# 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 of achieving fine-grained vision-language alignment at the 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. Especially, 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) [23] are key building blocks towards general-purpose visual assistants [22], and they have become increasingly popular in the research community. Though many recent MLLMs such as LLaVA [30], MiniGPT-4 [55], Otter [21], InstructBLIP [12], Qwen-VL [2] and LLaVA-1.5 [29] having 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], GPT4RoI [53] and GLaMM [42] 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

*Equal contribution.
†Corresponding author.

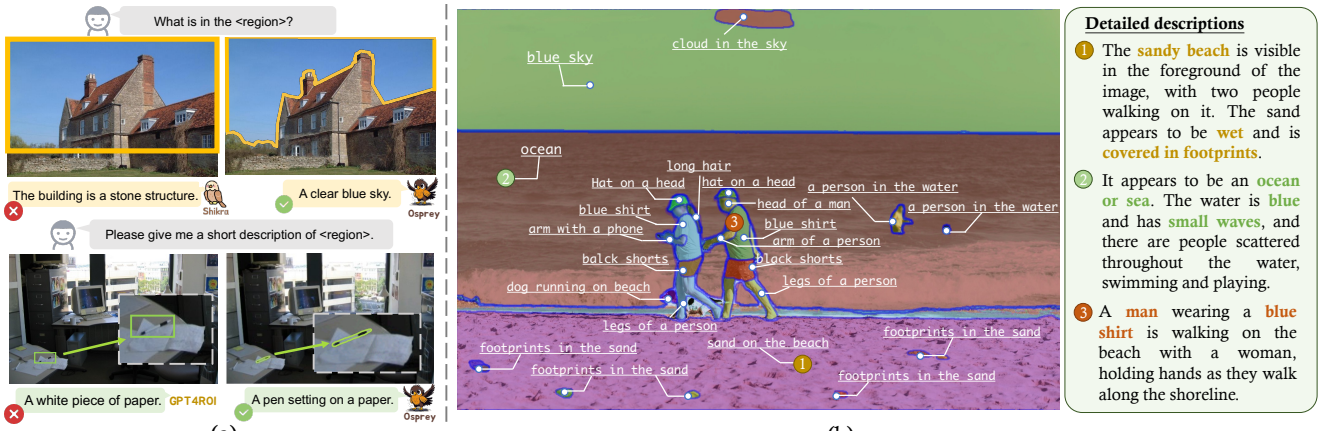



Figure 1. (a) Comparisons between our mask-level Osprey and box-level understanding approaches, *e.g.*, Shikra [5] and GPT4RoI [53]. Our Osprey can achieve accurate fine-grained region understanding. (b) An example of feeding Osprey with class-agnostic masks from off-the-shelf SAM [19]. One can see that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

input, we leverage the convolutional CLIP backbone [40] 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, detailed region description and region level captioning 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 [44], 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, 25]. Recent studies have been increasingly concentrated on how to leverage pre-trained LLMs for visual instruction tuning. Prominent examples include LLaVA [30], MiniGPT-4 [55], mPLUG-Owl [48], Otter [21], InstructBLIP [12], Qwen-VL [2] and LLaVA-1.5 [29], *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.

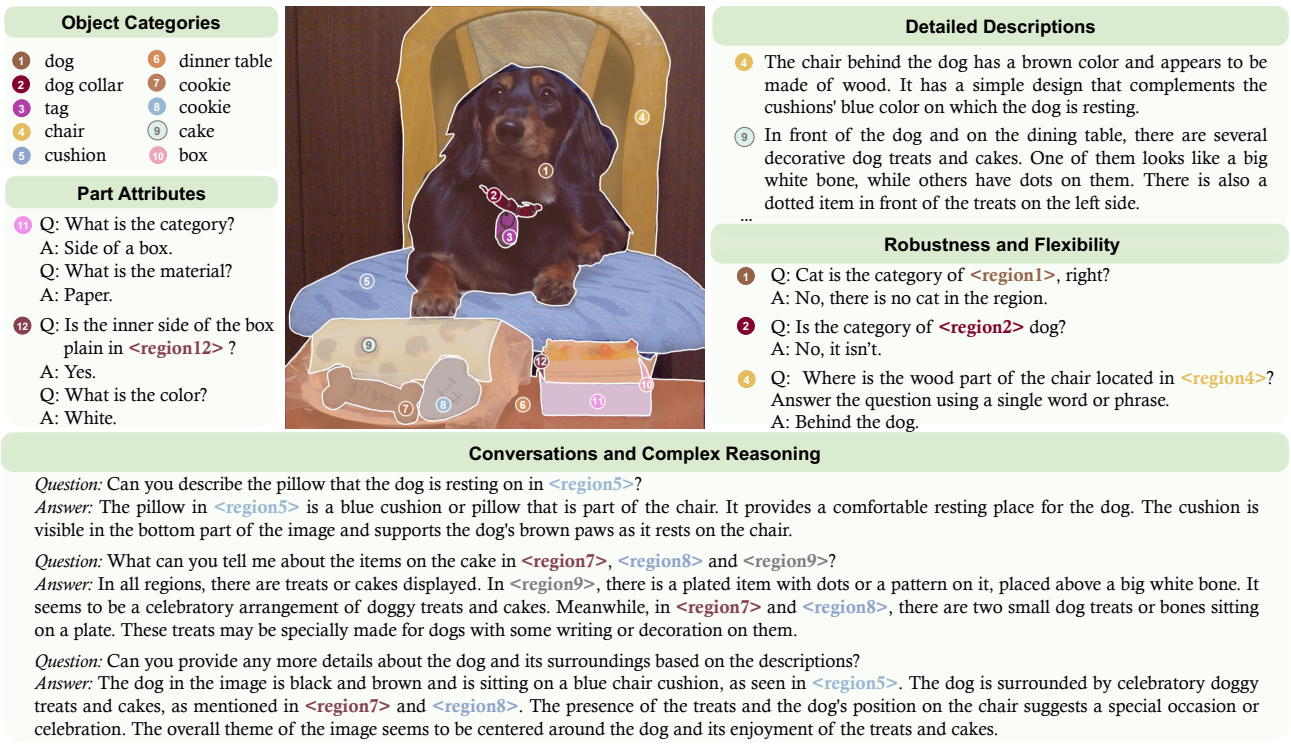



Figure 2. Example sample of the Osprey-724K dataset to illustrate the mask-based instruction-following data.

