PISA: Task-Oriented Part Segmentation with Language 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, katia}@andrew.cmu.edu

A Dataset Distribution

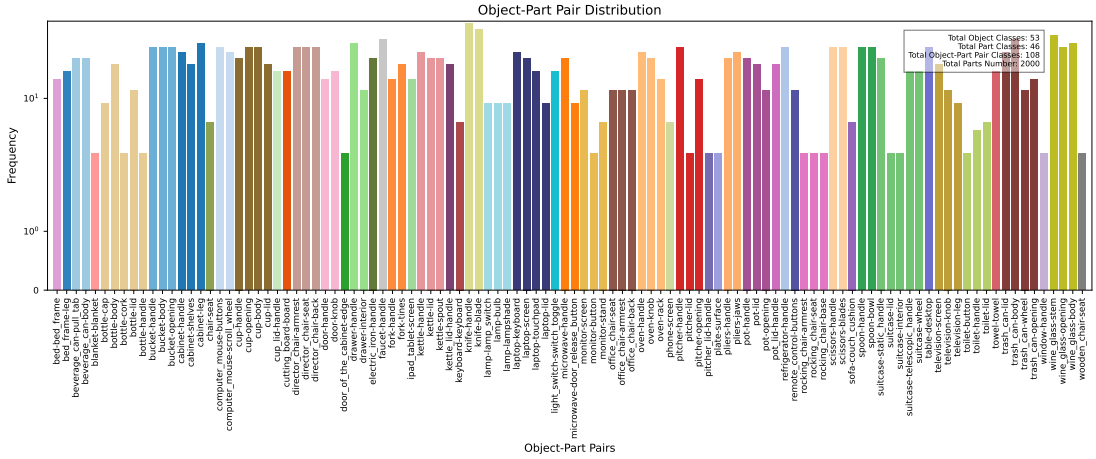

Figure A1: Object-part pair distribution. We collect 2,400 data pieces in total, containing 48 object classes and 44 part classes, constituting 98 different object-part pair classes. The x-axis shows the name of the object-part pairs, and the y-axis shows the frequency of each item. The parts belonging to the same object classes are highlighted with the same color in the bar chart.

We follow ADE20K [3] to provide the distribution of objects and parts within our PISA dataset. As shown in Fig. A1, 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. A2 and Fig. A3, 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. B4 presents two examples of annotations from our PISA 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.

Submitted to the 38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks. Do not distribute.

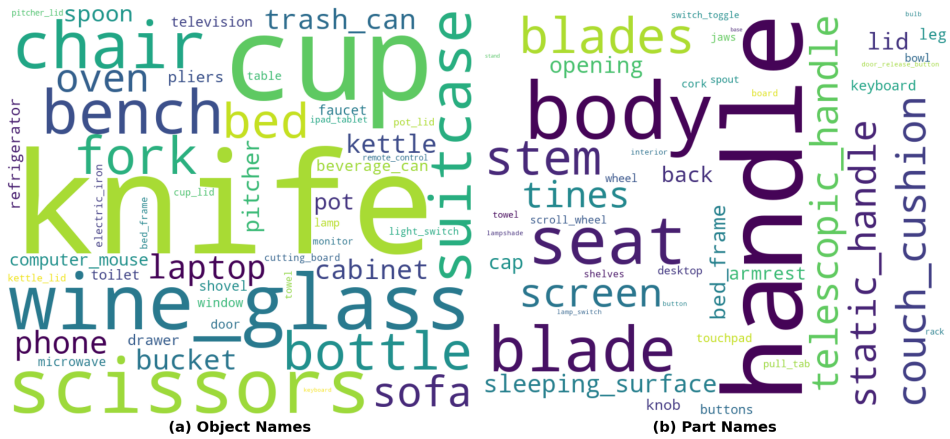

Figure A2: PISA dataset object and part classes. The left part shows the object class names and the right part shows the part class names.

