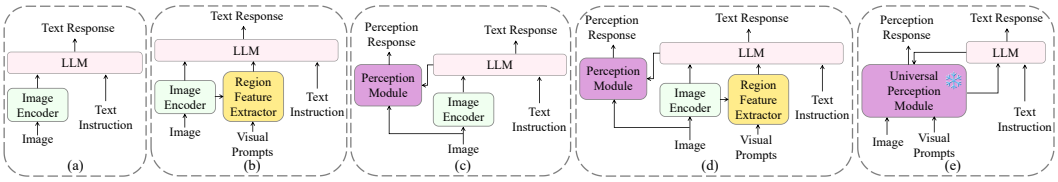


Figure 2: Summary of Current MLLM Architectures: (a) MLLMs with only image-level capability, including [71; 72; 70; 65], etc., (b) MLLMs with object-level capability, including [126; 87], (c) MLLMs with pixel-level capability, including [49; 88], etc., (d) MLLMs with both object-level and pixel-level capabilities but with a very complex system, such as [87], (e) OMG-LLaVA's architecture, which possesses an elegant and simple design while having image-level, object-level, and pixel-level capabilities.

image-level understanding as LLaVA, including caption and conversation, where most MLLMs for grounding lose such ability. In addition, OMG-LLaVA also supports the visual prompts as inputs, which results in object level understanding, such as visual prompt-based conversation and region-level captions. We achieve all these abilities using one LLM, one encoder, and one decoder.

In particular, to better encode the visual segmentation outputs, we propose a perception prior embedding module to absorb the object queries into object-centric visual tokens, which are the inputs of LLMs. We present a unified instruction formation strategy, which lets the model accept visual images, texts, and visual prompts as inputs and generate the response of text, segmentation tokens, segmentation masks, and labels. Following the LLaVA [71], we adopt pretraining and instruct tuning pipelines. Extensive experiments show the effectiveness of our components and training strategy. In addition to visual segmentation, OMG-LLaVA can also achieve good enough performance on 6 datasets, including COCO panoptic segmentation, VIPSeg video panoptic segmentation, refCOCO, refCOCO+, refCOCOg referring expression segmentation, GranDf grounded conversation generation, and refCOCOg region caption datasets. We hope our research can inspire the research on MLLM design in a more elegant way for the community.

2 Related Work

Multimodal Large Language Models. Early multimodal models [53] explore better fusion strategies, various feature extractors, and different meta-architectures. Most works focus on single tasks, such as caption and VQA. With the development of the large language models [6; 93; 40], recent works [52; 3; 92; 71; 17] mainly explore building an instruction-tuning pipeline for multiple multimodal benchmarks [39; 74; 62; 32]. LLaVA [71; 70; 69; 106; 136; 30; 28] is one earlier work that treats visual features as tokens. After that, several works [126] explore visual cues to enhance the visual inputs of LLaVA. On the other hand, several works [129; 127; 88; 131; 24; 25; 66; 130; 83; 38; 49] add extra components to adapt LLaVA for visual grounding, detection, segmentation, and video analysis. In particular, several works explore language-driven grounding and segmentation. However, these works are all trained with a specific purpose. We aim to build the simplest model to unify segmentation, instruction tuning, and prompt-driven segmentation in one model. To the best of our knowledge, we are the first model to achieve this goal.

Unified Segmentation Models. The vision transformers [10; 26; 79; 94] have led to research interest in universal segmentation. Recent works [95; 19; 123; 56; 21; 117; 115; 73; 91; 124; 122; 139; 111; 54; 141] have developed mask classification architectures with an end-to-end set prediction approach, outperforming previous specialized models [12; 46; 36; 57; 34; 60; 140] in both image, video and generalization segmentation tasks [45; 61; 59]. In particular, several works explore open-world segmentation, including entity segmentation [85; 84], open-vocabulary segmentation [125; 105]. Meanwhile, several works [58; 41; 111; 112; 33; 2] adopt one model with shared parameters to perform various segmentation tasks. One recent work, OMG-Seg [58], first unifies image, video, open-vocabulary, and interactive segmentation in one simple model. However, all of these works focus on visual segmentation and cannot generate interactive text and visual prompts, like MLLMs. Our work builds such a bridge to align MLLMs, visual segmentation, and prompt-driven segmentation models from joint co-training and model sharing, which serves as a new baseline for this field.

Language-driven Location and Segmentation. Early works [120; 68; 44; 23; 104; 135] in this direction mainly define the various language-driven tasks, including referring segmentation and

Table 1: Comparison of capabilities of different models. We include several representative methods here. Our OMG-LLaVA offers the most comprehensive capabilities, encompassing image-level, object-level, and pixel-level understanding and reasoning. Compared to [87; 35], OMG-LLaVA features an elegant and simple system architecture with only a single visual encoder.

Method	Visual Encoder	Image-level		Object-level			Pixel-level		
		Caption	Conversation	Visual Prompts	Caption	Conversation	Universal Seg	RES	GCG
LLAVA [71]	1	✓	✓						
MiniGPT4 [142]	1	✓	✓						
mPLUG-Owl [118]	1	✓	✓						
LLaMA-Adapter [132]	1	✓	✓						
Mini-Gemini [65]	2	✓	✓						
InternVL 1.5 [18]	1	✓	✓						
VisionLLM [97]	1	✓	✓					✓	
Shikra [13]	1	✓	✓	Point & Box	✓	✓			
Kosmos-2 [82]	1	✓	✓	Box	✓	✓			
GPT4RoI [133]	1	✓	✓	Box	✓	✓			
Ferret [119]	1	✓	✓	Point & Box & Mask	✓	✓			
Osprey [126]	1	✓	✓	Mask	✓	✓			
SPHINX-V [67]	1	✓	✓	Point & Box & Mask	✓	✓			
LISA [49]	2	✓	✓					✓	✓
GLAMM [87]	2	✓	✓	Box	✓	✓		✓	✓
Groundhog [134]	4	✓	✓	Point & Box & Mask	✓	✓		✓	✓
AnyRef [35]	2	✓	✓	Box	✓	✓		✓	
PixelLM [88]	1	✓	✓					✓	
GSVA [109]	2	✓	✓					✓	
Groma [78]	1	✓	✓	Box	✓	✓			
VIP-LLaVA [8]	1	✓	✓	Point & Box & Mask	✓	✓			
PSALM [135]	1			Point & Box & Mask			✓	✓	
LaSagnA [102]	2							✓	
OMG-Seg [58]	1			Point			✓		
OMG-LLaVA	1	✓	✓	Point & Box & Mask	✓	✓	✓	✓	✓

referring localization. Most works [31; 5; 116; 77; 103; 105] design effective fusion modules to achieve better performance. Meanwhile, several works [55; 103; 108; 49; 87; 126; 81] explore more complex language-driven tasks from various aspects, including robustness, reasoning, and region-level caption. LISA [114] involves reasoning-based segmentation. Then, GLaMM [87] annotates a new dataset and proposes region-level caption and segmentation tasks. Meanwhile, several works [29; 72] use LLMs as agents to assign different visual experts. In contrast to these works, our method is a more elegant baseline, which contains **only** one visual encoder, one LLM, and one decoder.

Visual Prompts. With the prompting ability of LLMs, several works [100; 99; 4; 138; 90; 51; 81] also explore visual prompting methods in vision. According to the design and purposes, these works can be divided into different aspects, including learnable tokens [138], mask-visual-modeling for different tasks [100; 27; 98], and various visual prompting encoders for visual outputs [99; 101; 125; 47]. Our OMG-LLaVa also supports visual prompts for better interaction with the user's inputs, showing the potential for product purposes.

3 Methodology

3.1 Task Unification

Motivation and Our Goals. The LLMs unify most NLP tasks as token generation tasks and exhibit strong reasoning and instruction-following capabilities. As shown in Fig. 2 (a), LLaVA-like models [71; 70; 69; 110; 65; 131; 24; 25; 18; 64] further introduce visual tokens into LLMs, enabling LLMs to understand visual information and perform visual-based reasoning. However, they cannot accomplish fine-grained visual tasks like object-level and pixel-level understanding and reasoning. As shown in Fig. 2 (b), [126; 133; 11; 82; 13; 119] introduce region-level visual embeddings, allowing LLMs to achieve object-level understanding and reasoning tasks. However, these models rely on complex region embedding extraction designs. In addition, most cannot perform pixel-level understanding tasks. Thus, as shown in Fig. 2 (c),[49; 88; 35] introduce segmentation tokens, enabling LLMs to output segmentation masks and thus handle pixel-level understanding and reasoning tasks. Nonetheless, they require a large segmentation module, such as SAM [47], making the system highly redundant. As shown in Fig. 2 (d), GLAMM [87] combines the above pipelines to handle object-level and pixel-level tasks. However, this significantly increases the system's *complexity* and *redundancy*. Additionally, GLAMM relies on explicit instructions from the user, **losing** the perception ability to handle basic pixel-level understanding tasks such as instance segmentation, semantic segmentation, panoptic segmentation, and interactive segmentation.

