Osprey: Pixel Understanding with Visual Instruction Tuning

Yuqian Yuan[1*], Wentong Li[1*], Jian Liu[2], Dongqi Tang[2], Xinjie Luo[1], Chi Qin[3], Lei Zhang[4], Jianke Zhu[1†]

[1]Zhejiang University [2]Ant Group [3]Microsoft [4]The HongKong Polytechnical University

Abstract

*Multimodal large language models (MLLMs) have recently achieved impressive general-purpose vision-language capabilities through visual instruction tuning. However, current MLLMs primarily focus on image-level or box-level understanding, falling short in achieving fine-grained vision-language alignment at pixel level. Besides, the lack of mask-based instruction data limits their advancements. In this paper, we propose **Osprey**, a mask-text instruction tuning approach, to extend MLLMs by incorporating fine-grained mask regions into language instruction, aiming at achieving pixel-wise visual understanding. To achieve this goal, we first meticulously curate a mask-based region-text dataset with 724K samples, and then design a vision-language model by injecting pixel-level representation into LLM. Specifically, Osprey adopts a convolutional CLIP backbone as the vision encoder and employs a mask-aware visual extractor to extract precise visual mask features from high resolution input. Experimental results demonstrate Osprey's superiority in various region understanding tasks, showcasing its new capability for pixel-level instruction tuning. In particular, Osprey can be integrated with Segment Anything Model (SAM) seamlessly to obtain multi-granularity semantics. The source code, dataset and demo can be found at https://github.com/CircleRadon/Osprey.*

1. Introduction

Multimodal large language models (MLLMs) [24] are key building blocks towards general-purpose visual assistants [23], and they have become increasingly popular in the research community. Though many recent MLLMs such as LLaVA [33], MiniGPT-4 [55], Otter [22], InstructBLIP [12], Qwen-VL [2] and LLaVA-1.5 [32] have demonstrated impressive results on instruction-following and visual reasoning capabilities, they mostly perform vision-language alignment on image-level using image-text pairs. The lack of region-level alignment hinders them from fine-grained image understanding tasks, such as region classification, captioning and reasoning.

To enable region-level understanding in vision-language models, some recent works, *e.g.*, Kosmos-2 [37], Shikra [5], PVIT [4] and GPT4RoI [53], have attempted to process bounding box-specified regions and leverage visual instruction tuning with object-level spatial features. However, directly employing the sparse bounding box as the referring input region could involve irrelevant background features and may lead to inexact region-text pair alignment for visual instruction tuning on LLM. During inference, the box-level referring input may not be able to precisely indicate the object, resulting in semantic deviation, as illustrated in Fig. 1-(a). Besides, these models employ a relatively low input image resolution (*e.g.*, 224×224), and struggle with understanding the details of dense object regions where a much higher resolution is required for optimal performance.

Compared with coarse bounding box, using fine-grained mask as the referring input can represent objects precisely. By training with billions of high-quality masks, the recently developed SAM [19] supports using simple bounding boxes or points as prompts while demonstrating exceptional segmentation quality on zero-shot object, part or subpart. Several studies, like HQ-SAM [18], further enhance SAM's capability on fine-grained segmentation and generalization, making the segmentation more practical for real-world applications. However, these models cannot provide the primary semantic labels, let alone detailed semantic attributes and captions. As a result, the existing methods are limited in understanding the real-world scenes with inherent fine-grained multimodal information.

In this paper, we propose **Osprey**, a novel approach designed to extend the capability of MLLMs for fine-grained pixel-wise understanding. To this end, we present a mask-aware visual extractor to capture precise visual mask features with various granularity. These visual features are then interleaved with language instructions to form the input sequence to LLM. To facilitate the use of high resolution input, we leverage the convolutional CLIP backbone [40]

*Equal contribution.
†Corresponding author.

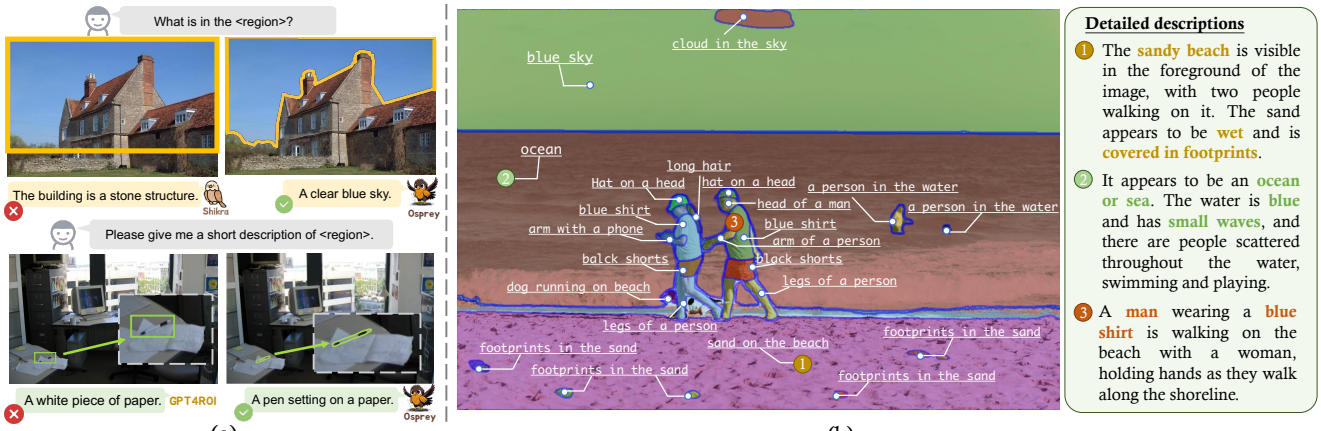

Figure 1. (a) Comparisons between our mask-level Osprey and box-level understanding approaches, *e.g.*, Shikra [5] and GPT4RoI [53]. Osprey can achieve accurate fine-grained region understanding. (b) An example of feeding Osprey with class-agnostic masks from off-the-shelf SAM [19]. One can see that Osprey enables the generation of semantic captions and detailed descriptions of the given image using different prompts.