**Region-level Image Understanding.** In the context of region-level image understanding, potential regions of interest are first located before delving into the visual understanding [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 [45] and Semantic SAM [24], 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.

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 [28], Ref-

| 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.

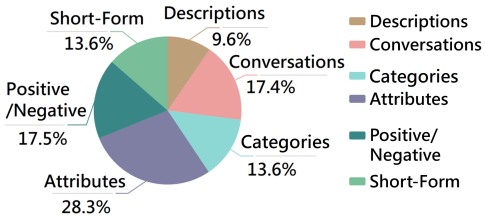



Figure 3. Data distribution of Osprey-724K.

COCO [50], RefCOCO+ [50] and RefCOCOg [34]. However, these captions are plain and short with few semantic context, which are insufficient to train an MLLM.

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 [30] 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 [34] 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 *Appendix* 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 for object regions, 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 for part-level region via a question-and-answer (QA) formatting dialogue. Please refer to the *Appendix* for the detailed prompts. This straightforward approach enhances the diversity in part categories and attributes. In total, we obtain 306K unique part mask-region instruction-following samples.

### 3.3. Robustness and Flexibility

**Robustness.** Previous studies have shown that MLLMs suffer from the object hallucination issue [26]. 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 affirmative or negative 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, 29] 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.

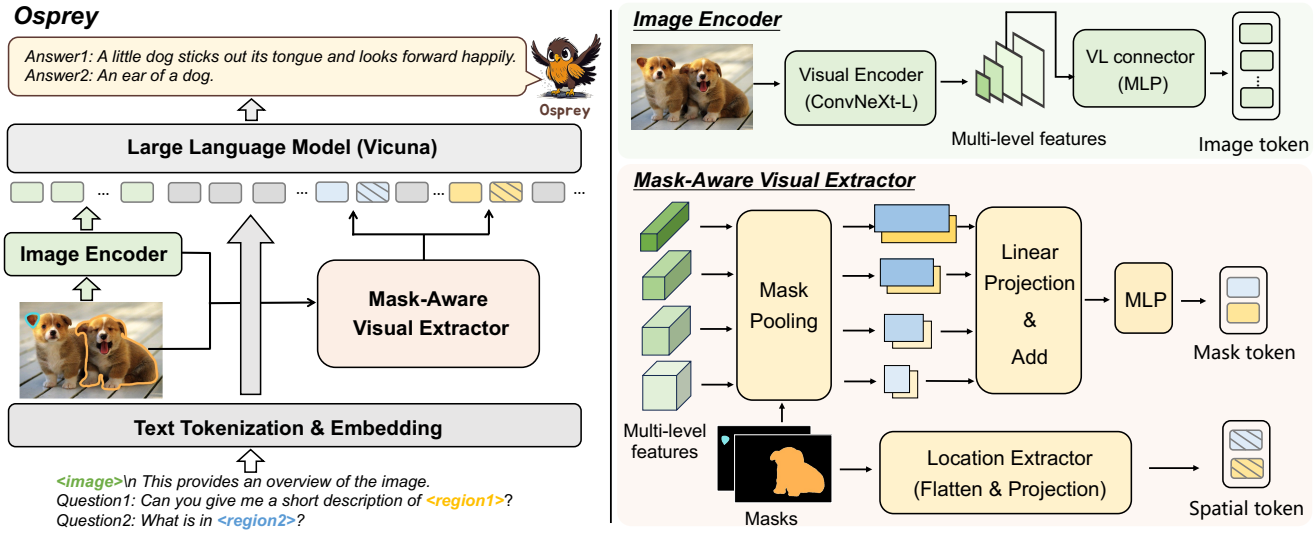



Figure 4. **Overview of Osprey.** The left shows the overall model architecture and the right illustrates the detailed image encoder and mask-aware visual extractor. With the input image, referring mask regions and input language, the corresponding tokenization can be carried out. The interleaved mask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

#### 4.1.1 Convolutional CLIP Vision Encoder

The vision encoder in the majority of MLLMs [5, 30, 49, 53, 55] is exemplified with the ViT-based CLIP model [14, 40], 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 [31], 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 object based 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 $\sigma$ 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 [44], 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 [30, 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 [29], we adopt an MLP as the vision-language connector to improve the multimodal capabilities of the model. The filtered CC3M data introduced in LLaVA [29] 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 [28], 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. Implementation Details

The AdamW [32] is used as the optimizer and the cosine annealing scheduler [33] is used to adjust learning rate. At the first training stage, we set the batch size to 128 and the learning rate to $1\times10^{-3}$ for one epoch. At the second stage, we decrease the learning rate to $2\times10^{-5}$ with a batch size of 4 and train for two epochs. At the final stage, the learning rate is further reduced to $1\times10^{-5}$ with a batch size of 4 for two epochs. The maximum length of sequence in LLM is set to 2,048. All training is conducted on four NVIDIA A100 GPUs with 80GB memory. We leverage the DeepSpeed framework [35] for efficient large-scale model training. The training of the three stages cost 7, 15, and 48 hours, respectively. The input image size is set to 512 $\times$ 512. All the training datasets are aggregated into a single dataloader to ensure the representational integrity. In the training process, the image and its corresponding mask-based instruction/response pairs are randomly selected from each dataset.

### 5.2. 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 descriptions across four representative tasks, including open-vocabulary segmentation, referring object classification, detailed region description and region level captioning. Fig. 5 shows some visual examples to better illustrate the effectiveness of Osprey. In Fig. 6, visual results are showcased based on the mask regions obtained from the off-the-self SAM [19] in "segment everything" mode.