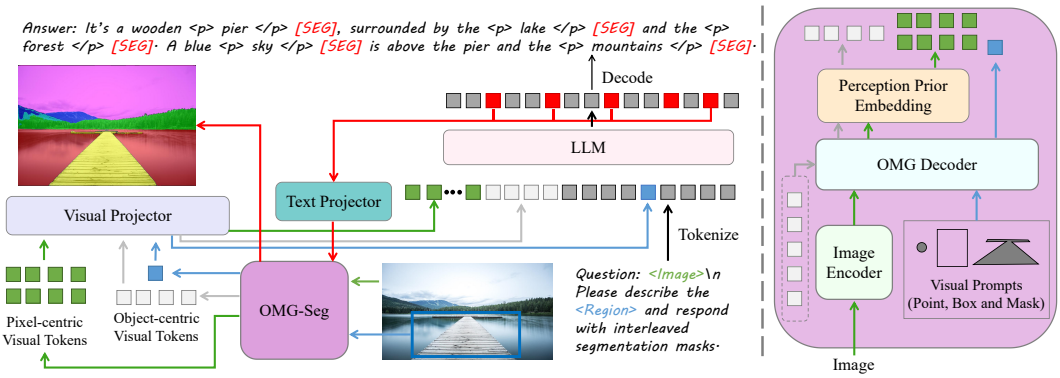


Figure 3: The Overview of OMG-LLaVA. OMG-LLaVA consists of OMG-Seg and LLM. OMG-Seg tokenizes the image into pixel-centric visual tokens, the detected objects, and inputs visual prompts into object-centric visual tokens. Additionally, the [SEG] token output by LLM is decoded by OMG-Seg into segmentation masks. OMG-Seg remains frozen at all stages.

In this paper, we focus on addressing all the challenges above in a more simple yet elegant way. Our OMG-LLaVA unifies image-level (such as image caption and image-based conversation), object-level (such as region caption and visual prompt-based conversation), and pixel-level (such as universal segmentation, referring segmentation, reasoning segmentation, and grounded conversation generation) visual understanding and reasoning tasks into token-to-token generation. The framework follows a simple and elegant system design, including only one visual perception module and one large language model.

Unified View of Different Tasks. We model various tasks as the token-to-token generation to bridge the gap between image-level, object-level, and pixel-level understanding and reasoning. To support these tasks, we define three types of tokens: text tokens T_t, pixel-centric visual tokens T_{pv}, and object-centric visual tokens T_{ov}. Text tokens encode textual information. Pixel-centric visual tokens represent dense image features, providing the LLM with comprehensive image information. Object-centric visual tokens encode the features of specified objects, offering the LLM object-centric information, and can be easily decoded into segmentation masks.

Then, all the tasks can be unified as:

$$T_t^{out}, T_{ov}^{out} = LLM(T_{pv}^{in}, T_{ov}^{in}, T_t^{in}) \tag{1}$$

For example, in the classic image-level understanding task, i.e., image caption, a text response T_t^{out} is generated based on text instruction T_t^{in} and image features T_{pv}^{in}. In the object-level understanding task, region captioning, the text response T_t^{out} is generated based on text instruction T_t^{in}, image features T_{pv}^{in}, and specified object-centric visual tokens T_{ov}^{in}. The pixel-level reasoning task, referring segmentation, involves generating object-centric visual tokens T_{ov}^{out} based on text instruction T_t^{in} and image features T_{pv}^{in}. Additionally, OMG-LLaVA can support various mixed-level tasks, such as providing grounded descriptions around specified objects.

Pixel-centric visual tokens can be obtained by tokenizing images using a CLIP backbone as the tokenizer. However, object-centric visual tokens require encoding object information to be easily decoded into segmentation masks. Therefore, methods like mask pooling in Osprey [126] and ROI pooling in GLaMM [87] fail to meet these requirements. We found that a universal perception decoder can meet all the requirements. Thus, we chose the OMG-Seg decoder [58] as the object-centric tokenizer due to its comprehensive capabilities.

3.2 OMG-LLaVA Framework

The framework of OMG-LLaVA is shown in Fig. 2 (e). OMG-LLaVA comprises a large language model (LLM) and a *frozen* universal perception module. The universal perception module encodes images and visual prompts from users into pixel-centric and object-centric visual tokens. It obtains object-centric visual tokens output by the LLM into explicit segmentation mask responses. The LLM accepts text instruction tokens and pixel-centric and object-centric visual tokens from the universal perception module as inputs and then outputs text responses along with object-centric visual tokens. The detailed architecture of OMG-LLaVA is illustrated in Fig. 3. The universal perception module

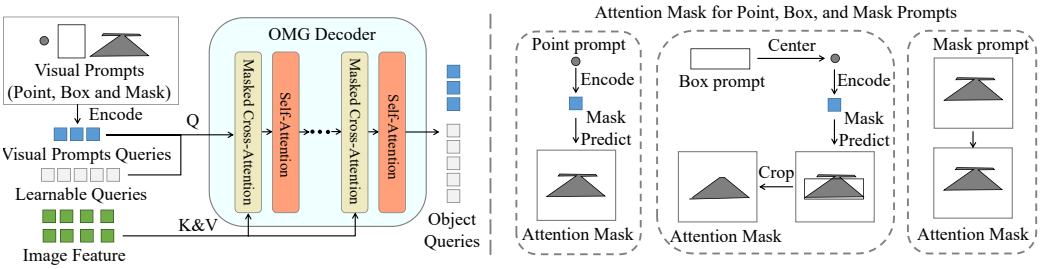


Figure 4: The Architecture of the OMG Decoder. A simple attention mask generation strategy enables the OMG decoder to encode point, box, and mask prompts.

comprises an image encoder, an OMG decoder [58], and a non-trainable perception prior embedding component.

Image Encoder. To maximize the perception capabilities of the universal perception module, we use the ConvNeXt-L [75]-based CLIP [86] model as the image encoder and employ a high image resolution (1024×1024). However, the large image resolution results in excessive visual tokens input into the LLM, leading to significantly higher computational costs than using lower-resolution images (such as 224×224 or 336×336). We address this issue by utilizing the lowest resolution image features (32× downsampling). Additionally, we use the pixel shuffle operator to further reduce the image features' resolution. Ultimately, the downsampling factor for the image features used to generate visual tokens is 64, meaning that a 1024×1024 image produces 256 visual tokens.

OMG Decoder. We utilize the OMG decoder [58] to generate object-centric visual tokens, furnishing the LLM with information regarding the primary objects in the image and those mentioned by the user's input visual prompts. As shown on the left side of Fig. 4, the OMG decoder comprises masked cross-attention [19] and self-attention layers. The OMG decoder's input includes a set of learnable object queries [20; 19; 10] for automatically capturing all objects of interest and visual prompt queries derived from encoded input visual prompts [47]. The visual prompt queries and learnable object queries are collectively termed object queries. The OMG decoder probes feature for object queries from the image features by employing masked cross-attention and models relationships between objects through self-attention. The object queries can be decoded into segmentation masks and object categories via a simple FFN layer. With the OMG decoder, OMG-LLaVA can efficiently tokenize object information into object-centric visual tokens, thereby equipping the LLM with information about objects in the image and those referenced by the user.

The OMG decoder can accept point prompts as input. While box and mask prompts can be easily converted into point prompts, this crude conversion significantly loses prompt information, complicating the explicit encoding of the user's intent. To address this, we can impose constraints on the attention masks of the masked cross-attention layers based on the visual prompt to precisely encode the object information referenced by the prompt. As depicted on the right side of Fig. 4, we utilize the box coordinates to define attention masks for all pixel features outside the box for box prompts. Similarly, we directly employ the provided object mask to generate attention masks for mask prompts. With this straightforward attention mask modification strategy, OMG-LLaVA can accurately capture the user's visual prompts, encompassing point, box, and mask prompts.

Perception Prior Embedding. We find that directly combining a frozen perception module with LLM doesn't perform well, as also observed in LISA [49]. To retain the full capabilities of the universal perception module, OMG-LLaVA doesn't fine-tune the perception module to adapt to the output of the large language model. Instead, we propose

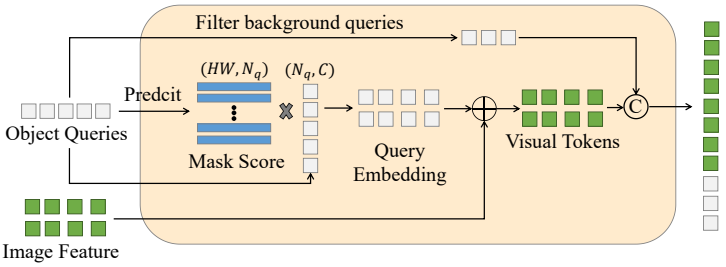


Figure 5: The process of the perception prior embedding strategy. The perception prior embedding strategy integrates object queries into image features based on segmentation prior.

a perception prior embedding strategy to tackle this challenge. Fig. 5 illustrates the perception prior embedding strategy.