as the vision encoder. Compared to ViT-based model, convolutional CLIP generalizes well to larger input resolution with efficiency and robustness. With the above designs, Osprey is capable of achieving fine-grained semantic understanding for part-level and object-level regions, providing primary object category, detailed object attributes, and more complex scene descriptions.

To obtain fine-grained pixel-level alignment between vision and language features, we meticulously curate a large-scale mask-based region-text dataset, namely **Osprey-724K**, where the mask and text description of each region are carefully annotated. The majority of data are crafted from publicly available datasets with thoughtfully designed prompt templates to make them instruction-following, including object-level and part-level samples. It includes not only detailed descriptions and conversations but also enriched attributes information. Moreover, we empirically introduce spatial-aware and class-aware negative data mining and short-form response instructions, which further enhances the robustness and flexibility of Osprey's response.

By taking advantage of visual instruction tuning, our proposed model enables new capabilities beyond box-level and image-level understanding. As shown in Fig. 1-(b), Osprey can generate fine-grained semantics based on the class-agnostic masks from the off-the-shelf SAM [19]. Extensive experimental results on open-vocabulary recognition, referring object classification, referring description&reasoning and object hallucination tasks demonstrate the superiority of our approach. The contributions of this work can be summarized as follows.

- We propose a novel approach, namely Osprey, to enable MLLM the pixel-level instruction tuning capability for fine-grained and open-world visual understanding.
- We construct a large-scale instruction tuning dataset with mask-text pairs, called Osprey-724K, which contains object-level, part-level and additional instruction samples for robustness and flexibility.
- Our method, as a fine-grained visual understanding approach, outperforms the previous state-of-the-art methods on a wide range of region understanding tasks.

2. Related Work

Multimodal Large Language Models. Large language models (LLMs), such as GPT-3 [3], Flan-T5 [9], PaLM [8] and LLaMA [45], have significantly advanced the research on Natural Language Processing (NLP). Such progresses have consequently facilitated the development of multimodal language models by expanding the training data and enlarging the model size. This scale-up has led to the breakthrough application of ChatGPT [36]. The great successes of LLMs and MLLMs have also inspired the research on computer vision, enabling multimodal in-context learning [1, 26]. Recent studies have been increasingly concentrated on how to leverage pre-trained LLMs for visual instruction tuning. Prominent examples include LLaVA [33], MiniGPT-4 [55], mPLUG-Owl [48], Otter [22], InstructBLIP [12], Qwen-VL [2] and LLaVA-1.5 [32], *etc*. The common architecture among these models involves a pre-trained visual backbone to encode visual input, an LLM to understand user instructions and generate responses, and a vision-language cross-modal connector to align the output of vision encoder with the language model. While having demonstrated promising capabilities in the image-level multimodal tasks, these models show limited performance when specific regions are required as reference.

Region-level Image Understanding. In the context of region-level image understanding, potential regions of interest are first located before delving into the visual under-

