# InstructPart: Task-Oriented Part Segmentation with Instruction Reasoning

**Zifu Wan  Yaqi Xie  Ce Zhang  Zhiqiu Lin  Zihan Wang**
**Simon Stepputtis  Deva Ramanan  Katia Sycara**

Robotics Institute, Carnegie Mellon University

{zifuw, yaqix, cezhang, zhiqiul, zihanwa3, sstepput, deva, sycara}@andrew.cmu.edu

https://zifuwan.github.io/InstructPart/

## Abstract

Large multimodal foundation models, particularly in the domains of language and vision, have significantly advanced various tasks, including robotics, autonomous driving, information retrieval, and grounding. However, many of these models perceive objects as indivisible, overlooking the components that constitute them. Understanding these components and their associated affordances provides valuable insights into an object's functionality, which is fundamental for performing a wide range of tasks. In this work, we introduce a novel real-world benchmark, *InstructPart*, comprising hand-labeled part segmentation annotations and task-oriented instructions to evaluate the performance of current models in understanding and executing part-level tasks within everyday contexts. Through our experiments, we demonstrate that task-oriented part segmentation remains a challenging problem, even for state-of-the-art Vision-Language Models (VLMs). In addition to our benchmark, we introduce a simple baseline that achieves a twofold performance improvement through fine-tuning with our dataset. With our dataset and benchmark, we aim to facilitate research on task-oriented part segmentation and enhance the applicability of VLMs across various domains, including robotics, virtual reality, information retrieval, and other related fields.

## 1 Introduction

Large Vision-Language Models (LVLMs) (Radford et al., 2021; Alayrac et al., 2022; You et al., 2023) have been extensively utilized across various domains, such as robotics (Driess et al., 2023), autonomous driving (Zhou et al., 2023), medical imaging (Han et al., 2023), and information retrieval (Liu et al., 2021), owing to their strong language reasoning and perceptual capabilities. In these cases, LVLMs are primarily employed for language grounding, enabling the identification of visual targets within a scene based on associated language descriptions. By leveraging large datasets composed of image-text pairs, LVLMs can map visual content to textual semantic representations (Radford et al., 2021) within joint embedding spaces. However, while this approach yields powerful models with strong text-image alignment, they often focus on understanding entire objects (Liu et al., 2023b; Zou et al., 2023b,a; Xu et al., 2023; Liang et al., 2023; Sun et al., 2023), overlooking the fact that grounding is not solely about classifying whole objects but also about recognizing fine-grained parts. As illustrated in Figure 1, given the task of adding water and a visual observation of a kettle, the system must not only identify the entire kettle but also recognize each part of the target and its corresponding affordances before grounding to task-related regions.

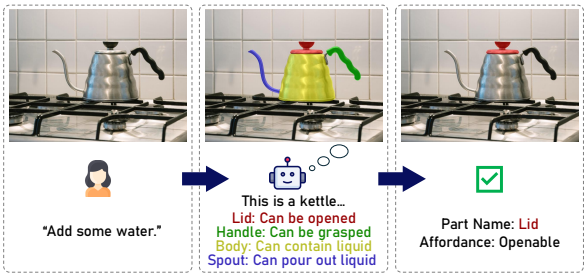


Figure 1: The task-oriented part segmentation task: Presented with an image observation (left) and a corresponding task to add some water, the system is required to reason about specific parts to fulfill the task.

To advance task-oriented part segmentation, we believe that establishing a benchmark is essential for the field. However, most large-scale vision datasets primarily focus on object-level understanding (Liu et al., 2023b; Zou et al., 2023b,a; Xu et al., 2023; Liang et al., 2023; Sun et al., 2023), while existing part-level recognition datasets either cover only a limited range of part categories (Nguyen et al., 2017; Myers et al., 2015; Roy and Todorovic, 2016) or are derived from simulations (Geng

et al., 2023; Deng et al., 2021; Xiang et al., 2020; Mo et al., 2019). We attribute this primarily to the challenge of annotating part-level labels and task-related descriptions, which is both time-consuming and expensive (Wan et al., 2024).

To address this challenge, we introduce a new real-world dataset, **InstructPart**, consisting of 2,400 images across 48 object classes and 44 part classes, with hand-labeled segmentation masks, as well as 9,600 hand-labeled task instructions, 2,400 part queries, and 2,400 affordances. Each image is accompanied by human-annotated and GPT-polished instructions for common household tasks and detailed part segmentation masks. As part of our benchmark, we propose two distinct tasks: a) Task Reasoning Part Segmentation (TRPS): identifying a particular part given an instruction to fulfill a task, e.g., "Locate the part meant for pulling to open the microwave"; and b) Oracle Referring Part Segmentation (ORPS): identifying an object part given a part query, e.g., "handle of the microwave". Thorough evaluations of current vision-language models on the two tasks reveal a significant deficiency in their ability to comprehend natural language and accurately ground it across diverse objects and parts. This finding highlights the need to address a critical shortcoming in vision-language models for fine-grained segmentation.

Finally, we explore the training potential of our dataset by proposing a simple yet effective baseline, which leads to a nearly 100% improvement. With our proposed benchmark, we emphasize the importance of advancing vision-language models to excel not only in object-level understanding but also in discerning fine-grained part-level details. By utilizing our dataset, we hope to envision advancements in robotics, particularly for assistive robots, as well as in manipulation tasks, object segmentation, virtual reality, affordance learning, and other related domains. Our contributions are as follows:

- To the best of our knowledge, we introduce the first dataset that bridges task-oriented interactions with part segmentation for common household tasks.
- We rigorously evaluate various vision-language models on our dataset, revealing their limitations in fine-grained recognition with language reasoning.
- We fine-tune a simple baseline based on a state-of-the-art model, achieving performance gains of over twofold, highlighting the quality and training potential of our dataset.

| Dataset | #Object | #Part | #Affordance | #Action | Instruction |
|---|---|---|---|---|---|
| PartImageNet | 11/158 | 13 | N/A | N/A | ✗ |
| Pascal-Part | 20 | – | N/A | N/A | ✗ |
| PACO | 75 | – | N/A | N/A | ✗ |
| UMD | 17 | N/A | 7 | N/A | ✗ |
| NYUv2-AFF | 40 | N/A | 5 | N/A | ✗ |
| IIT-AFF | 10 | N/A | 9 | N/A | ✗ |
| AGD20K* | 50 | N/A | 36 | N/A | ✗ |
| InstructPart (Ours) | 48 | 44 | 30 | 37 | ✓ |

Table 1: Comparison of relevant part segmentation datasets. We show the number of object classes (#Object), part classes (#Part), affordances (#Affordance), actions (#Action), and whether instructions are included (Instruction). N/A means there is no such type of data, while – means the data exists while no relevant information is provided. 11/158 indicates the super-class and sub-class numbers in PartImageNet. * indicates the dataset only contains point annotations instead of accurate masks for target affordances.

## 2 Related Work

### 2.1 Part Segmentation.

The problem of segmenting an object into a collection of semantic parts is not a novel problem in it itself. Prior works mainly ustilized fully supervised approaches, which need to be trained on large datsets (Sun et al., 2023), such as PartImageNet (He et al., 2022), Pascal-Part (Chen et al., 2014), ADE20K (Zhou et al., 2019), and PACO (Ramanathan et al., 2023). However, these datasets contain only a limited subset relevant to human-robot interaction (e.g., PartImageNet includes just one related category: bottle), thus restricting their applicability to daily tasks. In robotics, part segmentation is used to understand the components of objects and their associated affordances, which are crucial for manipulation tasks (Gadre et al., 2021; Yi et al., 2018). While many datasets have been created for this domain (Mo et al., 2019; Xiang et al., 2020; Deng et al., 2021; Geng et al., 2023), they are all generated from simulators, which introduces potential challenges when generalizing to real-world scenarios. To address this issue, real-world affordance datasets such as UMD-Affordance (Myers et al., 2015), NYUv2-Affordance (Roy and Todorovic, 2016), and IIT-AFF (Nguyen et al., 2017) exist. However, due to the difficulty of collecting large quantities of real-world data, these datasets are limited in the number of affordances they present. On the other hand, AGD20K (Luo

et al., 2022) collects egocentric and exocentric images for affordance learning. However, it only provides sparse point annotations, which can be insufficient for accurate task execution, such as manipulation. Similarly, Where2Act (Mo et al., 2021) extracts actionable information from articulated objects with movable parts but is limited to six action types and a single contact point, which may be sub-optimal. Furthermore, the aforementioned datasets only contain simple word phrases outlining the target; however, full language comprehension is crucial in a human-robot interaction task. Understanding language can be ambiguous even for simple objects like a light switch, which can be "turned on", "pressed" or "twisted" depending on the switch's type, and people tend to refer to such objects as parts of larger task descriptions instead of a single word. Motivated by this, we construct a comprehensive dataset with task descriptions and object-part classes, as shown in Tab. 1.

### 2.2 Open-Vocabulary Segmentation.

Open-vocabulary segmentation aims to perform zero-shot segmentation with the assistance of vision-language foundation models, such as CLIP (Radford et al., 2021). For example, OVSeg (Liang et al., 2023) proposes to crop the region proposals and finetune CLIP using a mask prompt tuning mechanism. SAN (Xu et al., 2023) applies a side adapter network to a frozen CLIP to get the class of masks. Going beyond object-level segmentation, VLPart (Sun et al., 2023) performs open-vocabulary part segmentation by parsing the novel object into parts using its semantic correspondence with the base object and classifies it with CLIP.