First, we fuse the image features $\mathcal{F} \in \mathbb{R}^{HW \times C}$ outputted by the image encoder with the object queries $\mathcal{Q} \in \mathbb{R}^{N_q \times C}$ outputted by the OMG decoder $\mathcal{D}$. Specifically, we utilize the segmentation mask $\mathcal{M} \in \mathbb{R}^{N_q \times HW}$ obtained from the object queries and the corresponding confidence score $\mathcal{S} \in \mathbb{R}^{1 \times N_q}$ to derive a mask score $MS \in \mathbb{R}^{HW \times N_q}$ for each pixel for the object queries:

$$MS = Softmax(\mathcal{M} \odot \mathcal{S}, dim = -1) \tag{2}$$

Then, we compute a weighted average of the object queries $\mathcal{Q}$ based on the mask score MS and obtain the corresponding weighted object queries for each pixel. Pixel-centric visual tokens T_{pv} are obtained by adding the weighted object queries to the image features $\mathcal{F}$:

$$T_{pv} = MS \cdot \mathcal{Q} + \mathcal{F} \tag{3}$$

Additionally, we treat the foreground object queries as object-centric visual tokens T_{ov}. The object-centric visual tokens T_{ov} are concatenated with the pixel-centric visual tokens T_{pv} to form the visual tokens $T_v = (T_{pv}, T_{ov})$, which are input to the LLM to provide rich perception prior information.

Visual Projector and Text Projector. Following [71], we use an MLP as the visual projector, which is responsible for mapping visual tokens to the LLM's text embedding space. Since our visual tokens are pixel-centric and object-centric tokens, the visual projector comprises two MLPs, each handling one type of visual token separately. Inspired by [49; 87], we also use a simple MLP to map the LLM output's hidden states of the [SEG] token to the visual space.

Instruction Formulation. OMG-LLaVA can accept **visual** input, **text** input, and **visual prompt** input and output text responses and segmentation token, segmentation masks and labels. Thus, it can handle tasks such as image captioning, image-based conversation, region captioning, visual prompt-based conversation, referring segmentation, reasoning segmentation, grounded conversation, etc. We use a unified instruction formulation to support these functionalities. As shown in Fig. 3, there are three special tokens: <Image>, <Region>, and [SEG]. Before being fed into the LLM, the <Image> token is replaced by visual tokens T_v, and the <Region> token can be replaced by any object-centric visual token encoded by the visual prompt. The [SEG] token in the LLM's output is sent to the frozen OMG decoder to be decoded into a segmentation mask.

3.3 Training and Testing Setup

Training. Following LLaVA [71], our OMG-LLaVA performs two-stage training: pretraining and instruction tuning. During the pretraining stage, the perception model and LLM are frozen, and only the visual and text projectors can be tuned. In addition to the text regression loss, we apply regularization penalties to the visual projector $\mathcal{P}_v$ and text projector $\mathcal{P}_t$ to preserve object-centric information as much as possible.

$$\mathcal{L}_{pretrain} = \mathcal{L}_{text} + \mathcal{L}_{reg}, \quad \mathcal{L}_{reg} = (T_{ov} - \mathcal{P}_t(\mathcal{P}_v(T_{ov})))^2 \tag{4}$$

During instruction tuning, in addition to finetuning the visual projector and text projector, we use LoRA [37] to finetune the LLM. Following [87; 58], besides the text regression loss, we apply cross-entropy loss and dice loss [80] to supervise the segmentation mask decoded by the [SEG] token, as shown in following (We set $\alpha = 5$ $\beta = 2$ by default):

$$\mathcal{L}_{intruction} = \mathcal{L}_{text} + \mathcal{L}_{mask}, \quad \mathcal{L}_{mask} = \alpha\mathcal{L}_{CE} + \beta\mathcal{L}_{DICE} \tag{5}$$

Testing. The image-level, object-level, and pixel-level understanding and reasoning tasks can all be encompassed within the Eq. 3.1 paradigm. During the inference stage, we encode the necessary task requirements, such as text prompts, visual prompts, and image features, into tokens to input into the LLM. The output tokens of LLM are then decoded into text responses and segmentation mask responses according to the task definition. We refer the readers to check the more details in the appendix.

Table 2: The comprehensive comparison of OMG-LLaVA and other MLLMs regarding pixel-level and object-level understanding and reasoning capability and performance. "-" indicates that the method does not handle this task. † indicates that the method used the GranD dataset [87] for pretraining, which is significantly larger than the datasets used by other methods.

Method	Visual Encoder Num	COCO PQ	VIPseg VPQ	refCOCO cIoU	refCOCO+ cIoU	GCG METEOR	GCG AP50	refCOCOg(C) METEOR
OSprey [126]	1	-	-	-	-	-	-	16.6
LISA [49]	2	-	-	74.1	62.4	13.0	25.2	-
NeXT-Chat [128]	2	-	-	74.7	65.1	-	-	12.0
LaSagnA [102]	2	-	-	76.8	66.4	-	-	-
GSVA [109]	2	-	-	76.4	64.5	-	-	
AnyRef [35]	2	-	-	74.1	64.1	-	-	16.2
GLaMM† [87]	2	-	-	79.5	72.6	15.2	28.9	15.7
PixelLM [88]	1	-	-	73.0	66.3	-	-	-
OMG-LLaVA	1	53.8	49.8	78.0	69.1	14.9	29.9	15.3

Table 3: Performance on referring expression segmentation datasets. The evaluation metric is cIoU. "ft" indicates finetuning on the referring expression datasets.

Method	Freeze Decoder	Visual Encoder	refCOCO Val	refCOCO TestA	refCOCO TestB	refCOCO+ Val	refCOCO+ TestA	refCOCO+ TestB	refCOCOg Val	refCOCOg Test
LISA [49]	×	2	74.1	76.5	71.1	62.4	67.4	56.5	66.4	68.5
LISA(ft) [49]	×	2	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
PixelLM [88]	×	1	73.0	76.5	68.2	66.3	71.7	58.3	69.3	70.5
GSVA(ft) [109]	×	2	77.2	78.9	73.5	65.9	69.6	59.8	72.7	73.3
OMG-LLaVA	✓	1	75.6	77.7	71.2	65.6	69.7	58.9	70.7	70.2
OMG-LLaVA(ft)	×	1	78.0	80.3	74.1	69.1	73.1	63.0	72.9	72.9
OMG-LLaVA(ft)	✓	1	77.2	79.8	74.1	68.7	73.0	61.6	71.7	71.9

Table 4: Performance on grounded conversation generation datasets. “ft” indicates finetuning on the GranDf [87] dataset. † indicates that the method used the GranD dataset [87] for pretraining.

Methods	ft	Visual Encoder	Val METEOR	Val CIDEr	Val AP50	Val mIOU	Test METEOR	Test CIDEr	Test AP50	Test mIOU
Kosmos-2 [82]	✓	1	16.1	27.6	17.1	55.6	15.8	27.2	17.2	56.8
LISA [49]	✓	2	13.0	33.9	25.2	62.0	12.9	32.2	24.8	61.7
GLaMM† [87]	✓	2	15.2	43.1	28.9	65.8	14.6	37.9	27.2	64.6
OMG-LLaVA	×	1	13.8	36.2	26.9	64.6	13.5	33.1	26.1	62.8
OMG-LLaVA	✓	1	14.9	41.2	29.9	65.5	14.5	38.5	28.6	64.7

Table 5: Ablation study on RES and GCG datasets.

Methods	refCOCO cIoU	refCOCO gIoU	refCOCO+ cIoU	refCOCO+ gIoU	refCOCOg cIoU	refCOCOg gIoU	GCG METEOR	GCG mIoU
Baseline (M0)	58.7	61.0	52.6	55.0	55.8	58.1	13.2	51.0
+ Perception prior embedding (M1)	72.5	74.3	63.2	65.4	67.8	70.6	13.6	62.1
+ Object query input (M2)	74.4	75.9	64.4	66.2	68.5	71.5	13.8	63.6

4 Experiment

Dataset Setup. During the pretraining stage, we use the LLaVA pretraining dataset [71] to perform visual-text alignment, following LLaVA. The instruction tuning process of OMG-LLaVA involves a diverse range of tasks and datasets. For image-level understanding and reasoning tasks, we use the LLaVA dataset [71; 72; 70], which includes 665K descriptions, reasoning, and conversation data. For object-level understanding and reasoning, we use the object-level description and conversation data from the Osprey dataset [126] and the object-level point-prompt data from the MDVP dataset [67], which contain approximately 74K and 200K data, respectively. For pixel-level understanding and reasoning, we use the referring segmentation datasets, including refCOCO, refCOCO+ [42], refCOCOg [121], and refClef, totaling 74K data. Additionally, semantic segmentation datasets, including ADE20k [137] and COCO-stuff [7], totaling 26K data, and the grounded conversation generation dataset GranDf [87], containing 200K data, are used.

Implementation Details. We use the pre-trained ConvNext-L [75] OMG-Seg [58] as the universal perception module and InterLM2-7B [9] as the LLM for OMG-LLaVA. We adopt xtuner codebase [22] to build our model and data pipeline. The image is resized to 1024×1024. During the pretraining stage, only the visual projector and text projector are trained, with an initial learning rate set to 1e-3. During the instruction tuning stage, the initial learning rate is set to 2e-4, with only the perception model kept frozen, and the LLM is fine-tuned using LoRA [37]. The maximum sequence length in the LLM is set to 2,048. All training is conducted on four NVIDIA A800 GPUs with 80GB of memory. The pretraining stage and instruction tuning stage took 7 hours and 48 hours, respectively.