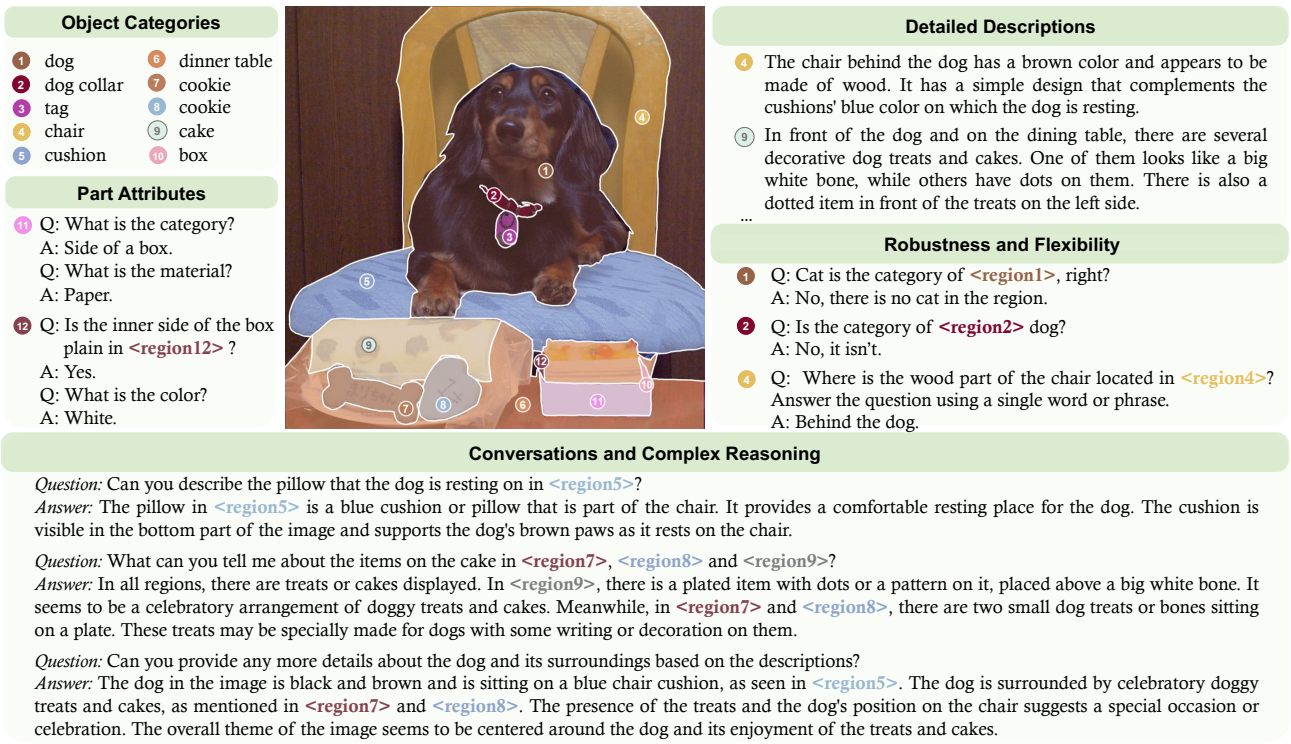

Figure 2. Example sample of the Osprey-724K dataset to illustrate the mask-based instruction-following data.

standing [27, 28, 38, 39]. The Segment Anything Model (SAM) [19], which was trained with billions of high-quality masks, has demonstrated exceptional zero-shot object/part/subpart segmentation quality with simple bounding boxes and points as prompts. As the vanilla SAM cannot provide semantic labels, various approaches, like SEEM [56], HIPIE [46] and Semantic SAM [25], extend the model to predict the semantic category for mask recognition. The primary semantic label only, however, is often insufficient for real-world applications. Therefore, it becomes imperative to incorporate additional semantics such as color, location, and even general descriptions for scene understanding and reasoning. Besides, though some works [21, 44] can achieve pixel-level grounding, they cannot provide the region-based descriptions.

Recent studies such as GPT4RoI [53], PVIT [4], Kosmos-2 [37], Shikra [5], Ferret [49] and GLaMM [42] have enabled MLLMs to achieve region-based image understanding. However, most of these methods employ the bounding box as the referring region, which could involve irrelevant image features from background and introduce inexact region-text pair alignment for visual instructions tuning on LLM. Moreover, these models only allow a small input image size, *e.g.*, 224×224, which may encounter difficulties in analyzing the details of dense object regions. To address these issues, in this work we introduce a pixel-level understanding method based on LLM. Our method supports the use of input masks for region referring and accommodates larger image resolution. Additionally, we curate a comprehensive dataset comprising mask-text pairs to facilitate instruction-based learning for this task.

3. Osprey-724K Dataset

In this section, we present Osprey-724K, an instruction dataset with mask-text pairs, containing around 724K multimodal dialogues to encourage MLLMs for fine-grained pixel-level image understanding. Specifically, Osprey-724K consists of *object-level* and *part-level* mask-text instruction data, which are created based on the publicly available datasets. To make the data instruction-following, we leverage GPT-4 to generate the high-quality mask-text pairs using carefully designed prompt templates. Additionally, to enhance the robustness and flexibility of the response, we introduce the negative sample mining method with short-form response formatting prompt. An example sample of Osprey-724K is shown in Fig. 2, and the detailed statistics and distributions of our Osprey-724K dataset are illustrated in Table 1 and Fig. 3, respectively.

3.1. Object-level Instructions

For an image with N object regions, we make full use of its image-level and object-level captions based on the publicly datasets with mask annotations, such as COCO [31], RefCOCO [50], RefCOCO+ [50] and RefCOCOg [35]. However, these captions are plain and short with few semantic context, which are insufficient to train an MLLM.

Type	Form	Raw Data	GPT-4	#Samples
Object-level	Descriptions	COCO/RefCOCO/RefCOCO+/	✓	70K
	Conversations	RefCOCOg/LLaVA-115K	✓	127K
Part-level	Categories	PACO-LVIS	✓	99K
	Attributes		✓	207K
Robustness	Positive/Negative	COCO/RefCOCO/RefCOCO+/	✗	64K/64K
&Flexibility	Short-Form	RefCOCOg/LLaVA-115K/LVIS	✓	99k

Table 1. Data statistics of Osprey-724K.

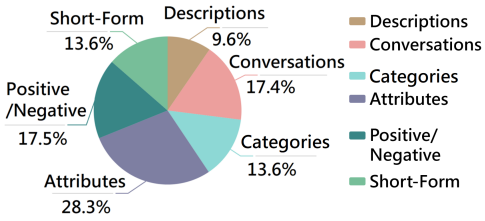

Figure 3. Data distribution of Osprey-724K.

To mitigate this issue, we curate a data processing pipeline to generate fine-grained region-based instruction data, including the object category, object type, object action, location, color, status, *etc*. Firstly, we employ the detailed description in LLaVA-115K [33] as the image-level description for the COCO images. Secondly, we leverage the language-only GPT-4 to create instruction-following data to generate the visual content of each object region with diversity. Specifically, we make full use of the bounding boxes and brief region captions, where each box encodes the object concept and its spatial location in the scene. The short captions collected from RefCOCO [50], RefCOCO+ [50] and RefCOCOg [35] typically describe the specific regions from various perspectives. Based on these information, we employ GPT-4 to generate two types of data, *i.e*., region level *Detailed Description* and *Conversation* samples. Please refer to the *Supplementary Material* for the detailed prompts for GPT-4. Finally, we collect 197K unique object-level mask-region instruction-following samples in total.