#### 5.2.1 Open-Vocabulary Segmentation

The primary goal of this task is to generate mask-based region recognition and the corresponding 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

| 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 [27] | 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** | **42.50** | 31.72 | 29.94 |

Table 2. Recognition performance on open-vocabulary panoptic segmentation (PQ), instance segmentaion (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 | |
|---|---|---|---|---|
| | Semantic Similarity | Semantic IoU | Semantic Similarity | Semantic IoU |
| LLaVA-1.5 [29] | 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.

corresponding ground-truth 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, including Kosmos-2 [37], Shikra [5], GPT4RoI [53] and Ferret [49], on Cityscapes [11] and ADE20K-150 [54] datasets. Most of these approaches employ the ground truth 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 [31] and CLIP-Surgery-ViT-L [27] 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. The results are also summarized in Table 2. 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 42.50% PQ, 31.72% AP and 29.94% mIoU, respectively. Compared with other approaches, Osprey excels in panoptic segmentation, while lags behind CLIP-ConvNeXt-L on instance segmentation and Ferret on semantic segmentation. These results demonstrate that Osprey can achieve robust recognition and understanding on fine-grained object regions. The encouraging results of Osprey are mainly attributed to its fine-grained mask-based region input and the convolutional backbone (*i.e.* ConvNeXt), which allows larger input image size.

#### 5.2.2 Referring Object Classification

In this task, the model needs to classify the object in a specific region of an image. We use two semantic 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/ground-truth labels in a semantic space, while the S-IOU reflects the overlap of words between the prediction and the ground-truth label. 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 performance evaluation.

We compare our method with image-, box- and mask-level approaches [5, 29, 37, 49, 53], and report the results in Table 3. As for image-level LLaVA-1.5 [29], 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-


Figure 5. Visual examples of Osprey on the input mask-based referring regions.

| Method | Detailed Description |
|---|---|
| LLaVA-1.5 [29] | 71.11 |
| Kosmos-2[37] | 40.89 |
| Shikra-7B [5] | 40.97 |
| GPT4RoI-7B [53] | 49.97 |
| Osprey-7B (Ours) | **77.54** |

Table 4. Detailed region description performance evaluated by GPT4 on the validation set of RefCOCOs.

| Method | Type | METEOR | CIDEr |
|---|---|---|---|
| GRIT [46] | Box | 15.2 | 71.6 |
| Kosmos-2[37] | Box | 14.1 | 62.3 |
| GLaMM [42] | Box | 16.2 | 105.0 |
| Osprey-7B (Ours) | Mask | **16.6** | **108.3** |

Table 5. Region caption performance evaluated on the validation set of RefCOCOg.

IoU. It surpasses previous best Ferret by 14.38% SS and 26.76% S-IoU, demonstrating its strong fine-grained part-level classification and understanding capability.

#### 5.2.3 Detailed Region Description

We evaluate the instruction-following detailed description capabilities of Osprey and other region-level approaches. The input prompt for inference is selected randomly from the list in Table A8 of *Appendix*. Motivated by [30], 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 [34, 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. The results are shown in Table 4. One can see that our Osprey model achieves the best performance with 77.54% accuracy, significantly outperforming region-based 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.

#### 5.2.4 Region Level Captioning

We further provide the quantitative comparisons on region level captioning task with box region-based approaches [37, 42, 46]. Specifically, we fine-tune

| CLIP Vision Encoder | 224 | 448 | 672 | 896 | 1120 |
|---|---|---|---|---|---|
| ViT-Surgery-L [27] | **26.52** | 28.15 | 27.26 | 25.18 | 24.61 |
| ConvNeXt-L [31] | 23.35 | **34.36** | **40.57** | **43.04** | **43.33** |

Table 6. Panoptic segmentation comparisons (PQ) using different vision encoders with different input sizes on ADE20K-150 [54]. The ground truth mask is used for recognition evaluation.

| Vision Encoder | PQ | AP | mIoU |
|---|---|---|---|
| ViT-L | 38.86 | 29.02 | 29.51 |
| ConvNeXt-L | **42.50** | **31.72** | **29.94** |

Table 7. Performance of Osprey with different vision encoders on ADE20K-150.

Osprey-7B on training set of RefCOCOg and employ the prompt like "`Please give me a short description of <region>.`" to prompt our model. The comparison results are shown in Table 5. One can see that our Osprey model exhibits the competitive performance with 16.6% in METEOR score and 108.3% in CIDEr score, thereby surpassing the recent GLaMM approach [42] by 0.4% and 3.3%, respectively. These results highlights the efficacy of Osprey with input referring pixel-level mask regions, demonstrating its superior capability in generating semantically relevant descriptions for object regions.

### 5.3. Ablation Study

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

**ConvNeXt-L *vs.* ViT-L as Vision Encoder.** To investigate the impact of ViT-based and ConvNeXt-based CLIP vision encoders across varying input sizes, we meticulously conduct the experiments on open-vocabulary panoptic segmentation using ViT-Surgery-L [27] and ConvNeXt-L [31] models. All experimental results are obtained by directly employing CLIP as a mask classifier with ground truth masks. Table 6 reports the comparison results. The experimental results reveal that the CNN-based CLIP exhibits superior generalization performance as the input size scales up. Specifically, we observe that the ViT-Surgery-L CLIP model achieves a higher PQ at a lower resolution (*i.e.*, input size 224) while facing challenges at higher resolutions. According to this phenomenon, we adopt a straightforward solution by embracing a CNN-based CLIP as the vision encoder in Osprey.