4.1 Main Results

Comparison with MLLMs. OMG-LLaVA is comprehensively compared with current MLLMs with perception capabilities, and the results are shown in Tab. 2. OMG-LLaVA demonstrates the most comprehensive capabilities. It achieves performance comparable to the SOTA in referring segmentation, grounded conversation generation, and region captioning. Additionally, OMG-LLaVA retains basic segmentation ability, enabling it to handle universal image and video segmentation tasks. Compared to other MLLMs, OMG-LLaVA features a simple and elegant system design, incorporating only a single visual encoder.

Referring Expression Segmentation. We evaluate OMG-LLaVA on refCOCO, refCOCO+, and refCOCOg, with the results shown in Tab. 3. OMG-LLaVA outperforms LISA [49] by 1.5 cIoU, 3.2 cIoU, and 4.3 cIoU on the validation sets of refCOCO, refCOCO+, and refCOCOg, respectively, while keeping the OMG decoder frozen and using only a single visual encoder. When we unfreeze the OMG decoder and finetune OMG-LLaVA on the referring expression segmentation task, OMG-LLaVA achieves 78.0, 69.1, and 72.9 cIoU on refCOCO, refCOCO+, and refCOCOg, respectively, surpassing LISA by 3.1, 4.0, and 5.0 cIoU. Compared to PixelLM [88], OMG-LLaVA shows performance improvements of 5.0 cIoU and 3.6 cIoU on refCOCO and refCOCOg, respectively.

Grounded Conversation Generation. Grounded conversation generation is a comprehensive and complex task that involves both image-level and pixel-level understanding and reasoning. MLLMs need to have the ability to provide fine-grained image descriptions and pixel-level understanding, linking the objects in the image captions to the corresponding segmentation masks. As shown in Tab. 4, when trained with comparable data, OMG-LLaVA surpasses LISA [49] by 1.9 METEOR and 7.3 CIDEr in image description ability. In terms of pixel understanding, OMG-LLaVA also outperforms LISA by 4.7 AP50 and 3.5 mIoU, even though LISA uses SAM and finetunes its segmentation decoder. Despite GLaMM [87] using much more training data than OMG-LLaVA, OMG-LLaVA demonstrates comparable pixel-understanding capabilities, outperforming GLaMM with 0.6 CIDEr, 1.4 AP50 and 0.1 mIoU on the test set.

4.2 Ablation and Analysis

Ablation Study. We conduct ablation studies on referring expression segmentation and grounded conversation generation datasets, with all training and testing settings consistent with the main experiments. We use a simple combination of OMG-Seg [58] and LLaVA [71] as our baseline, similar to LISA [49], where the [SEG] tokens output by the LLM were input into OMG-Seg to obtain segmentation masks, with OMG-Seg kept frozen.

As shown in Tab. 5, the baseline performed poorly on the RES datasets. Similarly, it exhibited low segmentation quality on the GCG dataset. This is because the LLM did not acquire any segmentation priors and needed to generate segmentation queries based on image features and adapt them to the input of the frozen perception module, which is a challenging task.

Figure 6: Visualization of the effectiveness of the proposed strategies. The left part shows the baseline (M0 in Tab. 5), the middle part shows the model with perception prior embedding (M1 in Tab. 5), and the right part shows the model with both perception prior embedding and object query input (M2 in Tab. 5).

When using our proposed perception prior embedding strategy, OMG-LLaVA exhibits performance gains of 13.8 cIoU, 10.6 cIoU, and 11.7 cIoU on refCOCO, refCOCO+, and refCOCOg, respectively. Additionally, the perception prior embedding strategy also brings a performance improvement of 11.1 mIoU on the GCG dataset and a slight improvement in image description capability (0.4 METEOR). When foreground object queries were provided to the LLM, OMG-LLaVA further improved its performance by 1.9 cIoU on refCOCO and 1.5 mIoU on GCG.

We conducted a visualization analysis of the proposed strategies. As shown in the left part of Fig. 6, the simple baseline has poor capability in associating text and segmentation, which is the crucial reason for its poor performance on RES. When using our proposed perception prior embedding strategy, the object query and pixel features are explicitly integrated according to the perception prior, resulting in significantly enhanced text-segmentation association capability. By adopting the object query input strategy, the quality of some challenging segmentation cases, such as the lower right corner of the fence in Fig 6, slightly improves.

Qualitative Results. We provide visualization results of OMG-LLaVA on multiple image-level, object-level, and pixel-level tasks in Fig. 1. Additional qualitative visualization results or comparable visual results for referring expression segmentation and grounded conversation generation are presented in the appendix.

5 Conclusion

We present a new MLLM, OMG-LLaVA, which bridges image-level, object-level, and pixel-level understanding and reasoning in one model. Our method only contains one image encoder, one LLM, and one decoder. With proposed perception prior embedding and unified task instruction tuning, OMG-LLaVA can perform over 8 different multi-modal learning tasks, as well as preserving the visual perception ability of the OMG-Seg baseline. Our method can achieve comparable results compared with previous combined works with much fewer trainable parameters and computation costs. We hope our work can inspire the community to rethink the design of the MLLM meta-architecture to minimize the model components and maximize the MLLM's functionalities.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (grant No. 42171430). This work is also supported by the CCF-Kuaishou Large Model Explorer Fund. It is also partially supported under the RIE2020 Industry Alignment Fund Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from the industry partner(s).

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 23, 24

[2] Ali Athar, Alexander Hermans, Jonathon Luiten, Deva Ramanan, and Bastian Leibe. Tarvis: A unified architecture for target-based video segmentation. In *CVPR*, 2023. 3

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 2, 3

[4] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. In *NeurIPS*, 2022. 4

[5] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *CVPR*, 2022. 4

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 1, 3

[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 8

[8] Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P. Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Making large multimodal models understand arbitrary visual prompts. In *CVPR*, 2024. 4

[9] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024. 9, 23, 24

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 3, 6

[11] 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. *arXiv preprint arXiv:2308.13437*, 2023. 2, 4

[12] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 3

[13] 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. 2, 4

[14] 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. 2

[15] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024. 24

[16] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *ICLR*, 2022. 1

[17] Wei-Ge Chen, Irina Spiridonova, Jianwei Yang, Jianfeng Gao, and Chunyuan Li. Llava-interactive: An all-in-one demo for image chat, segmentation, generation and editing. *arXiv preprint arXiv:2311.00571*, 2023. 3

[18] 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. 2, 4

[19] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 3, 6

[20] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 6

[21] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 3

[22] XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. https://github.com/InternLM/xtuner, 2023. 9

[23] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *ICCV*, 2023. 3

[24] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024. 2, 3, 4

[25] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Zhe Chen, Xinyue Zhang, Wei Li, Jingwen Li, Wenhai Wang, Kai Chen, Conghui He, Xingcheng Zhang, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*, 2024. 2, 3, 4

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3

[27] Zhongbin Fang, Xiangtai Li, Xia Li, Joachim M Buhmann, Chen Change Loy, and Mengyuan Liu. Explore in-context learning for 3d point cloud understanding. *NeurIPS*, 2024. 4

[28] Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. Video-of-thought: Step-by-step video reasoning from perception to cognition. In *ICML*, 2024. 3

[29] Hao Fei, Shengqiong Wu, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Vitron: A unified pixel-level vision llm for understanding, generating, segmenting, editing. *arxiv preprint*, 2024. 4

[30] Hao Fei, Shengqiong Wu, Meishan Zhang, Min Zhang, Tat-Seng Chua, and Shuicheng Yan. Enhancing video-language representations with structural spatio-temporal alignment. *PAMI*, 2024. 3

[31] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder fusion network with co-attention embedding for referring image segmentation. In *CVPR*, 2021. 4

[32] 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. *arXiv preprint arXiv:2306.13394*, 2023. 3, 23, 24

[33] Xiuye Gu, Yin Cui, Jonathan Huang, Abdullah Rashwan, Xuan Yang, Xingyi Zhou, Golnaz Ghiasi, Weicheng Kuo, Huizhong Chen, Liang-Chieh Chen, et al. Dataseg: Taming a universal multi-dataset multi-task segmentation model. *arXiv preprint arXiv:2306.01736*, 2023. 3

[34] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. *NeurIPS*, 2022. 3

[35] Junwen He, Yifan Wang, Lijun Wang, Huchuan Lu, Jun-Yan He, Jin-Peng Lan, Bin Luo, and Xuansong Xie. Multi-modal instruction tuned llms with fine-grained visual perception. *arXiv preprint*