3.2. Part-level Instructions

To capture the part-level knowledge, we leverage the PACO-LVIS [41] dataset, which encompasses 456 object-specific part classes distributed among 75 object categories. In specific, PACO-LVIS comprises 55 different attributes, including 29 colors, 10 patterns&markings, 13 materials and 3 levels of reflectance. By taking consideration of these information, we employ GPT-4 to construct the instruction-following data via a question-and-answer (QA) formatting dialogue. Please refer to the *Supplementary Material* for detailed prompts. This straightforward approach enhances the diversity in part categories and attributes. In total, we obtain 306K part mask-region instruction-following samples.

3.3. Robustness and Flexibility

Robustness. Previous studies have shown that MLLMs suffer from the object hallucination issue [29]. That is, objects that frequently appear in visual instructions or co-occur with other objects are susceptible to being erroneously hallucinated. To bolster the robustness of MLLM for accurate region understanding, we further construct positive/negative instruction samples. In specific, we formulate queries to inquire whether a given region belongs to a particular category, and anticipate responses with "`Yes/No`". The positive/negative samples are devised equally to ensure balance.

Negative sample mining intends to find spatial-aware and class-aware negative samples. The former enables the model to identify object-specific categories spatially nearest to a given object. For the latter, negative categories are selected based on high semantic similarities to the target class name, where SentenceBert [43] is employed to calculate the semantic similarity. Empirically, one category is randomly chosen from the top-8 semantically similar candidates to enhance diversity of the negative categories. We apply this scheme to LVIS [15], a large-vocabulary dataset containing around 1,200 object categories with mask annotations.

Flexibility. To improve the response flexibility of MLLMs based on user's instructions, we add the short-form response instructions, covering categories, colors, types, locations or quantities of a specific object region. We employ GPT-4 to generate the instruction samples using the same publicly available datasets as discussed in Sec. 3.1, expecting that GPT-4 can produce a concise response consisting of a single word or phrase. However, we observe that conventional dialogue-based prompts do not explicitly indicate the desirable output format, potentially resulting in the overfitting of an LLM to short-form answers. This issue has been acknowledged in previous works [12, 32] on image-level understanding. To tackle this challenge, we adopt to append the short-form response prompt explicitly at the end of questions when soliciting brief answers.

4. Method of Osprey

4.1. Model Architecture

The architecture overview of Osprey is shown in Fig. 4. Osprey consists of an image-level vision encoder, a pixel-level mask-aware visual extractor and a large language model (LLM). Given an image, the referring mask regions and the input language, we perform tokenization and conversion to obtain embeddings. The interleaved mask features and language embedding sequences are then sent to the LLM to obtain the fine-grained semantic understandings.

4.1.1 Convolutional CLIP Vision Encoder

The vision encoder in the majority of MLLMs [5, 33, 49, 53, 55] is exemplified with the ViT-based CLIP model [14, 40],

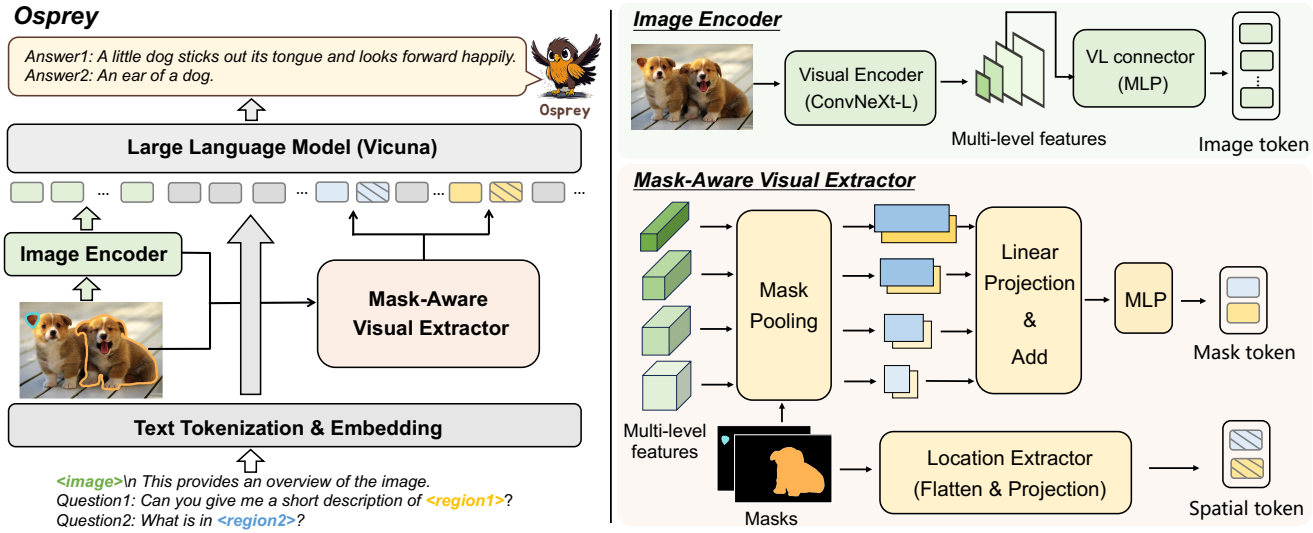

Figure 4. **Overview of Osprey.** The left shows the overall model architecture and the right illustrates the detailed image encoder and mask-aware visual extractor. With the input image, referring mask regions and input language, the corresponding tokenization can be carried out. The interleaved mask features and language embedding sequence are then transmitted to a large language model (LLM) to achieve the nuanced semantic understanding.

which adopts an image resolution of 224×224 or 336×336. However, such a resolution makes it difficult to achieve fine-grained image understanding with pixel-level representations, especially in small regions. Increasing the input image resolution is hindered by the computational burden associated with the global attention in ViT architecture.