Although these open-world recognition methods demonstrate potential in recognizing out-of-distribution classes, they have limited reasoning ability to understand complex instructional sentences, prohibiting their wider usage in daily tasks requiring complex language comprehension.

### 2.3 Referring Expression Segmentation.

Referring expression segmentation aims to generate a segmentation mask from a given language expression (Hu et al., 2016). Popular referring segmentation methods use a visual and a language encoder to extract features from the two modalities respectively, and design attention mechanisms to incorporate the features and assemble classes for region masks (Yang et al., 2022; Liu et al., 2023a; Ouyang et al.; Liu et al., 2023b). Recently, more works have applied pre-trained foundation models, e.g., SAM(Kirillov et al., 2023) and CLIP (Radford et al., 2021) as the encoder and focused on the design of the decoder, such as X-Decoder (Zou et al., 2023a) and SEEM (Zou et al., 2023b). Furthermore, ManipVQA (Huang et al., 2024) applies VLMs with manipulation-centric knowledge to detect tools and affordances. However, the referring expression task only takes short phrases as input and does not consider complex reasoning, for example, when the target name does not appear directly in the expression.

### 2.4 Reasoning Segmentation.

On the other hand, remarkable advances have been made in large language models (LLMs), which can understand complex language inputs and have the potential for more complex referring segmentation. Models such as BLIP-2 (Li et al., 2023b), LLaVA-1.5 (Liu et al., 2023c), MiniGPT-4 (Zhu et al., 2023), Flamingo (Alayrac et al., 2022), and GPT-4V (Yang et al., 2023b) have explored the design of multi-modal LLMs for visual understanding and demonstrate their ability through tasks such as image captioning, visual question answering (VQA), etc. To enable the grounding ability of multimodal LLMs, Shikra(Chen et al., 2023b) and MiniGPT-v2 (Chen et al., 2023a) process object coordinates as input and enable the localization ability by returning coordinates. However, these methods cannot produce segmentation masks and can only implicitly generate texts using LLMs rather than using a visual decoder for localization directly, which can be counterintuitive for image segmentation.

Recently, LISA (Lai et al., 2023) integrated a multi-modal LLM (Liu et al., 2023c) with a vision backbone and jointly trained a decoder to produce segmentation masks from language input. Despite using only 239 collected samples, LISA shows significant improvement in the reasoning process. However, its data is limited to entire objects, making it challenging for LISA to perform more fine-grained grounding. Motivated by this limitation, we introduce the *InstructPart* dataset, which contains instruction-part pairs, high-level affordance, low-level action, and part segmentation masks. With this dataset, we broaden the applicability of VLMs to various domains, such as manipulation, by enhancing their part grounding ability.

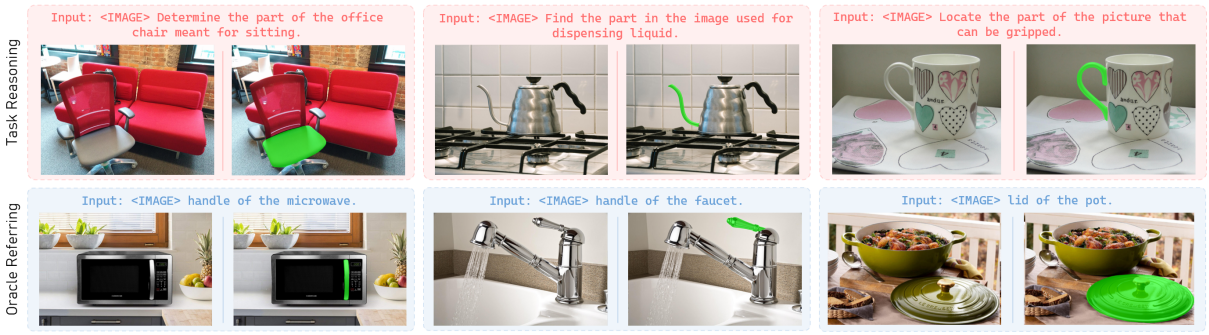


Figure 2: Examples from our InstructPart dataset are illustrated as follows: instruction queries are denoted in red text, while object and part names are indicated in blue. Each example includes an observation image (left), with the corresponding ground truth part segments (right), highlighted with a green mask.

## 3 The InstructPart Benchmark and Baseline Models

In this section, we describe the *InstructPart* benchmarking and introduce a simple baseline method for our benchmark.

### 3.1 InstructPart Task Definition

Motivated by scenarios where agents need to localize areas based on task-specific queries, we define two tasks. The first, **Task Reasoning Part Segmentation (TRPS)**, challenges models to combine linguistic reasoning with visual grounding. The second, **Oracle Referring Part Segmentation (ORPS)**, focuses exclusively on evaluating visual grounding using oracle information about the designated object and part.

**TRPS.** The TRPS task, illustrated in the first row of Fig. 2, is designed to explore the model's reasoning and part grounding abilities. The input is an instruction-image pair, and the goal is to identify the referred part's segmentation mask, as shown in green masks in Fig. 2. This task challenges the model to comprehend the instruction, analyze the image, and locate the corresponding part, Formally, the task is defined as: $\mathcal{F}(I_{\text{instruction}}, I_{\text{image}}) \Rightarrow M$, where $\mathcal{F}$ represents the evaluated model, and $I_{\text{instruction}} \in \{I_{\text{human}}, I_{\text{GPT}}\}$ is the instruction input that can either be annotated by human experts or rewritten by GPT-4.

**ORPS.** In the ORPS task, shown in the second row of Fig. 2, the model is provided with direct part names to ensure accurate textual input. The task can be formulated in two ways: We formulate the ORPS task in two formats:

1. Including both the part name and the object name, e.g., *the handle of the faucet*: $\mathcal{F}(P \text{ of } O, I_{\text{image}}) \Rightarrow M$.
2. Incorporating the affordance, e.g., *the handle of the cup that can be held*, which could assist the model in identifying the part: $\mathcal{F}(P \text{ of } O \text{ that } A_{\text{a}}, I_{\text{image}}) \Rightarrow M$, where $A_{\text{a}}$ refers to the affordance. We manually adjust the active and passive voice of the affordance according to ensure grammatical precision.

### 3.2 InstructPart Dataset

In line with our proposed tasks, we collect data to create the InstructPart dataset. This dataset is designed to evaluate the effectiveness of current models in understanding natural language and their ability to ground to specific parts. It comprises 2,400 images, carefully selected to align with everyday household tasks. Specifically, InstructPart includes 48 object classes, 44 part categories, 30 affordances, and 37 actions. During data selection, a uniform distribution of object classes is ensured to create a balanced dataset. More details are included in Appendix A.

In the first row of Fig. 2, we show annotated examples for the TRPS task. For each image sample, we manually design a task description based on the observed environment and the potential intention of an agent to interact with the object. For each sample, we annotate all the fine-grained segmentation masks relevant to the task description as the ground truth. These masks are human-labeled to ensure accuracy and alignment with human understanding of object parts, maintaining the high quality of our dataset. We deliberately avoid specific part names in the instructions to better adapt to real-world scenarios. For example, commonly used expressions such as *"Flush the toilet"* or *"Turn on the faucet"* are preferred over more detailed directives such as *"Press the toilet handle"* or *"Lift the faucet handle"*. The selection of these task descriptions aims to train models that are better at reasoning about

object parts and their affordances, rather than simply identifying the part name that would solve the task. By avoiding part names, our dataset more effectively analyzes the reasoning ability of models, requiring them to infer parts from implicit descriptions. We engaged six human experts to create free-form natural language task instructions, which were then refined using GPT-4 for grammatical precision and sentence diversity. This was followed by thorough human verification to prevent hallucinations or other issues that can arise from using large language models for phrase diversification. For the ORPS task, we use the part name and object name as the language input to evaluate the model's ability to directly ground to the part.

In addition to the instruction-image pairs, we provide the names of objects and parts relevant to the image, such as *seat of the chair*, *spout of the kettle*, *handle of the cup*. We also include a corresponding affordance and action for each instruction. Specifically, affordances refer to low-level actions performed to a specific part, like *"pull"*, *"push"*, or *"twist"*, while actions refer to the high-level function to be achieved, such as *"turn on"*, *"pick up"*, or *"open"*. Note that the affordance and action could be identical sometimes, e.g., *"pour"*, *"cut"*, etc. In the examples shown in the first row of Fig. 2, the affordances are *"support"*, *"pour"*, *"grip"*, and the actions are *"sit"*, *"pour"* and *"pick up"*. This allows us to categorize affordances into two levels, addressing the ambiguity in definitions as noted in previous studies (Nguyen et al., 2017; Roy and Todorovic, 2016; Myers et al., 2015). Note that in this work, we use the task descriptions and part names as the text input, while the affordance and action labels are reserved for future research.

In summary, the annotation for each of the samples in InstructPart can be represented as: $(I_{\text{task}}, I_{\text{image}}, O, P, M, A_{\text{affordance}}, A_{\text{action}})$, where these items refer to task instruction $I_{\text{task}}$, image observation $I_{\text{image}}$, object name $O$, part name $P$, segmentation mask $M$, affordance name $A_{\text{affordance}}$, and action name $A_{\text{action}}$). Note that $I_{\text{task}} \in \{I_{\text{human}}, I_{\text{GPT}}\}$, which means the text instruction is either directly annotated by humans or rewritten by GPT-4. More annotated examples can be found in Appendix B and H.