As for the vision encoder of our Osprey model, we further explore the impacts of ConvNeXt-L and ViT-L CLIP models on open-vocabulary segmentation. Table 7 reports the comparison results. Using ConvNeXt-L with 512×512 input size, Osprey obtains 42.50% PQ, which brings +3.64%PQ (42.50% *vs.* 38.86%) improvement against ViT-L model with 224×224 input size.

| Method | LVIS | | PACO | |
|---|---|---|---|---|
| | Semantic Similarity | Semantic IoU | Semantic Similarity | Semantic IoU |
| w/o Short-form | 56.41 | 25.65 | 50.26 | 23.29 |
| w/o Pos./Neg. | 63.55 | 36.70 | 71.59 | 50.39 |
| Osprey-724K | **65.24** | **38.19** | **73.06** | **52.72** |

Table 8. Performance comparisons with and without short-from prompt and positive/negative samples on *object-level* LVIS [15] and *part-level* PACO [41].

| Input | #Image Tokens | Speed | Semantic Similarity | Semantic 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 9. 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.

**Impacts of Short-form Prompt and Positive/Negative Data.** We conduct experiments to evaluate the impacts of short-form prompt and positive/negative samples on our Osprey-724K dataset. As depicted in Table 8, Osprey trained with both short-form prompt and positive/negative samples attains 65.24% SS and 38.19% S-IoU on the object-level LVIS dataset, bringing an improvement of +8.83% and +12.54% over the model trained without short-form prompt data. On the part-level PACO dataset, the Osprey model trained with only short-form prompt achieves +22.80% SS and +29.43% S-IoU improvements over that without short-form prompt. Regarding the inclusion of positive/negative samples, Osprey model trained with them attains +1.69% SS and +1.49% S-IoU over the model trained without them on object-level LVIS dataset. On Part-level PACO dataset, +1.47% SS and +2.33% S-IoU performance improvements are obtained when positive/negative sample data are used. These experimental results underscore the effectiveness of incorporating short-form prompt and positive/negative data in our Osprey-724K for enhancing model performance.

**Different Input Image Sizes.** We extend to explore the influence of varying input sizes on our ConvNeXt-based CLIP vision encoder in Osprey. Table 9 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


(a) (b) (c)

Figure 6. Visual results of Osprey based on the class-agnostic masks from off-the-self SAM [19]. With the pixel-level mask regions and task-specific prompts, the semantic understanding results are obtained, including (a) open-vocabulary categories, (b) short descriptions, and (c) detailed descriptions. Zoom-in for better view.

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 multimodal large language models (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 image understanding at both part-level and object-level regions. To facilitate the fine-grained pixel-level alignment between vision and language, we deliberately curated the Osprey-724K dataset, which comprised 724K high quality mask-based region-text pairs. Trained on the Osprey-724K dataset, our Osprey model demonstrated superior performance in various region understanding tasks, surpassing state-of-the-art methods. It is expected that our Osprey-724K dataset and Osprey model can facilitate the advancement of MLLMs in pixel-level visual understanding in real-world applications.

## 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, 5, 7, 8, 13, 14

[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 (accessed 14 April 2023)*, 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. 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. *arXiv preprint arXiv:2010.11929*, 2020. 5

[15] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 4, 7, 9

[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. Springer, 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, 6, 10

[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. *International journal of computer vision*, 123:32–73, 2017. 6

[21] 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

[22] 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

[23] 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

[24] 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

[25] 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

[26] 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

[27] 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. 7, 9

[28] 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

[29] 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, 13, 14

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 4, 5, 6, 8, 17

[31] 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, 7, 9

[32] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[34] 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. 4, 8

[35] Microsoft. Deepspeed. https://www.deepspeed.ai/, 2023. 6

[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, 8, 13, 14

[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. 2, 5, 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, 9

[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. 1, 3, 5, 8, 9

[43] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019. 4, 7

[44] 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

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

[46] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. Grit: A generative region-to-text transformer for object understanding. *arXiv preprint arXiv:2212.00280*, 2022. 8

[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, 7

[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. *arXiv preprint arXiv:2310.07704*, 2023. 3, 5, 7

[50] 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. 4, 6, 8

[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, 5, 6, 7, 8, 13, 14

[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. 7, 9

[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, 5, 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

# Appendix

## A. More Qualitative Results

We present additional visual examples to highlight the pixel-level semantic understanding performance with the input referring mask-based regions. Fig. A1 displays more visual cases involving unusual scenes, such as the "catcher's face mask", "bottle cap", "ladder step", and "rim of a plate". Osprey is capable to generate accurate semantic predictions with robust capabilities in these challenging scenarios. Furthermore, Fig. A2 provides comparisons with previous region-level and image-level methods [5, 29, 37, 53]. Our approach exhibits superior scene understanding results with fine-grained details. Please note that each box-cropped region is extracted as the input for image-level LLaVA-1.5 [29].

## B. More Details of Osprey-724K

### B.1. Example Illustrations

We provide several examples to illustrate the instruction-following data in our Osprey-724K dataset, including the object-level and short-form response instruction-following data in Table A1, the part-level instruction-following data in Table A2. Those data are generated through interactions with GPT-4, and the corresponding detailed prompts for GPT-4 are given from Table A4 to Table A7. Besides, Table A3 showcases positive and negative samples in robustness data.

### B.2. Details on Task Prompt

Different prompt templates are used for training Osprey based on various instruction-following data. The question templates are randomly selected from the corresponding lists. Please refer to Table A8∼Table A10 for more details.