To alleviate the above issue, we introduce the convolutional CLIP model, *e.g.*, ResNet [17] and ConvNeXt [34], as the vision encoder. The CNN-based convolutional CLIP has empirically demonstrated promising generalization capabilities across various input resolutions compared to ViT-based CLIP model, for example, in the open-vocabulary segmentation tasks [51]. Such a design allows for efficient training and fast inference without sacrificing performance. Additionally, multi-scale feature maps generated by the CNN-based CLIP vision encoder can be directly utilized for the subsequent feature extraction on each object region. In our implementation, we choose the ConvNeXt-Large CLIP model as the vision encoder and adopt the output at "res4" stage as the image-level features.

4.1.2 Mask-Aware Visual Extractor

In contrast to previous region-based approaches [4, 5, 37, 42, 53] using sparse bounding boxes as the referring input, Osprey adopts the fine-grained representations using detailed mask regions. To capture pixel-level features of each object region, we propose a Mask-Aware Visual Extractor, which not only encodes the mask-level visual features but also gathers the spatial position information of each region $\mathbf{R}_i$. To this end, we first adopt the mask-pooling operation $\mathcal{MP}$ [47] based on multi-level image features $\mathbf{Z}(x)$ from the output of the vision encoder $\mathbf{Z}$. For each single-level feature $\mathbf{Z}(x)_j$, we pool all the features that fall inside the mask region $\mathbf{R}_i$ as follows:

$$V_{ij} = \mathcal{MP}(\mathbf{R}_i, \mathbf{Z}(x)_j). \tag{1}$$

Then, to encode the features across multiple levels, we pass each feature V_{ij} through a linear projection layer $\mathbf{P}_j$ to generate the region-level embeddings with the same dimension, and perform summation to fuse multi-level features. We further employ an MLP layer σ to adapt and produce the visual mask token t_i as follows:

$$t_i = \sigma(\sum_{j=1}^{4} \mathbf{P}_j(V_{ij})). \tag{2}$$

To preserve the spatial geometry of the object region, we utilize the binary mask $\mathbf{M}^{H\times W} \in \{0, 1\}$ for each object region to encode the pixel-level position relationship. We first resize each $\mathbf{M}_i$ to 224×224, and then flatten and project it to generate the spatial token s_i. Finally, we incorporate the visual mask token and its corresponding spatial token as the embeddings for each mask region.

4.1.3 Tokenization for LLM Model

As illustrated in Fig. 4, we feed the image into a pre-trained visual encoder, ConvNeXt-Large CLIP model, to extract the image-level embeddings. For textual information, we tokenize the text sequence using the pre-trained LLM's tokenizer and project them into text embeddings. As for mask-based region, we define a special token as a

placeholder <region>, which is substituted with the mask token t along with spatial token s, denoted by <mask> <position>. When referring to an object region in the text input, the <region> is appended after its region name, like "region1" or "region2". In this way, the mask regions can be well mixed with texts to form complete sentences with the same tokenization space.

In addition to the user instructions, we incorporate a prefix prompt: "<image>\n This provides an overview of the picture." The <image> is a special token that acts as a placeholder, which would be replaced by the image-level embedding from the vision encoder. All of image-level and region-level visual tokens and text tokens are interleaved and fed into LLM to comprehend the image and user instructions with different object regions. We employ Vicuna [7], which is a decoder-only LLM instruction-tuned on top of LLaMA [45], as our LLM.

4.2. Training

The training process of our Osprey model consists of three stages, which are all supervised by minimizing a next-token prediction loss [33, 53, 55].

Stage 1: Image-Text Alignment Pre-training. With the use of convolutional CLIP vision encoder, *i.e.*, ConvNeXt-Large, we first train the image-level feature and language connector for image-text feature alignment. At this stage, Osprey includes a pre-trained vision encoder, a pretrained LLM and an image-level projector. Following LLaVA-1.5 [32], we adopt an MLP as the vision-language connector to improve the multimodal capabilities of the model. The filtered CC3M data introduced in LLaVA [32] are employed as the training data, and only the image-level projector is trained at this stage. The vision encoder and LLM are frozen.

Stage 2: Mask-Text Alignment Pre-training. At this stage, we load the weights trained in Stage 1, and add the Mask-Aware Visual Extractor introduced in Sec. 4.1.2 to capture pixel-level region features. Only the Mask-Aware Visual Extractor is trained in this stage to align mask-based region features with language embeddings. We collect short text and pixel-level mask pairs from the publicly available object-level datasets (COCO [31], RefCOCO [50], RefCOCO+ [50]) and part-level datasets (Pascal Part [6], Part Imagenet [16]), then transform them into instruction-following data to train the model.