arXiv:2403.02969, 2024. 4, 8
[36] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3
[37] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 7, 9
[38] Kuan-Chih Huang, Xiangtai Li, Lu Qi, Shuicheng Yan, and Ming-Hsuan Yang. Reason3d: Searching and reasoning 3d segmentation via large language model. *arXiv preprint arXiv:2405.17427*, 2024. 3
[39] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 3
[40] Louis Martin Hugo Touvron, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023. 1, 3
[41] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. *CVPR*, 2023. 3
[42] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014. 8
[43] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, pages 235–251. Springer, 2016. 23, 24
[44] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV*, 2018. 3
[45] Dahun Kim, Jun Xie, Huiyu Wang, Siyuan Qiao, Qihang Yu, Hong-Seok Kim, Hartwig Adam, In So Kweon, and Liang-Chieh Chen. Tubeformer-deeplab: Video mask transformer. In *CVPR*, 2022. 3
[46] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 3
[47] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *ICCV*, 2023. 1, 4, 6
[48] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 26
[49] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 1, 2, 3, 4, 6, 7, 8, 9, 23, 24, 25, 26
[50] 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. 23, 24
[51] Feng Li, Qing Jiang, Hao Zhang, Tianhe Ren, Shilong Liu, Xueyan Zou, Huaizhe Xu, Hongyang Li, Chunyuan Li, Jianwei Yang, et al. Visual in-context prompting. *arXiv preprint arXiv:2311.13601*, 2023. 4
[52] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 3
[53] 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. 3
[54] Xiangtai Li, Henghui Ding, Wenwei Zhang, Haobo Yuan, Guangliang Cheng, Pang Jiangmiao, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *arXiv pre-print*, 2023. 3
[55] Xiang Li, Jinglu Wang, Xiaohao Xu, Xiao Li, Bhiksha Raj, and Yan Lu. Robust referring video object segmentation with cyclic structural consensus. In *ICCV*, 2023. 4
[56] Xiangtai Li, Shilin Xu, Yibo Yang, Guangliang Cheng, Yunhai Tong, and Dacheng Tao. Panoptic-partformer: Learning a unified model for panoptic part segmentation. In *ECCV*, 2022. 3
[57] Xiangtai Li, Ansheng You, Zeping Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *ECCV*, 2020. 3
[58] Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation? *CVPR*, 2024. 1, 2, 3, 4, 5, 6, 7, 9, 26
[59] Xiangtai Li, Haobo Yuan, Wenwei Zhang, Guangliang Cheng, Jiangmiao Pang, and Chen Change Loy. Tube-link: A flexible cross tube baseline for universal video segmentation. In *ICCV*, 2023. 3
[60] Xiangtai Li, Li Zhang, Guangliang Cheng, Kuiyuan Yang, Yunhai Tong, Xiatian Zhu, and Tao Xiang. Global aggregation then local distribution for scene parsing. *IEEE TIP*, 2021. 3
[61] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video k-net: A simple, strong, and unified baseline for video segmentation. In *CVPR*, 2022. 3
[62] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *EMNLP*, 2023. 3
[63] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. 2023. 23, 24
[64] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023. 1, 2, 4
[65] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv*

preprint arXiv:2403.18814, 2024. 2, 3, 4
[66] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 3
[67] Weifeng Lin, Xinyu Wei, Ruichuan An, Peng Gao, Bocheng Zou, Yulin Luo, Siyuan Huang, Shanghang Zhang, and Hongsheng Li. Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want. *arXiv preprint arXiv:2403.20271*, 2024. 2, 4, 8
[68] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *CVPR*, 2023. 3
[69] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 3, 4, 23, 24
[70] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. 1, 2, 3, 4, 8
[71] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 3, 4, 7, 8, 9
[72] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. *arXiv preprint arXiv:2311.05437*, 2023. 1, 2, 3, 4, 8
[73] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3
[74] 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. 3, 23, 24
[75] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *CVPR*, 2022. 6, 9
[76] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 24
[77] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *CVPR*, 2020. 4
[78] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. *arXiv preprint arXiv:2404.13013*, 2024. 2, 4
[79] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2022. 3
[80] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 7
[81] Ting Pan, Lulu Tang, Xinlong Wang, and Shiguang Shan. Tokenize anything via prompting. *arXiv preprint arXiv:2312.09128*, 2023. 4
[82] 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. 2, 4, 8
[83] Lu Qi, Yi-Wen Chen, Lehan Yang, Tiancheng Shen, Xiangtai Li, Weidong Guo, Yu Xu, and Ming-Hsuan Yang. Generalizable entity grounding via assistance of large language model. *arXiv preprint arXiv:2402.02555*, 2024. 3
[84] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High-quality entity segmentation. In *ICCV*, 2023. 3
[85] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. *TPAMI*, 2022. 3
[86] 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 *ICML*, 2021. 1, 6
[87] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M. Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S. Khan. Glamm: Pixel grounding large multimodal model. *CVPR*, 2024. 1, 2, 3, 4, 5, 7, 8, 9, 23, 24, 25, 26
[88] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. *arXiv preprint arXiv:2312.02228*, 2023. 1, 3, 4, 8, 9, 23
[89] 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 *CVPR*, 2019. 26
[90] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, et al. Generative multimodal models are in-context learners. *arXiv:2312.13286*, 2023. 4
[91] Shuyang Sun, Weijun Wang, Andrew Howard, Qihang Yu, Philip Torr, and Liang-Chieh Chen. Remax: Relaxing for better training on efficient panoptic segmentation. *NeurIPS*, 2023. 3

[92] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1, 3

[93] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023. 1, 3

[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 1, 3

[95] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 3

[96] Junchi Wang and Lei Ke. Llm-seg: Bridging image segmentation and large language model reasoning. *arXiv preprint arXiv:2404.08767*, 2024. 2

[97] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023. 4

[98] Xinshun Wang, Zhongbin Fang, Xia Li, Xiangtai Li, and Mengyuan Liu. Skeleton-in-context: Unified skeleton sequence modeling with in-context learning. *CVPR*, 2024. 4

[99] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, 2023. 1, 4

[100] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. In *ICCV*, 2023. 1, 4

[101] Zhendong Wang, Yifan Jiang, Yadong Lu, Yelong Shen, Pengcheng He, Weizhu Chen, Zhangyang Wang, and Mingyuan Zhou. In-context learning unlocked for diffusion models. *arXiv preprint arXiv:2305.01115*, 2023. 4

[102] Cong Wei, Haoxian Tan, Yujie Zhong, Yujiu Yang, and Lin Ma. Lasagna: Language-based segmentation assistant for complex queries. *arXiv preprint arXiv:2404.08506*, 2024. 4, 8, 23

[103] Jianzong Wu, Xiangtai Li, Xia Li, Henghui Ding, Yunhai Tong, and Dacheng Tao. Towards robust referring image segmentation. *IEEE-TIP*, 2024. 4

[104] Jianzong Wu, Xiangtai Li, Chenyang Si, Shangchen Zhou, Jingkang Yang, Jiangning Zhang, Yining Li, Kai Chen, Yunhai Tong, Ziwei Liu, et al. Towards language-driven video inpainting via multimodal large language models. *CVPR*, 2024. 3

[105] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, and Dacheng Tao. Towards open vocabulary learning: A survey. *arXiv pre-print*, 2023. 3, 4

[106] Shengqiong Wu, Hao Fei, Xiangtai Li, Jiayi Ji, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Towards semantic equivalence of tokenization in multimodal llm. *arXiv preprint arXiv:2406.05127*, 2024. 3

[107] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *ICML*, 2024. 2

[108] Tsung-Han Wu, Giscard Biamby, David Chan, Lisa Dunlap, Ritwik Gupta, Xudong Wang, Joseph E Gonzalez, and Trevor Darrell. See, say, and segment: Teaching lmms to overcome false premises. *arXiv preprint arXiv:2312.08366*, 2023. 4

[109] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. *arXiv preprint arXiv:2312.10103*, 2023. 4, 8

[110] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024. 2, 4

[111] Shilin Xu, Haobo Yuan, Qingyu Shi, Lu Qi, Jingbo Wang, Yibo Yang, Yining Li, Kai Chen, Yunhai Tong, Bernard Ghanem, Xiangtai Li, and Ming-Hsuan Yang. Rap-sam: Towards real-time all-purpose segment anything. *arXiv preprint*, 2024. 3

[112] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *CVPR*, 2023. 1, 3

[113] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 23, 24

[114] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023. 4

[115] Zongxin Yang, Jiaxu Miao, Yunchao Wei, Wenguan Wang, Xiaohan Wang, and Yi Yang. Scalable video object segmentation with identification mechanism. *TPAMI*, 2024. 3

[116] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *CVPR*, 2022. 4
[117] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *TPAMI*, 2021. 3
[118] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl: Modularization empowers large language models with multimodality, 2023. 4
[119] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *ICLR*, 2024. 2, 4
[120] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 3
[121] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85. Springer, 2016. 8
[122] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Cmt-deeplab: Clustering mask transformers for panoptic segmentation. In *CVPR*, 2022. 3
[123] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV*, 2022. 3
[124] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means Mask Transformer. In *ECCV*, 2022. 3
[125] Haobo Yuan, Xiangtai Li, Chong Zhou, Yining Li, Kai Chen, and Chen Change Loy. Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively. *arXiv preprint*, 2024. 3, 4
[126] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. *arXiv preprint arXiv:2312.10032*, 2023. 1, 2, 3, 4, 5, 8
[127] Yuhang Zang, Wei Li, Jun Han, Kaiyang Zhou, and Chen Change Loy. Contextual object detection with multimodal large language models. *arXiv preprint arXiv:2305.18279*, 2023. 3
[128] Ao Zhang, Liming Zhao, Chen-Wei Xie, Yun Zheng, Wei Ji, and Tat-Seng Chua. Next-chat: An lmm for chat, detection and segmentation. *arXiv preprint arXiv:2311.04498*, 2023. 2, 8
[129] Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Lei Zhang, Chunyuan Li, and Jianwei Yang. Llava-grounding: Grounded visual chat with large multimodal models, 2023. 3
[130] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 3
[131] Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023. 2, 3, 4
[132] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 4
[133] 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. 2, 4
[134] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *CVPR*, pages 14227–14238, 2024. 4
[135] Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. Psalm: Pixelwise segmentation with large multi-modal model. *arXiv preprint arXiv:2403.14598*, 2024. 2, 3, 4
[136] Xiangyu Zhao, Xiangtai Li, Haodong Duan, Haian Huang, Yining Li, Kai Chen, and Hua Yang. Towards semantic equivalence of tokenization in multimodal llm. *arXiv preprint*, 2024. 3
[137] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *CVPR*, 2017. 8
[138] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. In *IJCV*, 2022. 4
[139] Qianyu Zhou, Zhengyang Feng, Qiqi Gu, Jiangmiao Pang, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Context-aware mixup for domain adaptive semantic segmentation. *IEEE TCSVT*, 2023. 3
[140] Qianyu Zhou, Xiangtai Li, Lu He, Yibo Yang, Guangliang Cheng, Yunhai Tong, Lizhuang Ma, and Dacheng Tao. Transvod: End-to-end video object detection with spatial-temporal transformers. *TPAMI*, 2022. 3
[141] Yikang Zhou, Tao Zhang, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Dvis-daq: Improving video segmentation via dynamic anchor queries. *arXiv preprint arXiv:2404.00086*, 2024. 3
[142] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*,