### 3.3 Baseline Method

For our InstructPart benchmark, we build a simple yet effective baseline model: Part Identification and Segmentation Assistant (PISA). PISA originates from LISA (Lai et al., 2023), which demonstrates superior capability in object-level reasoning segmentation. Motivated by (Li et al., 2023a), which shows the effectiveness of DINOv2 (Oquab et al., 2024) in extracting correspondence information among various parts, we improve LISA with a frozen DINOv2 backbone for feature extraction. As suggested by (Li et al., 2023a), we use linear layers to integrate multi-level features from DINOv2 for various granularity information fusion. The fused features are sent to an image decoder derived from SAM (Kirillov et al., 2023), where we apply Transpose Convolution and up-sampling for decoding in an alternating manner.

## 4 Experiments

### 4.1 Metrics

To evaluate our approach, we use standard metrics in LISA (Lai et al., 2023), namely gIoU and cIoU. gIoU reflects the average of all per-image Intersection-over-Unions (IoUs), while cIoU is defined by the cumulative intersection over the cumulative union. To evaluate the precision of the models, we adopt Precision@50 (P@50) metric as the previous referring segmentation works (Liu et al., 2023b; Mao et al., 2016) and develop a Precision@50:95 (P@50:95) metric according to COCO (Lin et al., 2014). The P@50 metric considers a mask to be a true positive when the IoU ratio exceeds 0.5, and P@50:95 calculates across a range of IoU thresholds from 0.50 to 0.95 with increments of 0.05, then averages across all the thresholds. The P@50:95 metric requires a higher least IoU for the prediction; hence, it is always lower than the P@50 metric. For the two metric types, IoU and Precision, the latter metric only counts those results greater than a threshold, hence can pose more challenges to the model and fairly evaluate the results with a high recall rate.

### 4.2 Evaluated Methods

Here, we introduce the set of baseline models utilized in our experiments. More details about the model settings can be found in Appendix C.

**Open-vocabulary Segmentation Models.** The open-vocabulary part segmentation model, i.e., VLPart (Sun et al., 2023), is intuitively suitable for our tasks since plentiful part segments were used for training. We also choose OVSeg (Liang et al., 2023) and SAN (Xu et al., 2023) to discover the performance of the open-vocabulary object seg-

| | Methods | Oracle Referring Part Segmentation | | | | | | | | Task Reasoning Part Segmentation | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Object-Part | | | | Object-Part-Affordance | | | | Human-Annotated | | | | GPT-4-Rewritten | | | |
| | | gIoU | cIoU | $P_{50\text{-}95}$ | $P_{50}$ | gIoU | cIoU | $P_{50\text{-}95}$ | $P_{50}$ | gIoU | cIoU | $P_{50\text{-}95}$ | $P_{50}$ | gIoU | cIoU | $P_{50\text{-}95}$ | $P_{50}$ |
| OVS | VLPart | 22.06 | 21.78 | 16.02 | 22.50 | 15.32 | 12.78 | 11.83 | 15.33 | 0.39 | 1.16 | 0.00 | 0.00 | 0.76 | 0.84 | 0.20 | 0.50 |
| | OVSeg | 28.58 | 20.49 | 10.37 | 22.33 | 28.60 | 20.99 | 10.87 | 22.50 | 22.44 | 14.11 | 7.07 | 15.33 | 23.14 | 15.51 | 7.13 | 15.17 |
| | SAN | 10.51 | 20.24 | 4.72 | 10.17 | 12.11 | 20.37 | 5.48 | 12.00 | 9.08 | 13.56 | 2.62 | 6.67 | 6.96 | 14.69 | 1.90 | 5.17 |
| RES | X-Decoder | 18.96 | 15.65 | 8.52 | 14.83 | 18.96 | 15.65 | 8.52 | 14.83 | 17.48 | 13.61 | 7.00 | 13.17 | 17.38 | 12.76 | 6.90 | 13.17 |
| | SEEM | 13.54 | 14.63 | 6.33 | 10.50 | 13.54 | 14.63 | 6.33 | 10.50 | 13.52 | 14.09 | 4.97 | 9.83 | 14.53 | 14.19 | 4.57 | 10.67 |
| | TRIS | 23.02 | 19.90 | 6.97 | 17.50 | 23.11 | 19.65 | 6.98 | 18.50 | 21.97 | 17.83 | 6.68 | 15.00 | 22.66 | 18.52 | 7.03 | 16.83 |
| | G-SAM | 34.33 | 24.83 | 15.03 | 28.83 | 33.63 | 24.79 | 14.42 | 27.83 | 29.95 | 21.45 | 11.98 | 25.17 | 29.57 | 21.88 | 11.60 | 23.00 |
| RS | LISA | 34.46 | **39.44** | 17.48 | 32.67 | **35.77** | **39.62** | **18.78** | **34.50** | **32.11** | **30.25** | **16.98** | **30.00** | **29.75** | **27.44** | **15.08** | **27.83** |
| | Shikra | 4.50 | 7.20 | 2.67 | 4.17 | 9.36 | 15.37 | 4.92 | 7.83 | 1.70 | 3.48 | 0.83 | 1.50 | 14.65 | 12.95 | 8.40 | 13.33 |
| | MiniGPT-v2 | **35.65** | 36.05 | **18.38** | **33.17** | 34.58 | 35.11 | 18.50 | 34.27 | 26.29 | 19.46 | 13.00 | 24.00 | 29.67 | 21.37 | 15.07 | 24.17 |
| | Average | 22.56 | 22.02 | 10.65 | 19.67 | 21.95 | 21.10 | 10.66 | 19.81 | 17.49 | 14.90 | 7.11 | 14.44 | 17.59 | 16.02 | 7.79 | 14.98 |

Table 2: Results on ORPS (left) and TRPS (right) tasks. We divide the methods into three categories, namely, open-vocabulary segmentation (OVS), referring expression segmentation (RES), and reasoning segmentation (RS). The best results are **bolded**, and the second-best are underlined.

mentation methods on our task. We select the best-reported models for the three methods.

**Refering Segmentation Models.** We conduct experiments with off-the-shelf models including X-Decoder (Zou et al., 2023a), SEEM (Zou et al., 2023b), and TRIS (Liu et al., 2023b). Besides, we also evaluate Grounding-DINO (Liu et al., 2023d), which has provided a great open-vocabulary referring detection ability and has been integrated with SAM (Kirillov et al., 2023), namely Grounded-SAM. We adopt the best models for these methods.

**Reasoning Segmentation Models.** For our tasks, LISA (Lai et al., 2023) is a natural choice since it can return masks and has been trained on several part segmentation datasets (He et al., 2022; Chen et al., 2014; Ramanathan et al., 2023). Other multi-modal LLMs, including Shikra (Chen et al., 2023b) and MiniGPT-v2 (Chen et al., 2023a) also have localization ability and have been chosen for our evaluation. Since they can only return bounding box outputs, we use the results as box prompts for SAM (Kirillov et al., 2023) to get a mask output for fair comparison.

**Grid-based GPT-4V.** The recent release of GPT-4V has demonstrated remarkable advancements in complex visio-linguistic reasoning (Yang et al., 2023b). However, GPT-4V API cannot return segmentation mask output directly, and our preliminary experiments showed that GPT-4V performs poorly when it is asked to generate text coordinates. As a result, we first use Grounding-DINO (Liu et al., 2023d) to find the bounding box of the entire object and crop it, then ask GPT-4V to virtually divide the box to $7 \times 7$ grids and identify the grids including the desirable parts. Afterward, the coordinates of the grids are used as a prompt for SAM (Kirillov et al., 2023) to obtain the segmentation mask.

**SoM-based GPT-4V.** SoM (Yang et al., 2023a) proposes to label the masks obtained by SAM (Kirillov et al., 2023) with numbers in the center of each object. As it proves that precise referring can boost the performance of GPT-4V, we apply a similar manner for our part segmentation task.

**PISA and Fine-tuning.** To evaluate our proposed method, we use all training data of LISA (Lai et al., 2023) for pertaining and fine-tuning with 1,800 samples of our data. As a comparison, we also fine-tune LISA with the same data. Besides, we also train the models with multiple numbers of samples. More results can be found in Appendix D.

### 4.3 Quantitative Results of SOTA VLMs

**Open-sourced VLMs Results.** The left part of Tab. 2 shows the result of our ORPS task, where object and part names are explicitly embedded into a template, mitigating the need for models' reasoning ability. The right part of Tab. 2 shows the result of TRPS, where part names are not present in the instruction and require more reasoning ability to understand the implicit meaning. Comparing the left and right parts of Tab. 2 we can find that the performance of oracle referring task is generally better than that of task reasoning. This demonstrates that current models lack the reasoning ability to infer from a task-image pair to the correct interactive part. For the ORPS task, incorporating the affordance in the instruction leads to no apparent increase in the average

performance. This indicates that most models may not possess the common sense to relate a part to an affordance, suggesting the potential of InstructPart for affordance learning. Besides, for the TRPS task, we can find that GPT-4 rewritten instructions lead to overall better performances. This indicates that the precise instruction descriptions generated by GPT-4 align more effectively with the language embedding space of multimodal LLMs, enhancing the reasoning capabilities of vision-language models for handling instructions.