Stage 3: End-to-End Fine-tuning. At this stage, we keep the vision encoder weights fixed and finetune the image-level projector, mask-based region feature extractor and LLM model of Osprey. We focus on extending the capability of Osprey to accurately follow user instructions and tackle complex pixel-level region understanding tasks. At this stage, we utilize our curated Osprey-724K dataset. Besides, Visual Genome (VG) [20] and Visual Commonsense Reasoning (VCR) [52] datasets are employed to add more multiple region understanding data. The bounding box annotations are available in VG, while mask-based ones are not. Hence, we employ HQ-SAM [18] to generate high-quality masks with the corresponding box prompts for the VG dataset. After this stage, Osprey is capable of understanding the complex scenarios based on the user instructions and pixel-level mask regions.

5. Experiments

5.1. Experimental Results

To evaluate the effectiveness of our proposed Osprey, we conduct experiments to demonstrate its capabilities of pixel-level region-based recognition, classification, and complex description&reasoning across various representative tasks.

5.1.1 Open-Vocabulary Segmentation

The primary goal of this task is to generate mask-based region recognition with the explicit category [13, 47, 51]. To this end, we utilize a prompt like "Can you give me a short description of <region>? Using a short phrase." The ground-truth (GT) mask regions are adopted for model inference to assess the open-vocabulary recognition performance. Based on the sentence-based response of MLLMs, we calculate the semantic similarity between the output and vocabulary list of each dataset using Sentence-BERT [43]. The category with the highest similarity is chosen as the final result.

Table 2 compares Osprey with state-of-the-art region-based MLLM methods on Cityscapes [11] and ADE20K-150 [54] datasets. Most of these approaches employ the GT bounding box as the input referring region. As Ferret [49] can support free-form input, we adopt the fine-grained mask as its input region to precisely reflect the object. Besides, we leverage the large-scale pretrained vision-language model CLIP [40] with ConvNeXt-L [34] and CLIP-Surgery-ViT-L [30] as vision encoder, and adopt the input mask region and mask-pooling operation [47] to extract visual features for each object. The input image resolution of these CLIP-based methods is set to 512×512, ensuring a fair comparison. On Cityscapes, our Osprey surpasses previous methods by a large margin (*e.g.*, +15.94% PQ, +7.24% AP and +13.05% mIoU against box-level GPT4RoI, +15.07% PQ, +2.23% AP and +11.38% mIoU against mask-level Ferret). On ADE20K-150, Osprey achieves highly competitive performance, obtaining 41.89% PQ, 41.24% AP and 29.63% mIoU, respectively.

5.1.2 Referring Object Classification

In this task, the model needs to classify the object in a specific region of an image. We use two semantic

Method	Type	Cityscapes			ADE20K-150		
		PQ	AP	mIoU	PQ	AP	mIoU
CLIP-ConvNeXt-L [40]	Mask	22.53	12.07	23.06	36.86	39.38	28.74
CLIP-Surgery-ViT-L [30]	Mask	27.24	28.35	21.92	26.55	29.70	21.42
Kosmos-2 [37]	Box	12.09	9.81	13.71	6.53	4.33	5.40
Shikra-7B [5]	Box	17.80	11.53	17.77	27.52	20.35	18.24
GPT4RoI-7B [53]	Box	34.70	21.93	36.73	36.32	26.08	25.82
Ferret-7B [49]	Mask	35.57	26.94	38.40	39.46	29.93	**31.77**
Osprey-7B (Ours)	Mask	**50.64**	**29.17**	**49.78**	**41.89**	**41.24**	29.63

Table 2. Recognition performance on open-vocabulary panoptic segmentation (PQ), instance segmentation (AP) and semantic segmentation (mIoU) upon the validation sets of Cityscapes [11] and ADE20K [54]. The ground truth box/mask is used for performance evaluation.

Method	LVIS		PACO	
	SS	S-IoU	SS	S-IoU
LLaVA-1.5 [32]	48.95	19.81	42.20	14.56
Kosmos-2 [37]	38.95	8.67	32.09	4.79
Shikra-7B [5]	49.65	19.82	43.64	11.42
GPT4RoI-7B [53]	51.32	11.99	48.04	12.08
Ferret-7B [49]	63.78	36.57	58.68	25.96
Osprey-7B (Ours)	**65.24**	**38.19**	**73.06**	**52.72**

Table 3. Semantic similarity and IoU results of referring object classification on *object-level* LVIS and *part-level* PACO. SS/S-IoU denotes Semantic Similarity/IoU, respectively.

relevance metrics, *Semantic Similarity* (SS) and *Semantic IoU* (S-IOU) [10], to evaluate the classification capability of a model. SS measures the similarity of predicted/GT labels in a semantic space, while S-IOU reflects the overlap of words. We conduct experiments on the validation set of object-level LVIS [15] and part-level PACO [41] datasets, and use a prompt like "`What is the category of <region>? Using only one word or phrase.`" Specifically, we randomly sample 1K images with 4,004 objects from LVIS dataset, and sample 1K images with 4,263 objects from PACO dataset for evaluation. We compare our method with image-, box- and mask-level approaches [5, 32, 37, 49, 53], and report the results in Table 3. As for image-level LLaVA-1.5 [32], we adopt the box-based cropped image region as its input. On LVIS [15], which has more than 1,200 object categories, our Osprey obtains 65.24% SS and 38.19% S-IoU, outperforming the state-of-the-art method by 1.46% and 1.62%, respectively. In particular, Osprey significantly outperforms previous MLLMs on PACO, achieving 73.06% SS and 52.72% S-IoU. It surpasses Ferret by 14.38% SS and 26.76% S-IoU, demonstrating its strong fine-grained part-level classification and understanding capability.