2023. 4

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We outlined the contributions in the abstract, specifically proposing an elegant MLLM architecture that achieves image-level, object-level, and pixel-level understanding and reasoning capabilities using only a single visual encoder.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations of the paper in the appendix, including the inability to segment part-level objects due to the constraints of OMG-Seg.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All details of the proposed method are included in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the code and model weights will be released upon paper acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All details is included in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: NA.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the experiments are conducted on 8 A800 80G GPUs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: No potential societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

Table 6: Performance on image-level benchmarks.

Method	MME [32]	MMBench [74]	SEED-Bench [50]	POPE [63]	AI2D [43]
Training only with LLaVA dataset					
LLaVA 1.5 [69]	1422/267	68.5	65.9	86.7	56.6
OMG-LLaVA	1448/282	67.5	68.9	89.7	61.7
Co-training with LLaVA dataset and segmentation datasets					
LISA [49]	1/1	0.4	-	0.0	0.0
PixelLM [88]	309/135	17.4	-	0.0	0.0
LaSagnA [102]	0/0	0.0	-	0.0	0.0
GLaMM [87]	14/9	36.8	-	0.94	28.2
OMG-LLaVA	1177/235	47.9	56.5	80.0	42.9

Table 7: Performance with different LLMs.

LLM	refCOCO		refCOCO+		MME		MMBench	SEED-Bench	POPE	AI2D	MMstar	SQA
	CIoU	GIoU	CIoU	GIoU	perception	reasoning						
Phi3-3.8B [1]	76.5	78.0	67.8	70.0	1291.6	265.0	59.6	60.6	86.7	56.9	37.1	64.7
InternLM2-7B [9]	76.3	77.8	67.7	69.9	1177.1	235.4	47.9	56.5	80.0	42.9	33.1	57.8
Qwen2-7B [113]	76.7	78.2	69.1	71.2	1215.7	251.1	62.8	60.7	84.3	52.6	37.2	66.4

A Appendix

Overview. In this appendix, we will first give more implementation and training details of our method. Then, we present more detailed ablation studies on several component designs. Next, we present more detailed visualization results. In the end, we discuss the limitations and future work.

A.1 More Implementation Details

Pre-training. Following LLaVA, OMG-LLaVA first performs pre-training to learn the projector that projects visual tokens into the text space. During the pre-training stage, we freeze the visual encoder, OMG head, and LLM to train the visual projector for projecting visual tokens into the text space and to train the text projector for restoring the projected object-centric visual tokens to the segmentation embedding. The training data used in the pre-training stage is the same as that used in LLaVA. In this stage, the OMG-LLaVA is trained for 1 epoch. The batch size is 256, with 32 per GPU, and the learning rate is 0.001.

Supervised fine-tuning. During the instruction tuning stage, we freeze the visual encoder and OMG head, finetune the LLM using LoRA, and fully finetune the text and visual projectors. We train OMG-LLaVA for 1 epoch on all instruction tuning datasets, including the LLaVA instruction tuning dataset, referring expression segmentation datasets, semantic segmentation datasets, grounded conversation generation datasets, mask-based visual prompt datasets, and point-based visual prompt datasets. The batch size is 128, with 16 per GPU, and the learning rate is 2e-4.

Inference details for each task. OMG-LLaVA generates answers token by token during the inference stage based on the given question. We use a fixed template for the referring expression segmentation task to create the question: "*Please segment {EXPRESSION} in this image.*" In rare cases where OMG-LLaVA does not predict the [SEG] token, we use an empty mask as the segmentation result. We use the fixed question for the grounded conversation generation task: "*Could you please give me a detailed description of the image? Please respond with interleaved segmentation masks for the corresponding parts of the answer.*" For other tasks, we remove special tokens such as , , and [SEG] from OMG-LLaVA's responses to ensure the answers contain only text.

A.2 More Experiment Results.

Evaluation results on image-level benchmarks. We evaluate OMG-LLaVA on several image-level benchmarks, including MME [32], MMBench [74], SEED-Bench [50], POPE [63], and AI2D [43] benchmarks. The evaluation results are shown in Tab. 6. When jointly Co-training with image-level and pixel-level datasets, OMG-LLaVA achieves 1412, 47.9, 46.5, 80.0 and 42.9 on MME, MMBench, SEED-Bench, POPE and AI2D benchmarks, respectively. Compared with GLaMM [87], PixelLM [88], and LISA [49], OMG-LLaVA demonstrates significant performance improvement.

Table 8: Ablation study of projector for object-centric visual tokens.

Methods	Cross Attn.	Individual	refCOCO		refCOCO+		refCOCOg		refCOCOg(C)
			cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	METEOR
Baseline (M0)			74.5	75.9	63.6	65.9	68.7	71.0	13.6
M1	✓		72.3	73.7	60.6	63.0	66.5	69.2	13.2
M2		✓	72.3	74.1	60.8	63.5	65.4	68.6	13.1

Table 9: Ablation study on answer format of segmentation-based tasks. The first row represents the RES task using the fixed answer: "*Sure, it is [SEG].*" and the GCG task using "* Expression [SEG].*" The second row represents the segmentation tasks' answer format being unified as "* Expression [SEG].*

Format	refCOCO		refCOCO+		refCOCOg		GCG		
	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	METEOR	AP50	mIOU
It is [SEG].	75.5	76.5	65.8	67.8	70.6	72.3	13.8	27.3	64.4
Expression [SEG].	75.6	76.8	65.6	67.6	70.7	72.6	13.8	26.9	64.6

Table 10: Ablation study on segmentation embeddings.

	refCOCO		refCOCO+		refCOCOg	
	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU
Last layer's hidden state	74.3	75.5	64.5	66.4	70.0	71.9
Mean of all layers' hidden states	74.3	75.8	64.5	66.4	68.7	71.4
Concatenate of all layers' hidden states	70.0	71.1	61.8	63.6	62.3	64.4

When training only with the LLaVA [69] dataset, OMG-LLaVA achieves 1730, 67.5, 68.9, 89.7, and 61.7 on MME, MMBench, SEED-Bench, POPE, and AI2D benchmarks. OMG-LLaVA outperforms LLaVA-1.5 [69] with 41, 3.0, 3.0, 5.1 on MME, SEED-Bench, POPE, and AI2D benchmarks with the same training data.

Performance with diverse LLMs. We construct the OMG-LLaVA using diverse LLMs. The performance is shown in Tab. 7. In addition to InternLM2 [9], we have tried using PHI-3.8b [1] and Qwen2-7B [113], which achieved better performance on pixel-level and image-level benchmarks than InternLM2. When using the stronger Qwen2-7B, OMG-LLaVA achieves 76.7 cIoU and 69.1 cIoU on RefCOCO and RefCOCO+ benchmarks, and 1466.8, 62.8, 60.7, 84.3, 52.6, 37.2, and 66.4 on MME [32], MMBench [74], SEED-Bench [50], POPE [63], AI2D [43], MMstar [15] and SQA [76] benchmarks.

A.3 More Detailed Ablation Studies.

Projector for object-centric visual tokens. We conducted ablation experiments on the vision projector. The results are shown in Tab. 8. We use a simple MLP projector as the baseline for object-centric visual tokens. When we added a cross-attention layer to the projector, performance on segmentation and visual prompt-based tasks decreased. This is because the introduction of the cross-attention layer caused the object-centric visual tokens to incorporate too many pixel-centric visual tokens, leading to interference with the object information. Furthermore, when the projector for object-centric visual tokens generated from visual prompt input and object queries is not shared, performance declines on segmentation and visual prompt-based tasks. Therefore, a shared MLP projector can effectively project object-centric visual tokens into the text space.

Answer format for segmentation-based tasks. In LISA [49], the response for the referring expression segmentation task is fixed as "*Sure, it is [SEG].*" However, this fixed answer may interfere with the instruction-following ability of the LLM, leading it to respond with "*Sure, it is [SEG].*" for new instructions. In GLaMM [87], for the grounded conversation generation task, the response is typically "* Expression [SEG].*" Since the "*Expression*" is flexible and variable, the LLM is less likely to overfit to a fixed response.