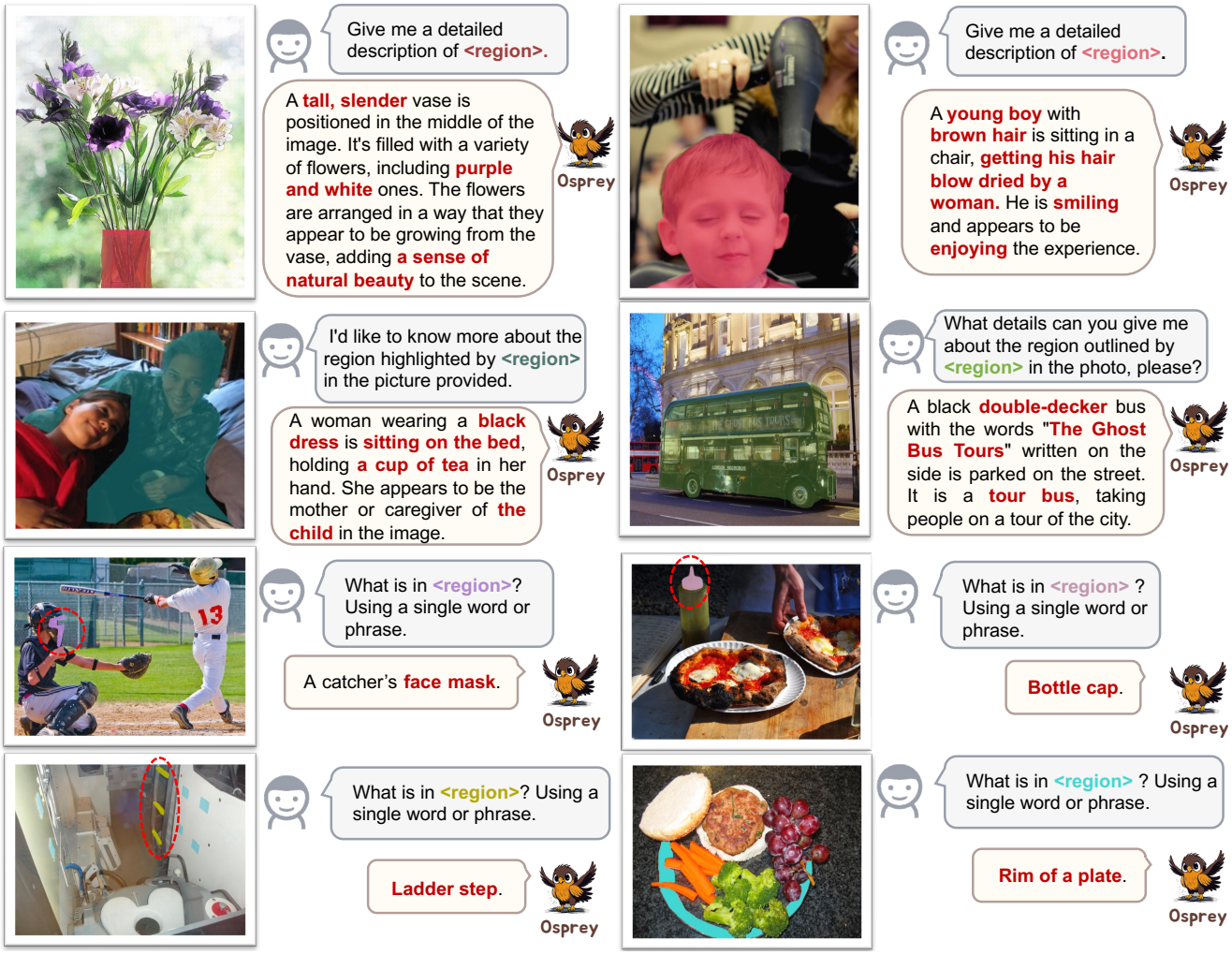



Figure A1. More visual examples of Osprey with object-level and part-level objects.


**Osprey** A young boy **wearing a red helmet** is riding a bicycle **with training wheels**, which is designed for children to learn how to ride a bike. He is **wearing a red shirt** and appears to be riding the bike confidently.

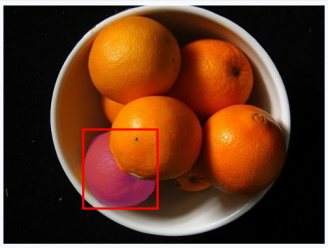
---

**Kosmos** Two young boys on a bike.

**Shikra** A young boy on a bike.

**GPT4RoI** A boy in an orange shirt riding a bike.

**LLaVA-1.5** The image features a young boy sitting on a wooden toy bike, which is designed to look like a real bicycle. The boy is wearing an orange shirt and appears to be enjoying his time on the toy bike. The bike is positioned in the middle of the scene, with the boy sitting comfortably on it.



**Osprey** The **bottom left** orange in the bowl is visible, with its stem pointing towards the left side of the bowl. It is the **closest orange** to the viewer, and is positioned **at the 7 o'clock position**.

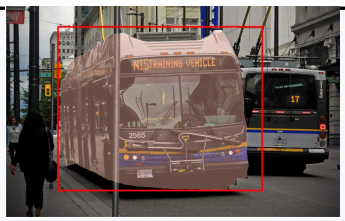
---

**Kosmos** Oranges in a bowl.

**Shikra** Fruit in the bowl.

**GPT4RoI** The orange on the left.

**LLaVA-1.5** The image features a close-up view of an orange, which is the main focus of the scene. The orange is placed in a bowl, and it appears to be a ripe, fresh fruit. The orange is surrounded by a few other oranges, which are located in the background. The bowl containing the oranges is placed on a table, creating a visually appealing and inviting display.




**Osprey** A bus with the **number 2565** on it is parked on the left side of the street, next to a bus with the **number 17** on it. The bus with the number 2565 is the closest to the viewer, and it has **a yellow sign** on the front.

---

**Kosmos** A bus and a truck driving down a street.

**Shikra** Bus with "training vehicle" written on it.

**GPT4RoI** A bus heading to vancouver.

**LLaVA-1.5** The image depicts a city street with a large black and white bus driving down the road. The bus is in motion, and it is passing by a bus stop where a person is waiting. The bus stop is equipped with a bus stop sign, which is visible in the scene. The street is lined with buildings, creating an urban atmosphere. There is also a traffic light present in the scene, indicating that the bus is traveling on a busy street with controlled intersections.

Figure A2. Qualitative comparisons with previous region-level and image-level approaches [5, 29, 37, 53]. The same prompt is adopted to obtain the detailed descriptions, which is selected randomly from Table A8. Our method showcases more accurate region-level semantic understanding with fine-grained details.