5.1.3 Referring Description and Reasoning

Detailed Description. We evaluate the instruction-following detailed description capabilities of each model.

Method	Detailed Description
LLaVA-1.5 [32]	71.11
Kosmos-2 [37]	40.89
Shikra-7B [5]	40.97
GPT4RoI-7B [53]	49.97
Osprey-7B (Ours)	77.54
Osprey-7B* (Ours)	**83.78**

Table 4. Detailed region description performance evaluated by GPT4 on the validation set of RefCOCOs. $*$ denotes the model trained with additional part of data mixture of 665K samples used in LLaVA-1.5 [32] in Stage 3 (the same below).

Ferret-Bench	Osprey-7B*	Ferret-7B	Kosmos-2	Shikra-7B
Referring Description	**72.2**	68.7	51.8	46.0
Referring Reasoning	**67.8**	67.3	33.7	41.6

Table 5. Results on Ferret-Bench [49]. Note that *we use box as the input region* due to lack of mask annotations on Ferret-Bench.

The input prompt for inference is selected randomly from the list in Table A13 of *Supplementary Material*. Motivated by [33], we leverage GPT-4 to comprehensively measure the quality of generated responses from the model to the input referring regions. Specifically, we randomly sample 80 images from the validation set of RefCOCOs [35, 50] for detailed region description. We generate the questions and obtain GPT-4's answers using the instruction generation pipeline outlined in Sec. 3.1. GPT-4 assesses both the precision of referring understanding and the correctness of semantics. The rating score ranges from 1 to 10, with higher scores indicating better performance. To gauge the effectiveness of MLLMs, we calculate the ratio of the predicted answer score to that of GPT-4 and present it as a percentage. As shown in Table 4, Osprey achieves 77.54% accuracy, significantly outperforming GPT4RoI by 27.57%. It is worth mentioning that we adopt the box-cropped region as the image-level input for LLaVA-1.5, which yields an accuracy of 71.11%, more than 6% lower than Osprey. With the additional image-level data used in LLaVA-1.5, Osprey attains 83.78% accuracy and performs the best.

Method	POPE		
	Random	Popular	Adversarial
LLava-1.5 [32]	88.73	85.83	72.10
Shikra-7B [5]	86.90	83.97	83.10
Ferret-7B [49]	**90.24**	84.90	82.36
Osprey-7B* (Ours)	89.47	**87.83**	**85.33**

Table 6. Results on the object hallucination benchmark across three evaluation settings of POPE [29] benchmark.

Method	Cityscapes	ADE	LVIS		PACO	
	PQ	PQ	SS	S-IoU	SS	S-IoU
ViT-L	38.58	38.86	60.89	31.02	70.23	48.57
ConvNeXt-B	48.49	41.94	64.52	37.02	72.86	51.62
ConvNeXt-L	**50.64**	**42.50**	**65.24**	**38.19**	**73.06**	**52.72**

Table 7. Comparisons with various vision encoders on Open-Vocabulary Segmentation and Referring Object Classification.

Ferret-Bench. We further conduct experiments on Ferret-Bench [49] to evaluate the capabilities of both referring description and referring reasoning. Notably, we adopt box as the input region due to the lack of mask annotations on Ferret-Bench. Results are summarized in Table 5. One can see that Osprey-Chat achieves the best performance in both Referring Description and Referring Reasoning tasks with accuracy of 72.2% and 67.8%, outperforming the state-of-the-art method by 3.5% and 0.5%, respectively.

5.1.4 Object Hallucination

As in previous methods [32, 49], we adopt POPE benchmark [29] to evaluate the hallucination of model. As shown in Table 6, we compare our Osprey with state-of-the-art approaches. In Random Sampling setting, Osprey exhibits stellar performance, achieving an accuracy of 89.47, which is close to that of Ferret at 90.24. Under the more challenging Popular and Adversarial Sampling settings, Osprey surpasses previous best methods in accuracy (*e.g.*, 87.83% *vs.* 85.83% with LLaVA-1.5, 85.33% *vs.* 83.10% with Shikra). These encouraging results can be largely attributed to the negative sample mining in Osprey-724K and the fine-grained mask representation.

5.2. Ablation Study

To evaluate the effectiveness of the key elements of our design, we conduct the following ablation experiments.

Different Vision Encoders. To investigate the impact of various CLIP vision encoders on Osprey, including ViT-L, ConvNeXt-B and ConvNeXt-L models, we conduct the experiments on open-vocabulary panoptic segmentation and referring object classification. Table 7 reports the comparison results. When using the ConvNeXt-L model with an input dimension of 512×512, Osprey achieves superior performance. Osprey, equipped with ConvNeXt-B, also delivers very close performance to that of ConvNeXt-L. However, there is a significant decrease in performance when Osprey adopts the ViT-L model with a smaller input size of 224×224 used in LLaVA [33].