**GPT-4V Based Methods Results.** Tab. 3 shows the results of two GPT-4V segmentation methods. We test the two methods on the oracle referring task to explore GPT-4V's localization ability. We select a subset consisting of 226 samples from the dataset according to the original category distribution. Although the results cannot be fairly compared with other methods in Tab. 2, it still reveals the poor performance of GPT-4V. Two reasons may explain this: 1) While GPT-4V can localize objects (Yang et al., 2023a), we hypothesize that it is not trained directly on fine-grained part data. 2) Labeling numbers in the center of fine-grained parts may lead to overlapping and ambiguity in referring.

| Methods | Object-Part | | | |
|---|---|---|---|---|
| | gIoU | cIoU | $P_{50\text{-}95}$ | $P_{50}$ |
| Grid-based GPT-4V | 14.14 | 17.15 | 5.67 | 12.37 |
| SoM-based GPT-4V | 25.41 | 26.82 | 17.90 | 25.81 |

Table 3: GPT-4's performance in the object-part oracle referring part segmentation task, as applied to a subset of InstructPart.

### 4.4 Quantitative Results of Fine-tuning with InstructPart

Tab. 4 shows the results of TRPS task with human-annotated instructions. The pre-trained PISA outperforms LISA by a large margin, demonstrating its strong reasoning part segmentation ability. After fine-tuning, both LISA and PISA gain great improvement in all metrics, indicating the exceptional quality and training utility of our data.

### 4.5 Qualitative Results

Fig. 3,4,5 shows the visualization results on the TRPS task. The first column depicts the ground truth labels, and the remaining columns include the results of off-the-shelf VLMs: X-Decoder (Zou et al., 2023a) , SEEM (Zou et al., 2023b) , TRIS (Liu et al., 2023b) , Grounded-SAM (Kirillov et al., 2023; Liu et al., 2023d) , MiniGPT-v2 (Chen et al., 2023a) , LISA (Lai et al., 2023). The last two columns show the results of fine-tuned LISA and PISA models. As shown by the examples, most VLMs tend to either obtain the entire object area or miss the correct regions, demonstrating the challenging tasks provided by InstructPart. In Fig. 3, we present examples where the fine-tuned PISA shows superior visual part segmentation results, demonstrating the effectiveness of our proposed method. Besides, both the pre-trained and fine-tuned LISA models also demonstrate great potential in part grounding. Here, we visualize additional results of the VLMs and fine-tuned models. As shown in Fig. 5, the pre-trained LISA(Lai et al., 2023) can better identify desired parts compared to other VLMs. This indicates the evaluation usage of our InstructPart dataset, where all the advanced VLMs can be evaluated and compared. Furthermore, in Fig. 4, the pre-trained LISA fails to recognize target parts, similar to other VLMs, while both fine-tuned models significantly improve the results. More visualizations are available in Appendix G.

| Methods | gIoU | cIoU | $P_{50\text{-}95}$ | $P_{50}$ |
|---|---|---|---|---|
| LISA-Pretrained | 32.11 | 30.25 | 16.98 | 30.00 |
| PISA-Pretrained | 43.46 | 46.76 | 20.00 | 44.50 |
| LISA-Tuned | 71.26 | 72.14 | 57.73 | 79.33 |
| PISA-Tuned | **76.19** | **78.39** | **62.20** | **87.00** |

Table 4: Comparison of pre-training and fine-tuning results. We use all datasets that LISA was trained on to get the pre-trained model. Fine-tuned models are trained with 1,800 samples in InstructPart.

## 5 Discussion

**Scale of InstructPart dataset.** We consider InstructPart a sufficient task-oriented part segmentation dataset for the following reasons: ***1)*** The size of InstructPart already exceeds that of several recent Vision-Language evaluation datasets, such as MMStar (Chen et al., 2024) (1500 samples, Arxiv'24), VisIT-Bench (Bitton et al., 2024) (592 images, NeurIPS'23), WHOOPS! (Bitton-Guetta et al., 2023) (500 images, ICCV'23), and TIFA160 (Hu et al., 2023) (800 generated images, ICCV'23). We believe that our data are adequate for thorough evaluations of current models. ***2)*** InstructPart addresses a gap in data related to reasoning about robot-object interaction and part segmentation (e.g., PartImageNet includes only one relevant category: bottle). ***3)*** Fine-tuning LISA


Figure 3: Qualitative comparison of different VLMs and the fine-tuned models. In these examples, the pre-trained LISA falls short of recognizing the correct part. After fine-tuning, PISA shows better potential for part understanding than LISA. More results can be found in Figure G13.

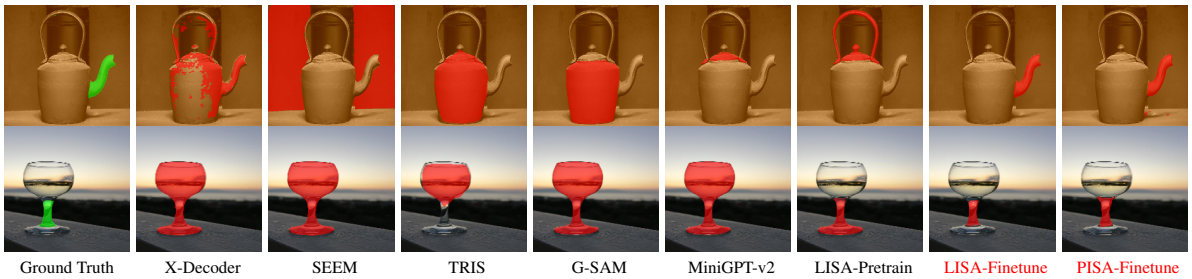


Figure 4: Qualitative comparison of different VLMs and the fine-tuned models. In these examples, the pre-trained LISA falls short of recognizing the correct part. After fine-tuning, both LISA and PISA perform well on the part identification. More results can be found in Figure G14.

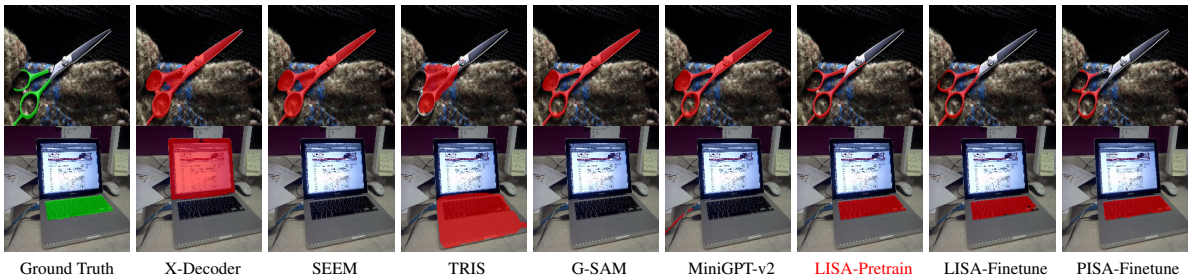


Figure 5: Qualitative comparison of different VLMs and the fine-tuned models. In these examples, the pre-trained LISA already delivers good identification of the target parts. More results can be found in Figure G14.

with a small subset of our dataset (200 samples) can lead to a nearly 100% performance increase (results included in the Appendix D), demonstrating the exceptional quality and utility of our dataset.

**Novelty of InstructPart.** The novelty of InstructPart lies not in our baseline method but in our comprehensive evaluation of SOTA VLMs, revealing their limitations in complex language reasoning and part-grounding. We hope that the established benchmark will foster progress in VLM-based part grounding, ultimately enhancing the real-world applicability of VLMs across various scenarios. Our proposed baseline is simple yet demonstrates the superior quality and training potential of our dataset. Additionally, we conduct a case study on real-world grasping data (see Appendix I), showing the potential of InstructPart for broader applications.

**Potential Applications.** Our dataset contains samples in various scenarios, including kitchen, living room, outdoor, etc., and can be used for robot manipulation and visual question answering. Besides, our dataset can provide data for affordance learning and semantic understanding. For benchmarking usage, one can also use the entire 2,400 images to evaluate current advanced VLMs.

## 6 Conclusion

In this work, we introduce a new benchmark, InstructPart, a novel dataset containing part annotations for common household objects as well as two tasks: task reasoning and oracle referring segmentation. We showed that even the most advanced vision-language models struggle with tasks that link specific affordances to the corresponding parts of an object when given high-level instructions. By fine-tuning a simple baseline with our dataset, we achieve a twofold improvement in part segmentation, showcasing the quality and training utility of our data. Through our work, we highlight a significant gap in foundation models for task-oriented part segmentation and hope that with our dataset, we can pave the way for further research into object-part reasoning.

**Limitations.** In this work, we propose a baseline method that has achieved significant performance improvement. However, we have not fully explored the potential of our dataset, as we did not utilize the affordance labels for training. An intriguing research topic would be to combine affordance learning and language reasoning to achieve even better performance.

## Acknowledgements

This work has been funded in part by the Army Research Laboratory (ARL) award W911NF-23-2-0007 and W911QX-24-F-0049, DARPA award FA8750-23-2-1015, and ONR award N00014-23-1-2840.

## References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schmidt. 2024. Visit-bench: A dynamic benchmark for evaluating instruction-following vision-and-language models. *Advances in Neural Information Processing Systems*, 36.

Nitzan Bitton-Guetta, Yonatan Bitton, Jack Hessel, Ludwig Schmidt, Yuval Elovici, Gabriel Stanovsky, and Roy Schwartz. 2023. Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2616–2627.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023a. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. 2023b. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*.

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

Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 2014. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978.

Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 2021. 3d affordancenet: A benchmark for visual object affordance understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1778–1787.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. 2020. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453.

Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. 2021. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15752–15761.

Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. 2023. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7081–7091.

Tianyu Han, Lisa C Adams, Sven Nebelung, Jakob Nikolas Kather, Keno K Bressem, and Daniel Truhn. 2023. Multimodal large language models are generalist medical image interpreters. *medRxiv*, pages 2023–12.

Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. 2022. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer.

Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. 2016. Segmentation from natural language expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 108–124. Springer.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. 2023. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20406–20417.

Siyuan Huang, Iaroslav Ponomarenko, Zhengkai Jiang, Xiaoqi Li, Xiaobin Hu, Peng Gao, Hongsheng Li,

and Hao Dong. 2024. Manipvqa: Injecting robotic affordance and physically grounded information into multi-modal large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7580–7587. IEEE.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643*.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. 2023. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*.

Gen Li, Deqing Sun, Laura Sevilla-Lara, and Varun Jampani. 2023a. One-shot open affordance learning with foundation models. *arXiv preprint arXiv:2311.17776*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Samuel Li, Sarthak Bhagat, Joseph Campbell, Yaqi Xie, Woojun Kim, Katia P. Sycara, and Simon Stepputtis. 2024. Shapegrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10527–10534.

Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. 2023. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070.

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

Chang Liu, Henghui Ding, and Xudong Jiang. 2023a. Gres: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23592–23601.

Fang Liu, Yuhao Liu, Yuqiu Kong, Ke Xu, Lihe Zhang, Baocai Yin, Gerhard Hancke, and Rynson Lau. 2023b. Referring image segmentation using text supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22124–22134.

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

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. 2023d. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. 2021. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2125–2134.

Hongchen Luo, Wei Zhai, Jing Zhang, Yang Cao, and Dacheng Tao. 2022. Learning affordance grounding from exocentric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2252–2261.

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

Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. 2021. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823.

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. 2019. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918.

Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. 2015. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1374–1381. IEEE.

Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. 2017. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5908–5915. IEEE.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. Dinov2: Learning robust visual features without supervision. *Preprint*, arXiv:2304.07193.

Shuyi Ouyang, Hongyi Wang, Shiao Xie, Ziwei Niu, Ruofeng Tong, Yen-Wei Chen, and Lanfen Lin. Slvit:

Scale-wise language-guided vision transformer for referring image segmentation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 1294–1302.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. 2023. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151.

Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Chen, Angjoo Kanazawa, and Ken Goldberg. 2023. Language embedded radiance fields for zero-shot task-oriented grasping. *Preprint*, arXiv:2309.07970.

Anirban Roy and Sinisa Todorovic. 2016. A multi-scale cnn for affordance segmentation in rgb images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 186–201. Springer.

Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. 2023. Going denser with open-vocabulary part segmentation. *arXiv preprint arXiv:2305.11173*.

Zifu Wan, Yaqi Xie, Ce Zhang, Zhiqiu Lin, Zihan Wang, Simon Stepputtis, Deva Ramanan, and Katia P Sycara. 2024. Instructpart: Affordance-based part segmentation from language instruction. In *AAAI-2024 Workshop on Public Sector LLMs: Algorithmic and Sociotechnical Design*.

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

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. 2020. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107.

Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. 2023. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2945–2954.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023a. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. 2022. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18155–18165.

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

Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. 2018. Deep part induction from articulated object pairs. *arXiv preprint arXiv:1809.07417*.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2023. Ferret: Refer and ground anything anywhere at any granularity. *Preprint*, arXiv:2310.07704.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2019. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321.

Xingcheng Zhou, Mingyu Liu, Bare Luka Zagar, Ekim Yurtsever, and Alois C. Knoll. 2023. Vision language models in autonomous driving and intelligent transportation systems. *Preprint*, arXiv:2310.14414.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. 2023a. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15116–15127.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. 2023b. Segment everything everywhere all at once. *arXiv preprint arXiv:2304.06718*.

# Appendix

## A Dataset Details

InstructPart dataset is collected from Flickr[1] website and AGD20K (Luo et al., 2022), where we selected free-licensed images from both sources. To better understand the categories of our dataset, we follow ADE20K (Zhou et al., 2019) to provide the distribution of objects and parts within InstructPart. As shown in Fig. 6, the dataset comprises 2,400 data items, encompassing 48 object classes and 44 part classes, which together form 98 distinct object-part pair classes. Besides, we also provide a word cloud to visualize the object-part classes and affordance-action categories, as depicted in Fig. A7 and Fig. A8, respectively. This diversity in classes indicates our dataset's wide coverage of various daily scenes, offering robust criteria for comprehensively analyzing the proficiency of current models in understanding task instructions and segmenting parts. Furthermore, this suggests that our dataset can be valuable for broad areas, including semantic segmentation, robot manipulation, visual question answering, and more.

## B Annotation Example

Fig. B9 presents two examples of annotations from our InstructPart dataset, focusing on the handle of a cup and the lid of a pod, respectively. In each JSON dictionary, the names of the object and its specific part are noted, aligned with a task instruction that pertains to a particular part shown in the image. Additionally, both a low-level affordance name and a high-level action name are provided in relation to the instruction.

Besides, in Fig. B10, we provide more examples that contain occlusions and human interactions to showcase the complexity of our dataset.

## C Evaluated Model Details

**Open-vocabulary segmentation models.** We choose OVSeg (Liang et al., 2023) and SAN (Xu et al., 2023) to discover the performance of the open-vocabulary object segmentation methods on our task. We select the best-reported models for the two methods, *ovseg_swinbase_vitL14_ft_mpt.pth* and *san_vit_large_14.pth* respectively.

**Refering expression segmentation.** We conduct experiments with off-the-shelf models including X-Decoder (Zou et al., 2023a), SEEM (Zou et al., 2023b), and TRIS (Liu et al., 2023b). We adopt *xdecoder_focalt_last.pt*, *seem_focall_v1.pt*, and *stage2_refcocog_google.pth* for the three models respectively. Besides, we also evaluate Grounding-DINO (Liu et al., 2023d), which has witnessed a great open-vocabulary referring detection ability and been integrated with SAM (Kirillov et al., 2023) to a project, Grounded-SAM[2].

**Reasoning segmentation.** For our tasks, LISA (Lai et al., 2023) can naturally be a good choice since it can return masks and has been trained on several part segmentation datasets. As a result, it is interesting to explore whether it possesses the ability to understand instructions and find part segments. Other multi-modal LLMs, including VisionLLM (Wang et al., 2023), Shikra (Chen et al., 2023b), also have localization ability. Since they can only return bounding box outputs, we use the results as box prompts for SAM to get a mask output for fair comparison. However, we cannot test on VisionLLM since it has not release code.

To prompt LISA, we follow its original setting to add *"Please output the segmentation mask."* at the end of each instruction. Besides, in order to formulate a query for the oracle referring part segmentation task, we embed the object and part name in a format of: "Where is the $I_{\text{text}}$ in the image", where $I_{\text{text}}$ stands for the text input.

To prompt Shikra, we integrate our instruction in its original template as follows:

- Instruction referring part segmentation:
  <$I_{\text{text}}$>. Can you point out all the related parts in the image <$I_{\text{image}}$> and provide the coordinates of their locations?

- Oracle referring part segmentation:
  Can you point out all the <$I_{\text{text}}$> in the image <$I_{\text{image}}$> and provide the coordinates of their locations?

We adopt LISA-7B-v1 (Lai et al., 2023) model that has been fine-tuned on both training and validation data of LISA's dataset. As for Shikra, we select the frequently updated model, Shikra-7B-delta-v1-0708.

[1] https://www.flickr.com/

[2] https://github.com/IDEA-Research/Grounded-Segment-Anything

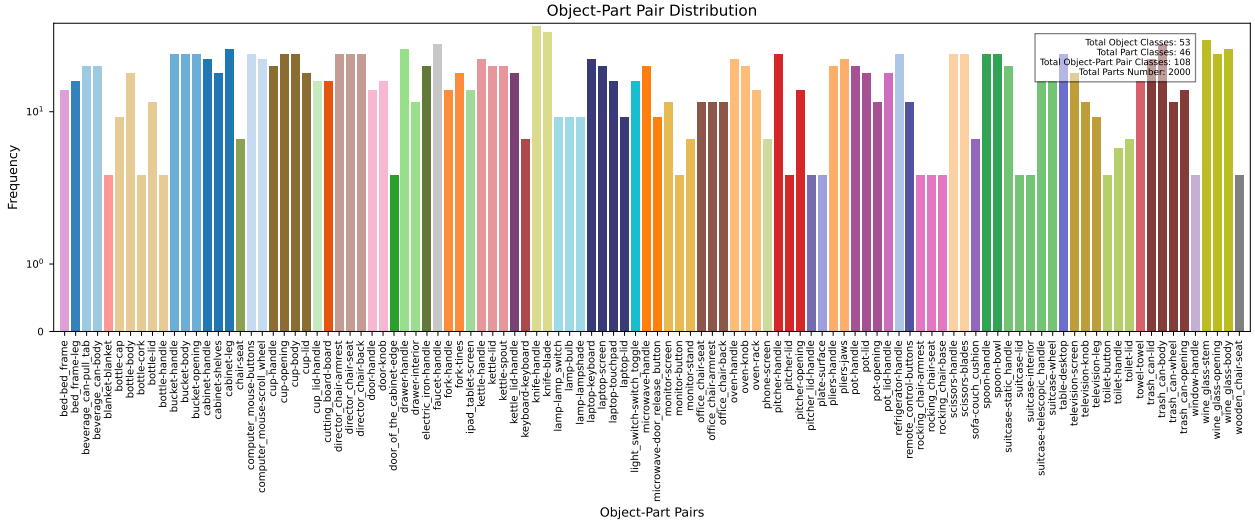


Figure 6: Object-part pair distribution. We collect 2,400 data pieces in total, containing 48 object classes and 44 part classes, constituting 98 different object-part pair classes. The x-axis shows the name of the object-part pairs, and the y-axis shows the frequency of each item. The parts belonging to the same object classes are highlighted with the same color in the bar chart.