**Context type 1: Image-level description**

The image presents a lively market scene with a group of people buying fruits and bags. There are multiple individuals in the market, all browsing through the fresh produce available. A significant variety of fruits are showcased in the market, including bananas, oranges, and apples. Bananas can be seen in several groups, with some green and yellow bananas occupying different areas of the market. Meanwhile, oranges and apples are displayed in smaller sections among the fruits. In addition to fruits, handbags are also being sold at the market, attracting the attention of the customers. Overall, the market bustles with activity as people gather around the fresh fruits and bags, contemplating their purchases.

**Context type 2: Boxes**

person: [0.507,0.409,0.698,0.740], person: [0.243,0.496,0.558,0.746], person: [0.196,0.422,0.395,0.708], orange: [0.761,0.537,0.820,0.569], orange: [0.809,0.553,0.841,0.570], orange: [0.841,0.552,0.868,0.571], banana: [0.671,0.814,0.770,0.887], banana: [0.599,0.703,0.820,0.817], banana: [0.885,0.829,0.941,0.893], apple: [0.811,0.584,0.851,0.603], apple: [0.873, 0.568,0.900,0.586], apple: [0.778,0.580,0.819,0.601], handbag: [0.473,0.110,0.607,0.201], handbag: [0.491,0.202,0.611,0.267], handbag: [0.583,0.105,0.696,0.204].

**Context type 3: Mask region captions**

<region1> (person: [0.507,0.409,0.698,0.740]):
gray shirt wearing glasses. // woman with gray shirt standing next to man. // woman in gray shirt facing camera on right. // the woman in the grey shirt with a watch on her wrist. // a short haired woman in jeans shopping.
<region2> (person: [0.243,0.469,0.558,0.746]):
the lady with the blue shirt. // the back of an older woman with her hair in a barrette with a blue jacket on. // navy blue shirt. // woman back in blue. // a woman is wearing blue sweater.
<region3> (person: [0.196,0.422,0.395,0.708]):
a woman in glasses shops in an open air fruit market. // a woman in a gray coat and scarf.

---

**Response type 1: mask region description**

<region1>: In another part of the market, a short-haired woman in a gray shirt and jeans is shopping. She seems to be looking directly at camera. A noticeable detail is the watch worn on her wrist. This lady, who is also wearing glasses, is standing next to a man.
<region2> : An older woman is visible, her back facing us. She's donning a navy blue sweater. Her hair, neatly fastened with a barrette, adds a note of detail to her appearance.
<region3>: Another woman, clad in a gray coat and scarf, is seen shopping in the open-air fruit market. This woman, similarly wearing glasses, appears to be engrossed in the selection of fresh produce.

**Response type 2: conversation**

Question 1: Can you describe the woman in <region1> and what she is doing?
Answer 1: The woman in <region1> is wearing a gray shirt, jeans, and glasses. She has short hair and a watch on her wrist. She is shopping in the lively market, browsing through the fresh produce, such as fruits and bags.
Question 2: What can you tell me about the woman in <region2>?
Answer 2: The woman in <region2> is wearing a navy blue shirt or sweater. She has her back towards the camera, with her hair styled in a barrette. She appears to be an older woman, also browsing through the market, looking at the various items for sale.
Question 3: How would you compare the person in <region1> and the person in <region2>?
Answer 3: The person in <region2> is positioned on the left side of the image, facing away from the camera. On the other hand, the person in <region1> is located more towards the center-right of the image, facing the camera. Both individuals are wearing different colored shirts, with the person in <region2> wearing a blue shirt and the person in <region1> wearing a gray shirt.
Question 4: Is there anything notable about the person in <region3> ?
Answer 4: The person in <region3> is described as a woman wearing a gray coat and scarf. They are wearing glasses and seem to be shopping in the open-air fruit market, providing an indication of the bustling activity taking place in the scene.
Question 5: Based on the image, how would you describe the overall atmosphere of the market?
Answer 5: The market has a bustling and lively atmosphere. There's a sense of excitement and energy as people gather around the fresh fruits and bags for sale. The variety of items available, as well as the interactions among customers, contribute to the vibrant ambience of the market scene.

**Response type 3: short-form response**

Question 1: What color is the shirt of the woman in <region1>?
Answer 1: Gray.
Question 2: What accessories does the woman in <region1> wear?
Answer 2: Glasses.
Question 3: What color is the woman's sweater in <region2>
Answer 3: Blue.
Question 4: What is the woman in <region2> wearing on her head?
Answer 4: Barrette.
Question 5: How is the woman in <region3> dressed?
Answer 5: In a gray coat and scarf.
Question 6: What type of market is shown in the image?
Answer 6: Fruit market.

Table A1. One example to illustrate the **object-level and short-from response instruction-following data** in Osprey-724K. The top block indicates the contexts such as *image-level descriptions and object boxes* used to prompt GPT-4, and the bottom block shows *three types of responses*. Note that we only show some regions for simple illustration, and the visual image is not used to prompt GPT-4. We only show it here as a reference.

**Context:**
<region1> spoon: dark grey,plain,metal,opaque
<region2> bowl: dark green,plain,ceramic,opaque
<region3> spoon:tip: dark grey,plain,metal,opaque
<region4> spoon:bowl: light grey,plain,metal,opaque
<region5> spoon:neck: dark grey,plain,metal,opaque
<region6> spoon:handle: dark grey,plain,metal,opaque
<region7> bowl:inner body: dark green,plain,ceramic,opaque
<region8> bowl:rim: dark green,plain,ceramic,opaque

**Response type:**
Question: What's in <region1>?
Answer: Spoon.
Question: What is the color of the <region2>?
Answer: Dark green.
Question: Tell me about the color of tip of spoon in <region3>.
Answer: Dark grey.
Question: What's the category of <region4>?
Answer: Bowl of a Spoon.
Question: What is the color of <region5>?
Answer: Dark grey.
Question: Is the handle of spoon plain in <region6>?
Answer: Yes.
Question: What is the reflectance of <region7>?
Answer: Opaque.
Question: What's the material of <region8>?
Answer: Ceramic.