Input	#Image Tokens	Speed	SS	S-IoU
224	196	**6.0**	53.20	26.12
336	441	5.8	56.70	28.90
512	1024	3.5	65.24	38.19
800	2500	1.9	**68.29**	**42.66**

Table 8. Comparisons across various input image sizes of ConvNeXt-based CLIP vision encoder on LVIS [15]. Note that *the speed is measured by the number of input mask-text pairs processed per second* during model inference. The evaluation is conducted on a single NVIDIA A100 GPU.

Various Input Image Sizes. We extend to explore the influence of varying input sizes on our ConvNeXt-based CLIP vision encoder in Osprey. Table 8 presents the experimental results on the referring object classification task. The results demonstrate that Osprey exhibits superior performance as the input size increases. Specifically, when the input size is set to 800×800, Osprey attains its peak performance with 68.29% SS and 42.66% S-IoU. However, it is noteworthy that as the input size increases, the number of tokens also rises significantly, adding computational overhead to LLM. With the input size of 800×800, the number of image tokens is 2,500 and 1.9 mask-text pairs are processed per second during inference, representing the slowest speed among the evaluated models. To strike a balance between performance and computational cost, we have opted for a 512×512 input image size in Osprey.

6. Conclusion

In this paper, we presented Osprey, a novel approach to incorporate pixel-level mask region references into language instructions, significantly enhancing MLLMs for fine-grained visual understanding. By incorporating a Mask-Aware Visual Extractor and leveraging a convolutional CLIP backbone, we enabled Osprey the capability of region-based image understanding. To facilitate the fine-grained pixel-level alignment between vision and language, we deliberately curated the Osprey-724K dataset, which comprised 724K mask-based region-text pairs. Trained on the Osprey-724K dataset, our Osprey model demonstrated superior performance on various region understanding tasks, setting new state-of-the-arts. It is expected that our Osprey-724K dataset and Osprey model can facilitate the advancement of MLLMs for visual region understanding in real-world applications.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants (62376244).

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 2

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1, 2

[3] 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*, pages 1877–1901, 2020. 2

[4] 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. 1, 3, 5

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

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *ECCV*, pages 1971–1978, 2014. 6

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org*, 2023. 6

[8] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 2

[9] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 2

[10] Alessandro Conti, Enrico Fini, Massimiliano Mancini, Paolo Rota, Yiming Wang, and Elisa Ricci. Vocabulary-free image classification. In *NeurIPS*, 2023. 7

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *ECCV*, pages 3213–3223, 2016. 6, 7

[12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Hoi Steven. Instructblip: Towards general-purpose visionlanguage models with instruction tuning. In *NeurIPS*, 2023. 1, 2, 4

[13] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary universal image segmentation with maskclip. In *ICML*, 2023. 6

[14] 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. In *ICLR*, 2021. 4

[15] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 4, 7, 8

[16] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *ECCV*, pages 128–145, 2022. 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *ECCV*, pages 770–778, 2016. 5

[18] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv preprint arXiv:2306.01567*, 2023. 1, 6

[19] 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. In *ICCV*, 2023. 1, 2, 3

[20] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2017. 6

[21] 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. 3

[22] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 1, 2

[23] Chunyuan Li, Haotian Liu, Liunian Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, et al. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. In *NeurIPS*, pages 9287–9301, 2022. 1

[24] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal foundation models: From specialists to general-purpose assistants. *arXiv preprint arXiv:2309.10020*, 1, 2023. 1

[25] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023. 3

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

[27] Wentong Li, Yuqian Yuan, Song Wang, Wenyu Liu, Dongqi Tang, Jianke Zhu, Lei Zhang, et al. Label-efficient segmentation via affinity propagation. In *NeurIPS*, 2023. 3

[28] Wentong Li, Wenyu Liu, Jianke Zhu, Miaomiao Cui, Risheng Yu Xiansheng Hua, and Lei Zhang. Box2mask: Box-supervised instance segmentation via level-set evolution. *IEEE TPAMI*, 2024. 3

[29] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 4, 8

[30] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023. 6, 7

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 3, 6

[32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 1, 2, 4, 6, 7, 8

[33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 4, 6, 7, 8

[34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 5, 6

[35] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *ECCV*, pages 11–20, 2016. 3, 4, 7

[36] OpenAI. Chatgpt. https://openai.com/blog/chatgpt, 2022. 2

[37] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 1, 3, 5, 7

[38] Lu Qi, Jason Kuen, Weidong Guo, Jiuxiang Gu, Zhe Lin, Bo Du, Yu Xu, and Ming-Hsuan Yang. Aims: All-inclusive multi-level segmentation. In *NeurIPS*, 2023. 3

[39] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Wenbo Li, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High quality entity segmentation. In *ICCV*, pages 4047–4056, 2023. 3

[40] 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*, pages 8748–8763. PMLR, 2021. 1, 4, 6, 7

[41] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *CVPR*, pages 7141–7151, 2023. 4, 7

[42] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. *arXiv preprint arXiv:2311.03356*, 2023. 3, 5

[43] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019. 4, 6

[44] 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. 3

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 6

[46] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation. In *NeurIPS*, 2023. 3

[47] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, pages 2955–2966, 2023. 5, 6

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

[49] 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. In *ICLR*, 2024. 3, 4, 6, 7, 8

[50] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016. 3, 4, 6, 7

[51] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In *NeurIPS*, 2023. 5, 6

[52] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, pages 6720–6731, 2019. 6

[53] 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. 1, 2, 3, 4, 5, 6, 7

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 6, 7

[55] 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. 1, 2, 4, 6

[56] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *NeurIPS*, 2023. 3