We conduct ablation experiments on the answer format for segmentation tasks, and the results are shown in Tab. 9. We find that unifying the answer format for segmentation tasks (including RES and GCG) as "* Expression [SEG]*" yields better performance. This more flexible answer

Figure 7: Qualitative comparison on the referring expression segmentation task. LISA uses the 13B LLM, while GLaMM and our proposed OMG-LLaVA use the 7B LLM.

format not only achieves better performance in the referring expression segmentation task compared to the fixed answer but also avoids the damage to the LLM's instruction-following ability.

Segmentation embeddings. We conduct ablation experiments on the generation strategy of segmentation embedding, and the results are shown in Tab. 10. We explore whether the hidden states of the intermediate layers corresponding to the [SEG] token are helpful for segmentation. Compared to using the hidden states of the last layer of the [SEG] token as the segmentation embedding, using the mean of the hidden states from all layers as the segmentation embedding resulted in negligible improvement on refCOCO but led to a significant performance drop on the more challenging refCOCOg. Concatenating the hidden states from all layers of the [SEG] token as the segmentation embedding resulted in a significant performance drop across all RES tasks. Therefore, the hidden state of the last layer already contains sufficient features to generate the segmentation mask, and introducing hidden states from other intermediate layers does not yield better segmentation results.

A.4 More Visualization Results

Qualitative comparison with SOTA methods. We conduct qualitative comparisons and analyses on various tasks, including referring expression segmentation, grounded conversation generation, and image-based conversation, against the SOTA methods LISA [49] and GLaMM [87]. Fig. 7 shows the visualization results of the RES task for LISA, GLaMM, and our proposed OMG-LLaVA. OMG-LLaVA demonstrates a more stable segmentation performance than LISA and GLaMM. Additionally, OMG-LLaVA exhibits better image and text understanding capabilities than LISA (13B) and GLaMM, as illustrated in the fourth column with the example of "*the smallest chair*".

Fig. 8 shows the visualization results of the GCG task for GLaMM [87] and OMG-LLaVA. Our proposed OMG-LLaVA provides more detailed and accurate descriptions of the scene, such as "*lighthouse*" and "*bear*." Additionally, OMG-LLaVA demonstrates more stable segmentation capabilities, as seen in the "*mountain*" in the bottom-right corner image.

Fig. 9 shows the visualization results of the visual prompt-based description task for GLaMM [87] and OMG-LLaVA. Compared to GLaMM, OMG-LLaVA supports more flexible visual prompts, including point, box, and mask prompts. Additionally, OMG-LLaVA can generate more detailed object captions and demonstrate a more accurate image understanding.

Fig. 10 shows the visualization results of the image-based conversation task for LISA [49], GLaMM [87], and OMG-LLaVA. Compared to LISA and GLaMM, OMG-LLaVA has stronger instruction-following ability. For example, when answering the question, "*What is the number on the jersey of the athlete squatting on the ground?*" both LISA and GLaMM incorrectly segmented "*the jersey of the athlete squatting on the ground*." Compared to GLaMM, OMG-LLaVA can provide more detailed and accurate answers to user questions. Compared to LISA, OMG-LLaVA demonstrates stronger scene understanding and reasoning abilities. For instance, in question 3 of Fig. 10, LISA gave an utterly incorrect answer despite using a larger LLM (13B).

Visualization results of RES. We provide additional visualization results of OMG-LLaVA on the RES task in Fig. 11. OMG-LLaVA demonstrates a strong understanding of spatial relationships and human actions, enabling it to accurately and reliably segment the specified objects based on these descriptions. Furthermore, even without training on any reasoning segmentation data, OMG-LLaVA exhibits the ability to perform reasoning segmentation. As shown in Fig. 12, OMG-LLaVA can infer the target based on the question and accurately segment the corresponding object.

Visualization results of GCG. As depicted in Fig. 13, our method performs well on the grounded conversation generation task. OMG-LLaVA demonstrates strong scene understanding and object segmentation capabilities. Although some objects are overlooked, this is due to the omission of many objects in the image captions of the Grandf dataset. We believe that using higher-quality data for training would result in even better performance for OMG-LLaVA.

Visualization results of visual prompts-based description. Fig. 14 shows more visualization results for the visual prompt-based description task. OMG-LLaVA supports input of point, box, and mask-based visual prompts and provides detailed descriptions. These descriptions include information about the objects and their relationships with other objects in the scene.

A.5 Limitation and Future Work Discussion

Limitations of OMG-LLaVA. Although OMG-LLaVA achieves image-level, object-level, and pixel-level capabilities with a concise and elegant architecture, much room still exists for improvement. Firstly, joint training with pixel-level understanding data often leads to decreased image-level capability, a phenomenon widely observed in LISA [49] and GLaMM [87]. This challenge could be addressed by organizing the data to eliminate this conflict. Secondly, due to the lack of multi-granularity segmentation capability in OMG-Seg, OMG-LLaVA cannot perform part-level segmentation. This challenge could be addressed using a more powerful and universal perception module by adding part-level visual inputs.

Future Works. Several future directions can be explored with our new meta-architecture. We list two potential directions, including video and more instruction-tuning data. Although OMG-Seg [58] can acquire the video inputs, OMG-LLaVA still cannot perform pixel-level spatial-temporal reasoning. This is due to the lack of such datasets. Moreover, more instruction-tuning data involve more localization outputs, and multiple round conversations can be used to build a stronger MLLM model. For example, we plan to use full GLaMM datasets [87] and more detection datasets [48; 89] for joint co-training as future work if more computation resources are available.

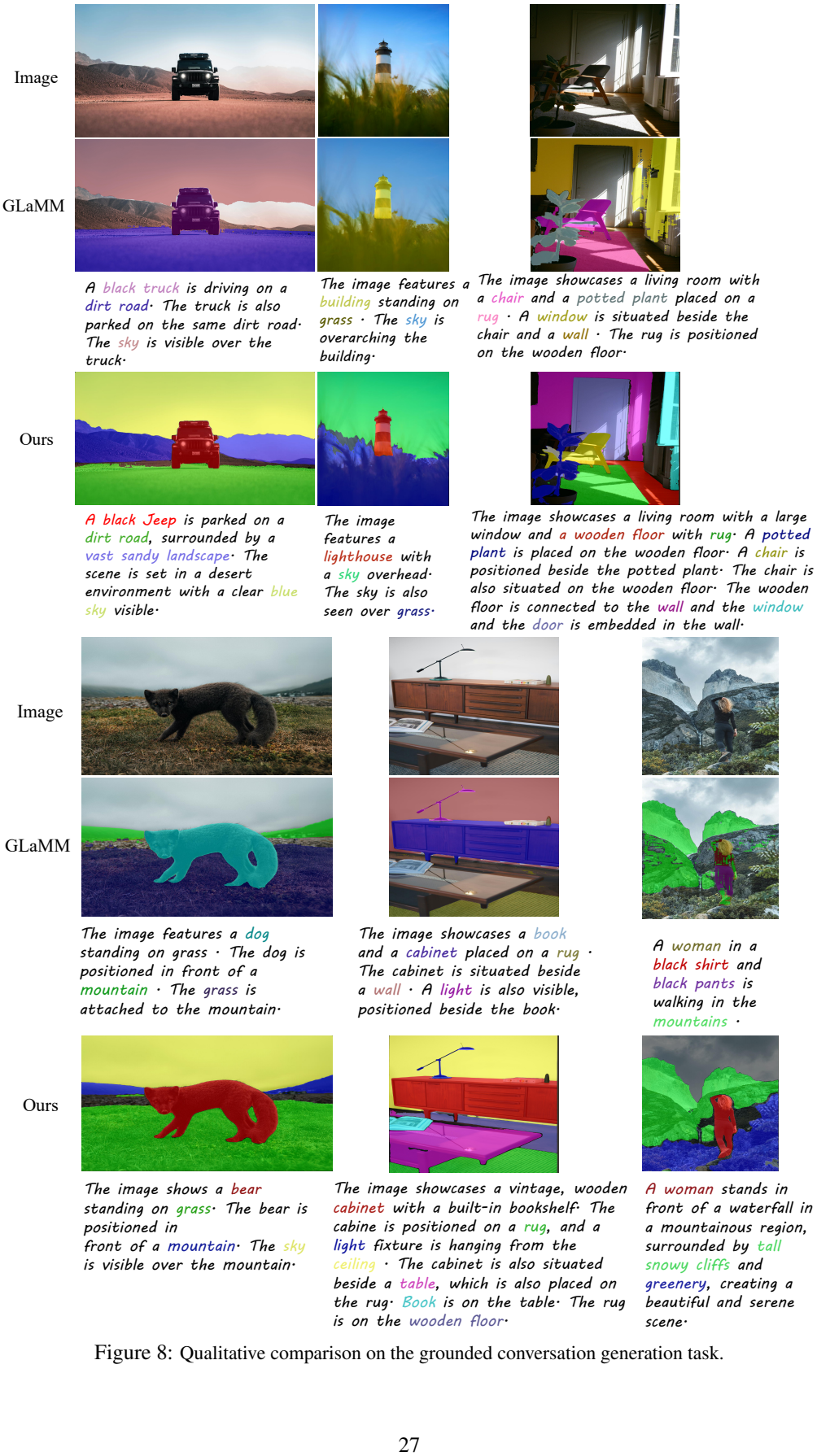


Figure 8: Qualitative comparison on the grounded conversation generation task.