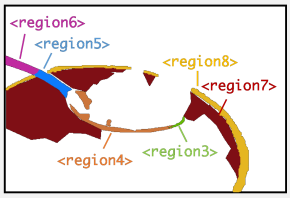




Table A2. One example to illustrate the **part-level instruction-following data** in Osprey-724K. The top block shows *the attributes of each mask region* used to prompt GPT-4, and the bottom block indicates *conversations of the responses*. Note that the visual image and region masks are not used to prompt GPT-4. We only show them here as the reference.

**Example1:**
**groundtruth**:
<region1>: sock
<region2>: sock
Question: Is the category of <region1> calf?
Answer: No, it isn't.
Question: Soccer ball is the category of <region2>, right?
Answer: Not at all.


**Example2:**
**groundtruth**:
<region1>: spoon
<region2>: soup
<region3>: soup
Question: Does this area <region1> belong to category fork?
Answer: This is not so.
Question: The category of <region2> is soup, right?
Answer: Yes, it is.
Question: Does the category salad accurately describe this region <region3>?
Answer: No, there is no salad in the region.

Table A3. Two examples to show **robustness data** in Osprey-724K based on the categories of LVIS. The top block shows negative instruction samples and the bottom block indicates the positive ones.

**messages** = [ {"role":"system", "content": f"""You are an AI visual assistant that can analyze a single image. You receive a detailed description/several descriptions of this image. In addition, most object locations within the image are given, along with detailed coordinates. These coordinates are in the form of bounding boxes, represented as (x1, y1, x2, y2) with floating numbers ranging from 0 to 1.These values correspond to the top left x, top left y, bottom right x, and bottom right y.

Your role is to give a detailed description of each special region in the image. Instead of directly mentioning the bounding box coordinates, utilize this data to explain each region using natural language. Include details like **object category, object type, object color, attributes of the object, object locations, object state and other attributes**.
When using the information from the image and object region captions and coordinates, directly explain the region, and do not mention that the information source is the caption or the bounding box. Always answer as if you are directly looking at each region. Provide a direct answer without mention"this region". The answer template is: '<region1>: ...'"""}
]

```
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']} )
messages.append({"role":"user", "content":'\n'.join(query)})
```

Table A4. The prompt used to generate the **detailed region description** in Osprey-724K. For each query, we show the prompt construction process for ChatGPT/GPT-4 to collect `query['response']` from `query['context']`, using few-shot in-context-learning, where examples are from `fewshot_samples`, each example including input `sample['context']` and output `sample['response']` as in [30]. `messages` is the our final prompt. The prompt templates below also adopt the similar manner. Please see Table A5, Table A6 and Table A7 for the specific details.

**messages** = [ {"role":"system", "content": f"""You are an AI visual assistant, and you are seeing several object regions in a single image. What you see are provided with a detailed description for the whole image and each object region in this image, describing you are looking at. Answer all questions as you are seeing the image. The location of each object region is given in the form of bounding boxes, represented as (x1, y1, x2, y2) with floating numbers ranging from 0 to 1. These values correspond to the top left x, top left y, bottom right x, and bottom right y.
Design a conversation between you and a person asking about each object region of this image. The answers should be in a tone that a visual AI assistant is seeing the image and answering the question. Ask diverse questions and give corresponding answers. All the regions given should be mentioned in the questions, when referring to each region, use <region1>, <region2>, etc. Include questions asking about the visual content of each object region in the image, including the **object category, object type, object color, object actions, object locations, relative positions between objects and other attributes, etc**. Only include questions that have definite answers:
(1) one can see the content in the object region of this image that the question asks about and can answer confidently;
(2) one can determine confidently from the object region of this image that it is not in the image.
Do not ask any question that cannot be answered confidently. Also include complex questions that are relevant to the content of each object region in the image, for example, asking about background knowledge of the objects, asking to discuss about events happening in the image, etc. Again, do not ask about uncertain details.
Provide detailed answers when answering complex questions. For example, give detailed examples or reasoning steps to make the content more convincing and well-organized. You can include multiple paragraphs if necessary."""}
]

```
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']} )
messages.append({"role":"user", "content":'\n'.join(query)})
```

Table A5. The prompt used to generate the **conversations response** data in Osprey-724K.

```
messages = [{"role":"system", "content": f"""You are an AI visual assistant, and you are seeing several object regions in a single image. What you see are provided with a detailed description for the whole image and each object region in this image, describing you are looking at. Answer all questions as you are seeing the image. The location of each object region is given in the form of bounding boxes, represented as (x1, y1, x2, y2) with floating numbers ranging from 0 to 1. These values correspond to the top left x, top left y, bottom right x, and bottom right y.
Design a conversation between you and a person asking about each object region of this image. The answers must be in one word or one phrase. Ask diverse questions and give corresponding answers. All the regions given should be mentioned in the questions, when referring to each region, use <region1>, <region2>, etc. Include questions asking about the visual content of each object region in the image, including the object category, object type, object color, object actions, object locations, relative positions between objects and other attributes, etc. Only include questions that have definite answers:
(1) one can see the content in the object region of this image that the question asks about and can answer confidently;
(2) one can determine confidently from the object region of this image that it is not in the image.
Do not ask any question that cannot be answered confidently. Do not ask any question that is not mentioned. Do not ask any question that cannot be answered with one word or phrase.
Most importantly, the answer must be in one word or short phrase."""}
]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']} )
messages.append({"role":"user", "content":'\n'.join(query)})
```