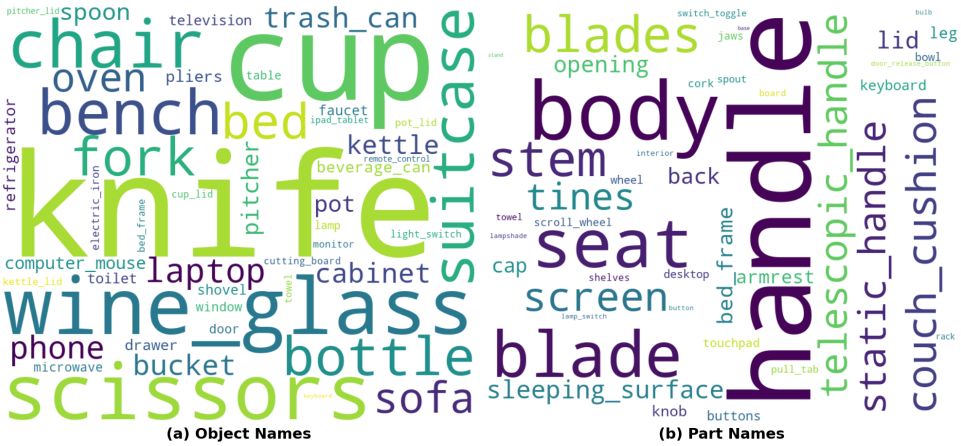


Figure A7: InstructPart dataset object and part classes. The left part shows the object class names and the right part shows the part class names.

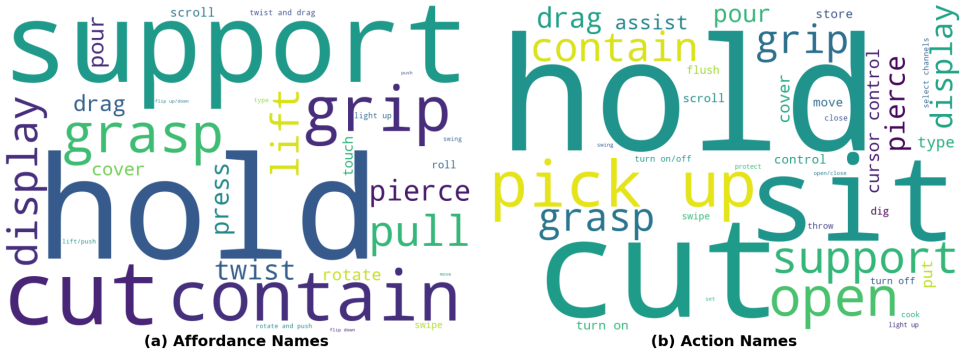


Figure A8: InstructPart dataset affordance and action categories. The left part shows the affordance names and the right part shows the action names. Specifically, affordances refer to low-level actions performed to a specific part, while actions refer to the high-level function to be achieved.


```
"image_path": "8102393963_5f895d013c_o.jpg"
"part_list":
    "object": "cup"
    "part": "handle"
    "affordance": "grip"
    "action": "hold"
    "instruction":
        "If I want to hold the cup, which part in the picture can be utilized?"
        "Locate the part of the picture that can be grasped to hold the cup."
        "handle of the cup"
        "handle of the cup that can be gripped"
```



```
"image_path": "pot_1.jpg"
"part_list":
    "object": "pot"
    "part": "lid"
    "affordance": "cover"
    "action": "cover"
    "instruction":
        "Which part is used to cover the pot?"
        "Identify the part in the picture that serves as a pot lid."
        "handle of the cup"
        "lid of the pot that can cover"
```


Figure B9: Annotation Example: Each data item is represented by a JSON dictionary, which details the components involved. This includes the object to which these parts belong, the name of each part, a specific instruction related to these parts, a low-level affordance associated with the instruction, and a high-level action performed on the parts. Corresponding parts are highlighted in green in the images on the right.

Figure B10: More complex examples in InstructPart, including occlusions and human-object interactions.

## D Effect of Training Samples

To verify the quality and training potential of the PISA dataset, we gradually increase the number of training samples from 200 to 1,800 and observe the performance improvement. Specifically, we start with 200 samples for training, then gradually increase the number of training samples to 600, 1,200, and finally 1,800. Each increment includes all the previously used training samples. As shown in Fig. D11, with the increasing number of training samples, the IoU metric gradually increases and exhibits a logarithmic convergence tendency. This indicates that our high-quality data significantly boosts performance, even with just 200 samples. The performance of both models improves substantially from the outset.

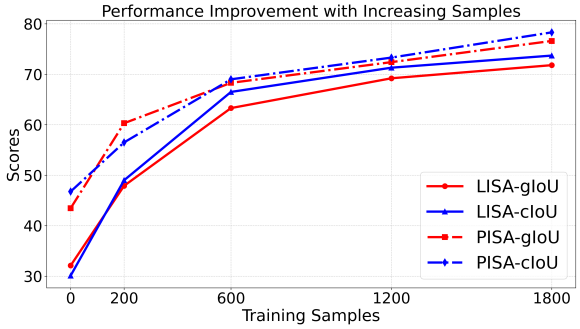


Figure D11: Performance improvement with increasing number of training samples. We gradually add training samples to 200, 600, 1,200, and 1,800.

## E Does object recognition hinder part segmentation?

To explore whether the bottleneck lies in current VLMs' object recognition ability, we use the object classes as instruction and obtain the results in Tab. E5. Since we do not have object-level labels, we use the recall rate as a reflection of whether the model can find the entire object. From the results in Tab. E5, the precision is much lower compared to the recall rate, and the recall rate is close to 1 after the third quartile (75th percentile). This indicates that the predicted masks can generally cover the part labels, so the poor performance of TRPS cannot derive from the object recognition ability.

| Methods | Object-Level | | | | |
|---|---|---|---|---|---|
| | Prec. | Rec.@A | Rec.@25% | Rec.@50% | Rec.@75% |
| OVSeg | 20.93 | 81.80 | 85.00 | 99.26 | 100.00 |
| SAN | 19.46 | 73.55 | 56.37 | 98.49 | 100.00 |
| G-SAM | 25.53 | 89.83 | 92.59 | 96.99 | 99.37 |

Table E5: Precision and recall rate on object-level segmentation results. The five metrics refer to precision (Prec.), average recall (Rec.@A), first quartile recall (Rec.@25%), median recall (Rec.@50%), and third quartile recall (Rec.@75%), respectively.

## F GPT-4V Qualitative Results

We show the results of GPT-4V-based methods, namely SoM-based GPT-4V and Grid-based GPT-4V, in Fig. F12. While GPT-4V-based methods deliver clear boundaries, they sometimes select the wrong segments from SAM (Kirillov et al., 2023), leading to poor overall performance.

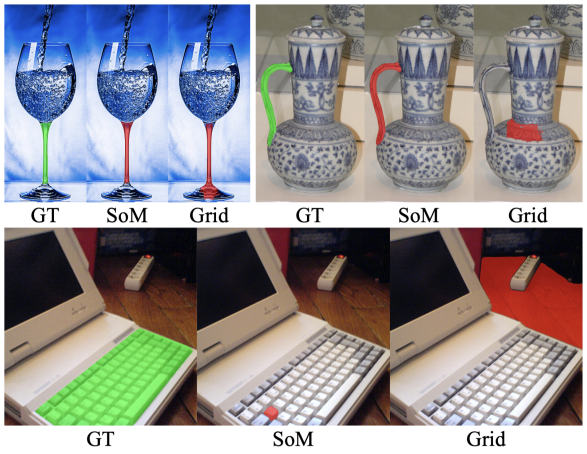


Figure F12: GPT4-V based methods.

| Fig. G13 | Fig. G14 | Fig. G15 |
|---|---|---|
| 1. 1009786005_d4a02fd811_o-faucet-handle<br>2. 2329134125_8a71be7470_o-kettle-handle<br>3. 3088942376_8681bb276f_o-spoon-handle<br>4. cup_000294-cup-handle<br>5. knife_000911-knife-handle<br>6. 410044558_6145ff0aaa_o-pot-handle<br>7. laptop_000445-laptop-keyboard<br>8. knife_000691-knife-handle | 1. 4178009615_ed8921d0d1_k-kettle-spout<br>2. cup_000324-cup-handle<br>3. bottle_002805-bottle-body<br>4. knife_000568-knife-handle<br>5. knife_000953-knife-blade<br>6. 34465720_f8f20ee31a_c-scissors-handle<br>7. 381204305_e5e937fccc_h-pitcher-handle<br>8. bench_001273-bench-seat<br>9. fork_002954-fork-handle<br>10. knife_000154-knife-handle<br>11. shovel_1-shovel-blade<br>12. suitcase_001098-suitcase-telescopic_handle<br>13. wine_glass_001774-wine_glass-stem<br>14. dining_4-chair-seat | 1. 2491323916_a05ac3648f_o-knife-handle<br>2. 4580224808_1194613deb_o-chair-seat<br>3. 4471021242_b9d855f193_k-bucket-handle<br>4. 8607578325_25221a7726_h-spoon-handle<br>5. bench_002898-bench-seat<br>6. cup_001798-cup-handle<br>7. cup_002055-cup-handle<br>8. knife_000530-knife-blade<br>9. scissors_001402-scissors-handle<br>10. cup_002062-cup-handle<br>11. 2939090254_2f01ebed6d_o-computer_mouse-scroll_wheel<br>12. 6217625873_411169d784_o-laptop-keyboard<br>13. cup_001104-cup-handle<br>14. fork_001529-fork-handle |

Table G6: Index name for samples in Fig. G13, Fig. G14, and Fig. G15.