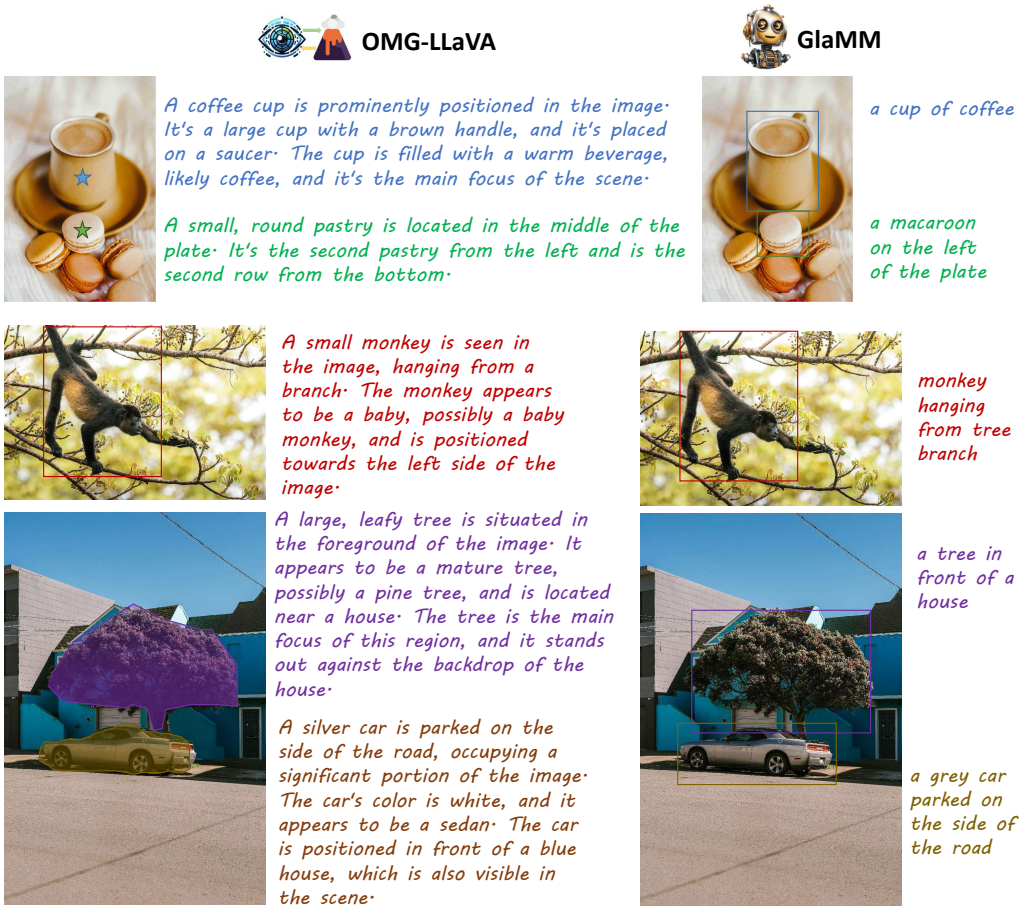


Figure 9: Qualitative comparison on the visual prompt-based description task.

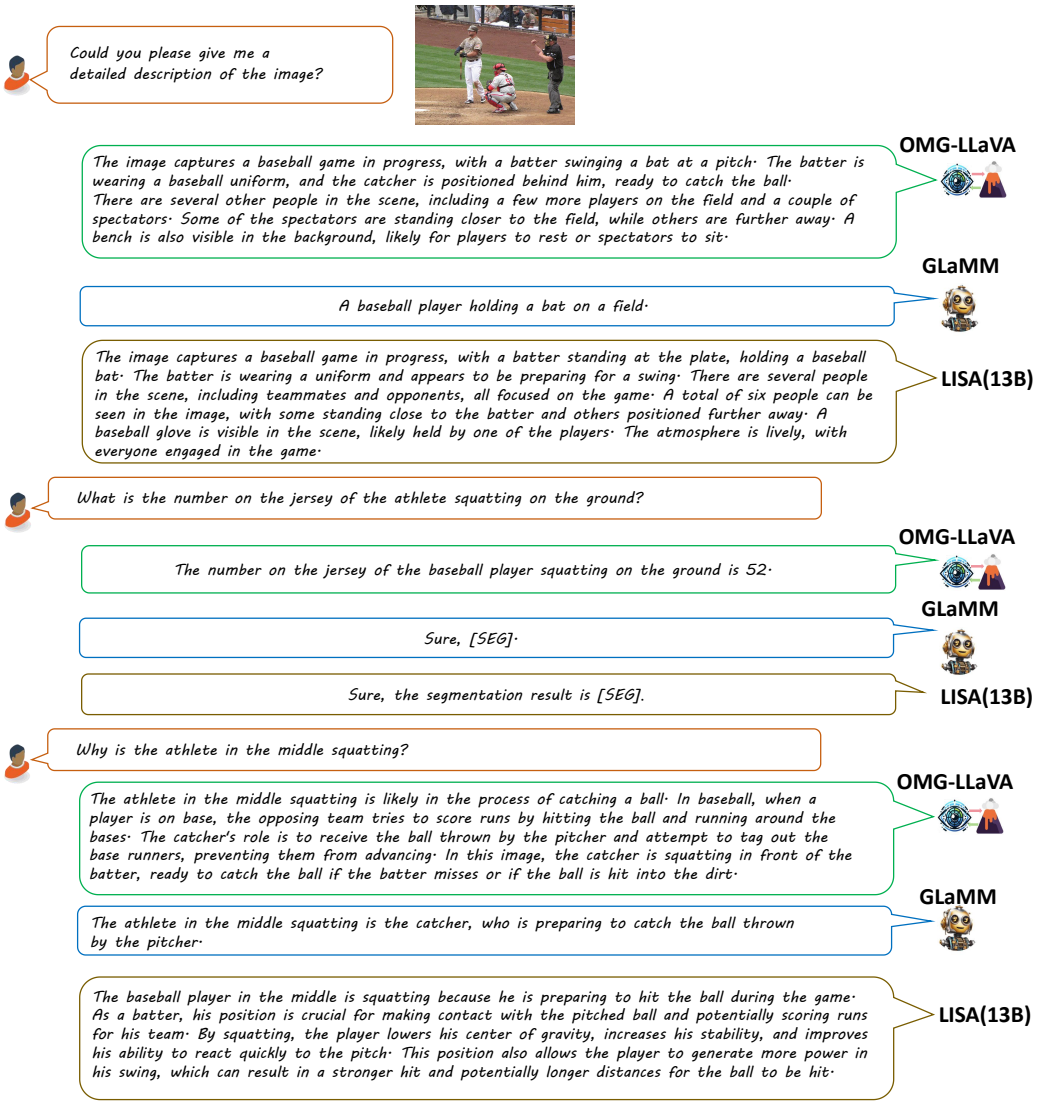
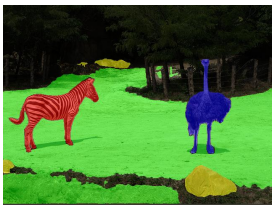


Figure 10: Qualitative comparison on the image-based conversation task.

Figure 11: More visualization results of referring expression segmentation.

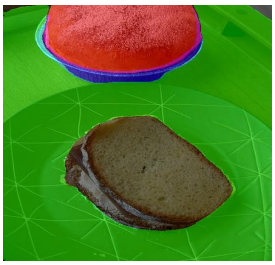
Figure 12: Visualization results of reasoning segmentation.



The image shows a zebra standing on dirt. A bird is also present on the dirt. The dirt is attached to a rock.

A skateboarder is grinding his board on a ramp

The image shows a bench situated on snow. The bench is located in front of a tree. The sky is visible over the tree.



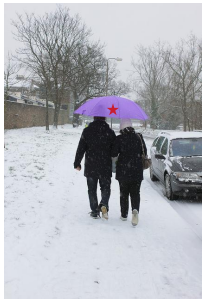
The image shows a cake placed on a dining table. The cake is attached to a bowl.

a giraffe standing behind another giraffe. a giraffe standing in front of another giraffe

A boy in a blue uniform kicks a soccer ball

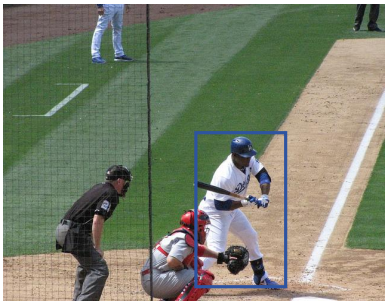
Figure 13: More visualization results of grounded conversation generation.

Point-prompt



The purple umbrella in the image is a large umbrella, providing protection from the falling snow. It is held by the man, who is walking alongside the woman. The umbrella is likely waterproof, allowing them to stay dry while they walk through the snow.

Box-prompt



The athlete wearing a blue helmet is the batter in the baseball game. He is in the process of swinging the bat, attempting to hit the ball that is coming towards him. The batter is likely focused on the ball's trajectory and timing his swing to make contact with it.

Mask-prompt

The sheep in the image is a large, wooly animal with a thick coat of wool. It is lying down on the grass, appearing to be relaxed and comfortable. The sheep is being sheared by a man, who is carefully removing the wool from its body.

Figure 14: More visualization results of grounded conversation generation.