Table A6. The prompt used to generate the **short-form response** data in Osprey-724K.

```
messages = [{"role":"system", "content": f"""You are an AI visual assistant that can analyze a single image. There are some regions in this image, each region is an object or a part of the object. You receive a short description with some words, separated by commas, for the common attributes of each region, which may contain category name, color, pattern & markings, material and reflectance etc. If a region is a part of an object, the category name is described as "object:part", like "person:body".

According to each description, design a conversation between you and a person asking about each region of this photo. The answers should be in a tone that a visual AI assistant is seeing the image and answering the question. Ask diverse questions and give corresponding answers.
Include diverse questions asking about the attributes of each region including category, part category, color, pattern & markings, material and reflectance. Each region must involve 1-2 questions, when referring to each region, use <region1>, <region2>, etc. Answer the question using as few words as possible (single or two words). Only include questions that have definite answers: one can see the content in the region of this image that the question asks about and can answer confidently.

Do not ask any question that cannot be answered confidently."""}
]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']} )
messages.append({"role":"user", "content":'\n'.join(query)})
```

Table A7. The prompt used to generate the **part-level attributes** instruction data in Osprey-724K.

- "Can you provide me with a detailed description of the region in the picture marked by <region>?"
- "I'm curious about the region represented by <region> in the picture. Could you describe it in detail?"
- "What can you tell me about the region indicated by <region> in the image?"
- "I'd like to know more about the area in the photo labeled <region>. Can you give me a detailed description?"
- "Could you describe the region shown as <region> in the picture in great detail?"
- "What details can you give me about the region outlined by <region> in the photo?"
- "Please provide me with a comprehensive description of the region marked with <region> in the image."
- "Can you give me a detailed account of the region labeled as <region> in the picture?"
- "I'm interested in learning more about the region represented by <region> in the photo. Can you describe it in detail?"
- "What is the region outlined by region in the picture like? Could you give me a detailed description?"
- "Can you provide me with a detailed description of the region in the picture marked by <region>, please?"
- "I'm curious about the region represented by <region> in the picture. Could you describe it in detail, please?"
- "What can you tell me about the region indicated by <region> in the image, exactly?"
- "I'd like to know more about the area in the photo labeled <region>, please. Can you give me a detailed description?"
- "Could you describe the region shown as <region> in the picture in great detail please?"
- "What details can you give me about the region outlined by <region> in the photo, please?"
- "Please provide me with a comprehensive description of the region marked with <region> in the image, please."
- "Can you give me a detailed account of the region labeled as <region> in the picture, please?"
- "I'm interested in learning more about the region represented by <region> in the photo. Can you describe it in detail, please?"
- "What is the region outlined by <region> in the picture like, please? Could you give me a detailed description?"
- "Please describe the region <region> in the image in detail."
- "Can you offer a thorough analysis of the region <region> in the image?"
- "Could you elaborate on the region highlighted by <region> in the picture provided?"
- "Please share more information about the zone emphasized with <region> in the photo."
- "What insights can you give ablout the area denoted by <region> in the image presented?"
- "Can you share a comprehensive rundown of the region denoted by <region> in the presented image?"
- "I'd like to know more about the region highlighted by <region> in the picture provided."
- "Work through the important details of the area <region> in the image."
- "Illustrate the area represtented by <region> through a descriptive explanation."
- "Examine the region <region> closely and share its details."

Table A8. The list of instruction templates for detailed mask-region description used in Osprey.

- "Please give me a short description of region <region>."
- "Can you give me a short description of <region>? "
- "Can you provide me with a short description of the region in the picture marked by <region>?"
- "I'm curious about the region represented by <region> in the picture. Could you describe it in few words?"
- "What can you tell me about the region indicated by <region> in the image in few words?"
- "I'd like to know more about the area in the photo labeled <region>. Can you give me a concise description?"
- "Could you describe the region shown as <region> in the picture concisely?"
- "What can you give me about the region outlined by <region> in the photo?"
- "Please provide me with a brief description of the region marked with <region> in the image."
- "Can you give me a brief introduction of the region labeled as <region> in the picture?"
- "I'm interested in knowing the region represented by <region> in the photo. Can you describe it in several words?"
- "What is the region outlined by <region> in the picture like? Could you give me a streamlined description?"
- "Can you provide me with a brief description of the region in the picture marked by <region> please?"
- "I'm curious about the region represented by <region> in the picture. Could you describe it in few words please?"
- "What can you tell me about the region indicated by <region> in the image?"
- "I'd like to know more about the area in the photo labeled <region> please. Can you give me a simple description?"
- "Could you describe the region shown as <region> in the picture in several words?"
- "What attributes can you give me about the region outlined by <region> in the photo please?"
- "Please provide me with a simple description of the region marked with <region> in the image please."
- "I'm interested in learning more about the region represented by <region> in the photo. Can you describe it in few words please?"
- "What is the region outlined by <region> in the picture like please? Could you give me a simple and clear description?"
- "Please describe the region <region> in the image concisely."
- "Can you offer a simple analysis of the region <region> in the image?"
- "Could tell me something about the region highlighted by <region> in the picture briefly?"
- "Please share some information about the zone emphasized with <region> in the photo."
- "What insights can you give ablout the area denoted by <region> in the image presented?"
- "Can you share a simple rundown of the region denoted by <region> in the presented image?"
- "I'd like to know some arrtributes about the region highlighted by <region> in the picture provided."
- "Work through the important arrtributes of the area <region> in the image."
- "Illustrate the area represtented by <region> with some important arrtributes."

Table A9. The list of instruction templates for brief mask-region description used in Osprey.

- "<category> is the category of <region>, right?"
- "Is the category of <region> <category>?"
- "Dose this area <region> belong to category <category>?"
- "Is <category> the appropriate classification for this area <region>?"
- "Does category <category> accurately describe this region <region>?"
- "The category of <region> is <category>, right?"
- "Is this area <region> classified under category <category>?"
- "Is it correct to say this area <region> falls into category <category>?"
- "Is the classification of this region <region> aligned with category <category>?"

Table A10. The list of instruction templates for the mask-region positive/negative categories used in Osprey.