## G More Qualitative Results

In Figure 3-5 of the main paper, we only include six qualitative results due to space limitations. In Fig. G13, we present more examples where the fine-tuned PISA shows superior visual part segmentation results, demonstrating the effectiveness of our proposed method. Besides, both the pre-trained and fine-tuned LISA models also demonstrate great potential in part grounding. Here, we visualize additional results of the VLMs and fine-tuned models. As shown in Fig. G15, the pre-trained LISA(Lai et al., 2023) can better identify desired parts compared to other VLMs. This indicates the evaluation usage of our InstructPart dataset, where all the advanced VLMs can be evaluated and compared. Furthermore, in Fig. G14, the pre-trained LISA fails to recognize target parts, similar to other VLMs, while both fine-tuned models significantly improve the results.

In Tab. G6, we provide a list containing the name of each sample we evaluate so that their language input can be easily retrieved from our dataset.


Figure G13: Qualitative comparison of different VLMs and the fine-tuned models. In these examples, the pre-trained LISA falls short of recognizing the correct part. After fine-tuning, PISA shows better potential for part understanding than LISA.

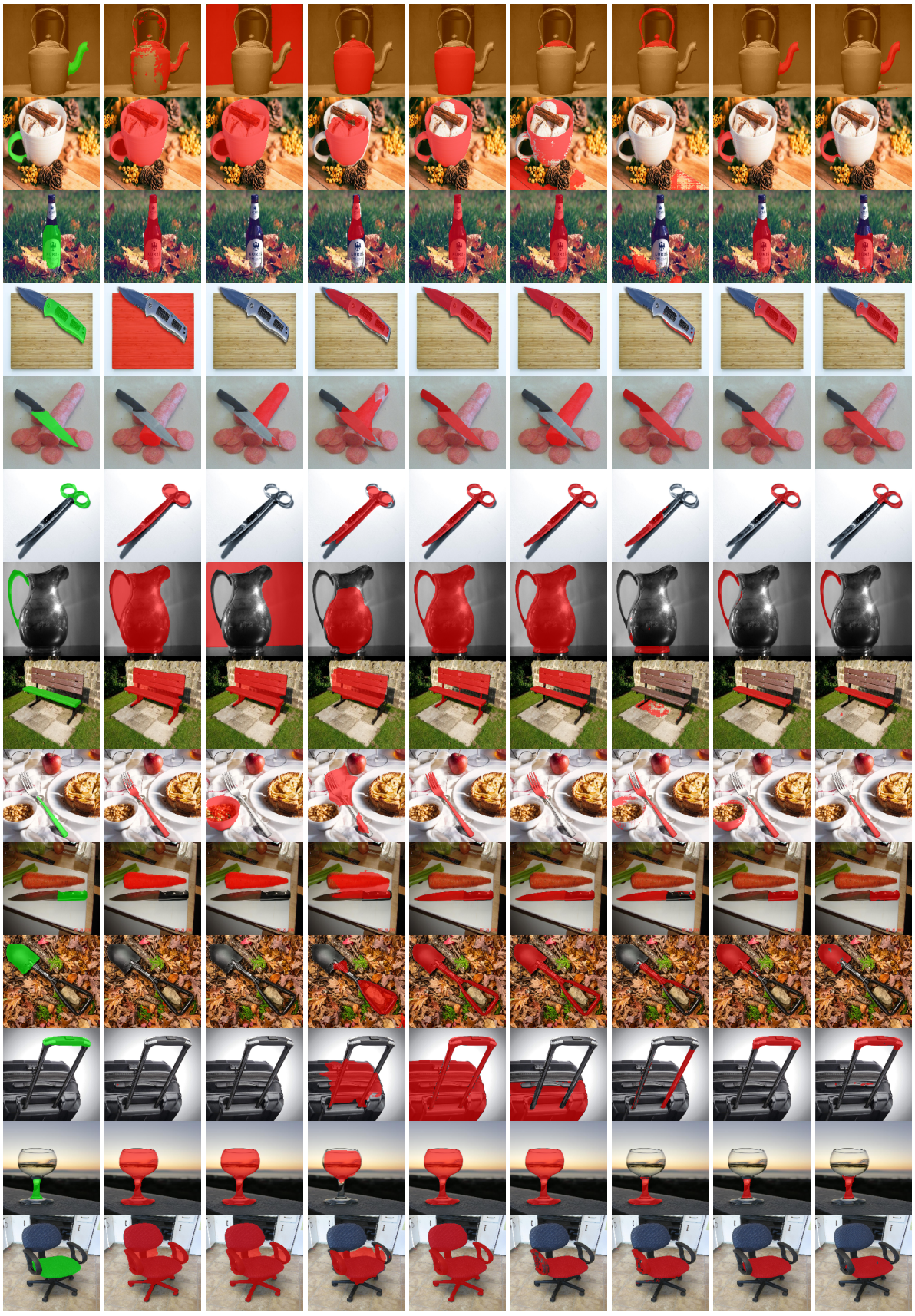


Figure G14: Qualitative comparison of different VLMs and the fine-tuned models. In these examples, the pre-trained LISA falls short of recognizing the correct part. After fine-tuning, both LISA and PISA perform well on the part identification.


Figure G15: Qualitative comparison of different VLMs and the fine-tuned models. In these examples, the pre-trained LISA already delivers good identification of the target parts.

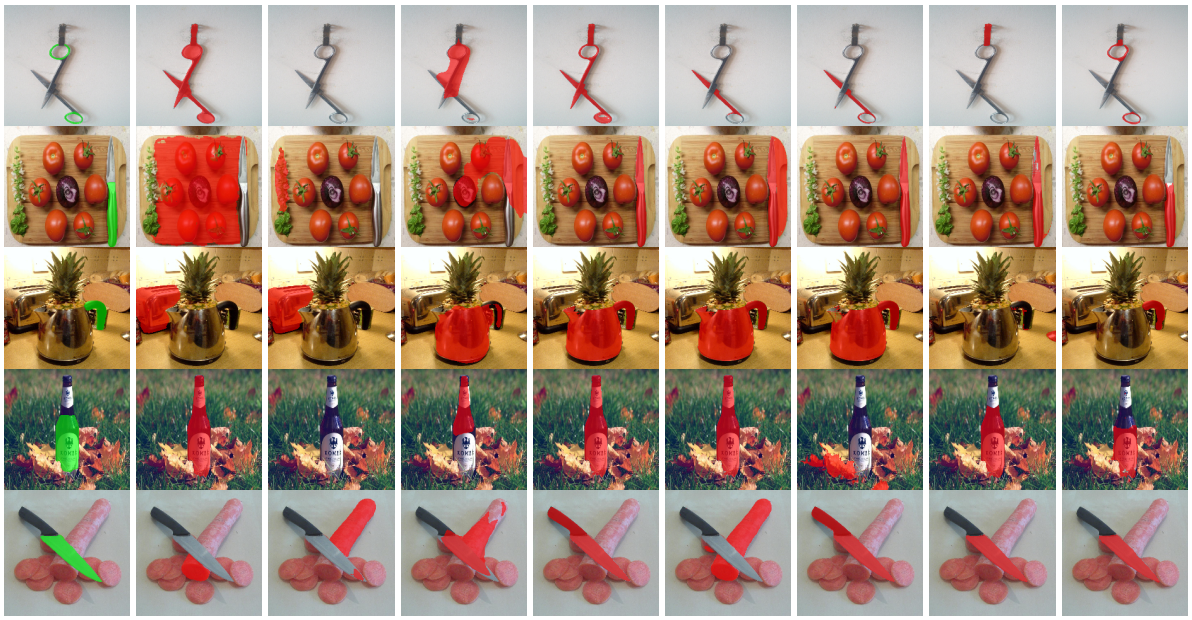


Figure H16: More qualitative examples with corresponding annotations recorded in Table H7.

## H More Annotation Samples

In addition to the annotation examples shown in Fig. B9, we include five more annotations for the samples in Fig. H16 in Table H7. The listed annotations correspond to the order of the images.

```
{
    "image_path": "538210619_c4def94c9b_o.jpg",
    "part_list": [
        {
            "object": "scissors",
            "part": "handle",
            "affordance": "hold",
            "action": "hold",
            "instruction": [
                "If I want to use the scissors, which part in the picture should I put my
                    ↪ fingers in?",
                "Describe the part of the scissors in the picture where fingers should be
                    ↪ placed.",
                "Where is the handle of the scissors in this image?",
                "Where is the handle of the scissors that can be held in this image?",
                "handle of the scissors",
                "handle of the scissors that can be held"
            ]
        }
    ]
}
```



```
{
    "image_path": "knife_002845.jpg",
    "part_list": [
        {
            "object": "knife",
            "part": "handle",
            "affordance": "hold",
            "action": "pick up",
            "instruction": [
                "If I want to pick up the knife, which part in the picture can be used?",
                "Which part of the knife is safe to hold when picking it up?",
                "Where is the handle of the knife in this image?",
                "Where is the handle of the knife that can be held in this image?",
                "handle of the knife",
                "handle of the knife that can be held"
            ]
        }
    ]
}
```

```
{
    "image_path": "2329134125_8a71be7470_o.jpg",
    "part_list": [
        {
            "object": "kettle",
            "part": "handle",
            "affordance": "hold",
            "action": "hold",
            "instruction": [
                "Which part in the picture can be utilized to hold the kettle?",
                "In the image, identify the part of the kettle that's meant to be held.",
                "Where is the handle of the kettle in this image?",
                "Where is the handle of the kettle that can be held in this image?",
                "handle of the kettle",
                "handle of the kettle that can be held"
            ]
        }
    ]
}
```

```
{
    "image_path": "bottle_002805.jpg",
    "part_list": [
        {
            "object": "bottle",
            "part": "body",
            "affordance": "hold",
            "action": "hold",
            "instruction": [
                "If I want to hold the bottles, which parts in the picture can be utilized?",
                "To hold the bottles, which parts are designed for grip?",
                "Where is the body of the bottle in this image?",
                "Where is the body of the bottle that can be held in this image?",
                "body of the bottle",
                "body of the bottle that can be held"
            ]
        }
    ]
}
```

```
{
    "image_path": "knife_000953.jpg",
    "part_list": [
        {
            "object": "knife",
            "part": "blade",
            "affordance": "cut",
            "action": "cut",
            "instruction": [
                "If I want to use the knife to cut the carrots, which part in the picture
                    ↪ should be used?",
                "Identify the part of the knife ideal for slicing the carrots.",
                "Where is the blade of the knife in this image?",
                "Where is the blade of the knife that can cut in this image?",
                "blade of the knife",
                "blade of the knife that can cut"
            ]
        }
    ]
}
```


Table H7: Corresponding annotations for images in Fig. H16.

## I A Case Study on Real-world Grasping Data.

Grasping is one vital aspect that our InstructPart benchmark aims to facilitate. Consequently, we evaluate the model trained with our data in a real-world tabletop grasping environment. We use the table setup from ShapeGrasp (Li et al., 2024), which consists of 38 objects covering 12 general categories and 49 tasks. These categories and tasks are the same as those in LERF-TOGO (Rashid et al., 2023). More details about the dataset are included in the supplementary material. Our trained PISA model is evaluated on the zero-shot task-oriented grasping task, as described in (Li et al., 2024; Rashid et al., 2023). We compare the successful part selection rate, defining a successful part selection as our output segmentation mask accurately aligned with the target part. As shown in Tab. I8, PISA's zero-shot part identification ability is comparable to state-of-the-art (SOTA) methods. Additionally, due to PISA's end-to-end advantage, its execution time significantly outperforms others.

| | PISA | ShapeGrasp | LERF-TOGO |
|---|---|---|---|
| Part Selection (%) | 80 | 86 | 82* |
| Time (s) | 2 | 25 | 120 |

Table I8: Comparison of part selection accuracy and execution time. * indicates that LERF-TOGO uses the same categories of objects, but not identical ones.

In Tab. I9, we list all the tasks (Li et al., 2024) evaluated in our case study in the discussion section. In Fig. I17, we showcase some results of our PISA model predicting in a zero-shot manner. It is evident that PISA, trained with our proposed dataset, demonstrates good generalization ability, successfully segmenting unseen parts like plant stems.

It is worth discussing that while the quantitative results shown in the discussion are not superior to ShapeGrasp (Li et al., 2024) and LERF-TOGO (Rashid et al., 2023), the entire real-world dataset contains only 49 tasks. Although LERF-TOGO achieves 6% higher accuracy than us, this difference equates to just 3 images. Moreover, our method is significantly faster than others, and this novel end-to-end prediction approach can be beneficial for real-time robot grasping. Our methods can easily be integrated with existing grasping baselines such as GraspNet (Fang et al., 2020). With our dataset, researchers can focus more on applying segmentation methods to grasping, creating a good bridge between 2D perception and 3D grasping.

## J Distinctions between InstructPart and LISA (Lai et al., 2023).

While both works fall under the category of *reasoning-based segmentation*, the goal, task definition, benchmark scale, and downstream applicability are fundamentally different:

- **Benchmarking Goals and Granularity:** LISA focuses primarily on *object-level scene understanding*, where the objective is to semantically interpret an image and segment an object based on abstract instructions (e.g., "segment the food with the most protein" or "segment the food that is not spicy"). In contrast, our work introduces *task-oriented part-level segmentation*, aiming to understand the affordance and functionality of object components. This finer-grained understanding is essential for practical applications that require actionable perception and reasoning grounded in object structure.

- **Benchmark Scale and Usefulness:** While LISA introduces an important first step toward reasoning-based segmentation, its benchmark contains 1,218 samples, which may be insufficient for a comprehensive evaluation of vision-language models. In contrast, our *InstructPart* benchmark includes 2,400 images, together with 9,600 diverse task instructions, making it more comprehensive and diverse. This enables a more thorough evaluation and offers greater potential for model training and fine-tuning.

- **Novelty and Research Opportunity:** We consider the reasoning-based segmentation task proposed by LISA as a combination of VQA and semantic segmentation—two tasks that have been well explored. However, *task-oriented part understanding* remains significantly under-explored, as discussed in Section 2.1 of our paper. Our work goes further by introducing the use of instructions and affordances to refer to different object parts. This creates a more challenging and novel setting, which we believe will encourage research into part-level reasoning and grounding.

Table I9: **Complete list of tasks for each scene**

| Scene | Tasks |
|---|---|
| Kitchen | 'pick up the grey spoon', 'pick up the teapot', 'scrub the dishes', 'dust the books' |
| Flowers | 'give the daisy', 'give the rose' |
| Mugs | 'pick up the mug', 'pick up the blue mug', 'pick up the grey mug' 'pick up the white mug', 'pick up the teacup' |
| Tools | 'pick up the retractable tape measure', 'pick up the screwdriver', 'cut the wire' 'pick up the soldering iron', 'swing the hammer' |
| Knives | 'cut the bread', 'cut the steak', 'cut the box' |
| Martinis | 'pick up the grey martini glass', 'pick up the green martini glass' |
| Fragile | 'hang the camera', 'wear the blue sunglasses' 'wear the black sunglasses', 'pick up the lightbulb' |
| Cords | 'pick up the power strip', 'plug in the power strip', 'pick up the usb dongle', 'push in the connector' |
| Messy | 'eat the ice cream', 'eat the lollipop', 'eat the red lollipop' |
| Pasta | 'pick up the wine bottle', 'uncork the wine', 'pick up the corkscrew', 'pick up the saucepan', 'open the saucepan' |
| Cleaning | 'pick up the clorox box', 'close the clorox box', 'grab a wet towel' 'pick up the tissue box', 'dispense a tissue' |
| Bottles | 'pick up the meyers cleaning spray', 'open the meyers cleaning spray', 'spray the meyers cleaning spray', 'pick up the purple cleaning spray', 'open the purple cleaning spray', 'spray the purple cleaning spray' |

## K Analysis on Sub-optimal Performance of Existing VLMs on InstructPart

The sub-optimal performance of state-of-the-art VLMs on our benchmark can be attributed to both the lack of task-relevant training data and limitations in current model architectures for part-level understanding and affordance reasoning.

- **Training Data Limitations:**
  Most existing VLMs are not trained with supervision at the part level, nor are they exposed to task-oriented instructions that require grounding specific object components. This leads to a gap in their ability to localize and reason about fine-grained object parts based on functional cues—capabilities that our task explicitly targets. We present two findings to support the claim that current VLMs lack suitable training data:
  - In Section 4.5 (Figures 3–5), we show that many VLMs tend to either segment the entire object or miss the correct regions entirely—indicating difficulty in fine-grained localization.
  - As shown in Appendix D, even simple fine-tuning on our dataset leads to a significant performance boost, suggesting that the models possess latent capability but lack the appropriate supervision signal.

- **Architectural Limitations:**
  Most VLMs use a CLIP-based image encoder, which is optimized for object-level semantic understanding and lacks explicit mechanisms for part-level grounding or affordance reasoning. To address this, we incorporate a DINOv2 vision encoder in our baseline, which better captures part-level correspondences across diverse objects (e.g., the handle of a knife vs. the handle of scissors). As a result, our baseline outperforms state-of-the-art


Figure I17: PISA zero-shot prediction on novel objects. Green masks represent the prediction, and the label below each image highlights the object-part name.

VLMs on the proposed task.

## L Justification for Including ORPS

Referring Expression Segmentation (RES) generally aims to generate segmentation masks from natural language expressions, and our ORPS task can indeed be viewed as a specialized form of RES. However, there are several important distinctions:

- Existing RES tasks primarily focus on using expressions to identify entire entities (e.g., "the woman in the red shirt"). In contrast, ORPS focuses on identifying specific object parts, using a consistent and controlled format: "`[part name] of [object name]`".
- ORPS can be considered the "optimal condition" of TRPS — that is, it strips away complex instruction reasoning and isolates the challenge of part-level visual grounding. This enables us to more precisely understand a model's bottleneck: is it struggling with language reasoning or with part segmentation?
- As shown in Table 2, by comparing the performance gap between ORPS and TRPS:
  - Reasoning segmentation (RS) methods show a smaller drop in performance from ORPS to TRPS, indicating stronger generalization to complex instructions.
  - In contrast, Open-Vocabulary Segmentation (OVS) and Referring Expression Segmentation (RES) baselines show a larger drop, highlighting limited ability to handle task-oriented reasoning.
- This analysis demonstrates that ORPS complements TRPS by offering a controlled setting for part-level grounding, and jointly, they allow us to better characterize the strengths and limitations of different segmentation approaches — especially when comparing models with or without integrated language reasoning.