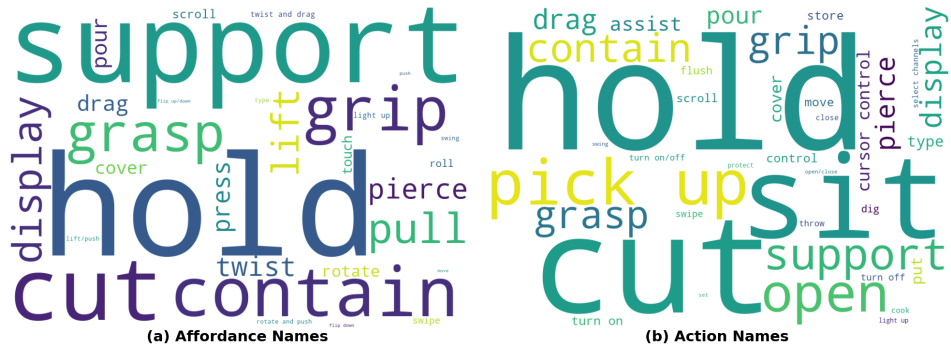

Figure A3: PISA dataset affordance and action categories. The left part shows the affordance names and the right part shows the action names. Specifically, affordances refer to low-level actions performed to a specific part, while actions refer to the high-level function to be achieved.

Besides, in Fig. B5, we provide more examples that contain occlusions and human interactions to showcase the complexity of our dataset.

C PISA Dataset Training Results

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. C6, 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.

D 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. D7. While GPT-4V-based methods deliver clear boundaries, they sometimes select the wrong segments from SAM [1], leading to poor overall performance.

Figure B4: 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 B5: More complex examples in PISA, including occlusions and human-object interactions.

E More Qualitative Results

In Fig. 3 of the main paper, we only include five qualitative results due to space limitations, most of which illustrate cases where the fine-tuned PISA outperforms other methods. In Fig. E8, 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. E10, the pre-trained LISA[2] can better identify desired parts compared to other VLMs. This indicates the evaluation usage of our PISA dataset, where all the advanced VLMs can be evaluated and compared. Furthermore, in Fig. E9, the pre-trained LISA fails to recognize target parts, similar to other VLMs, while both fine-tuned models significantly improve the results.

In Tab. E1, we provide a list containing the name of each sample we evaluate so that their language input can be easily retrieved from our dataset.

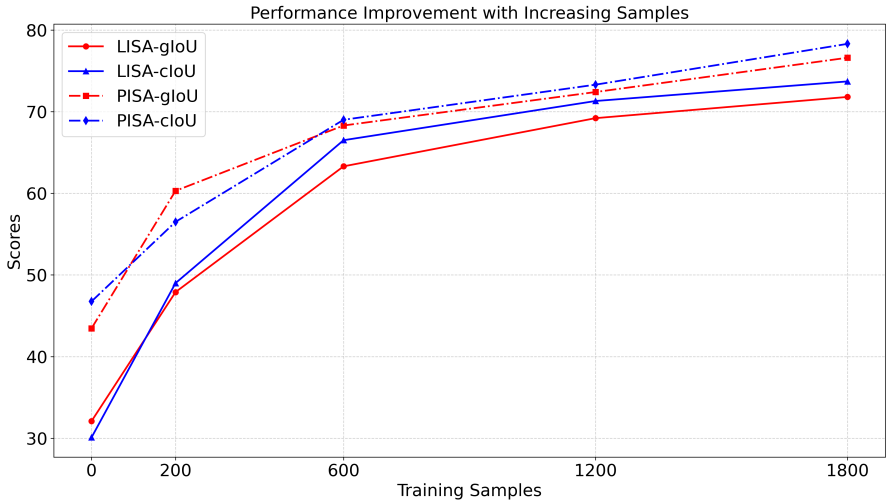

Figure C6: More complex examples in PISA, including occlusions and human-object interactions.

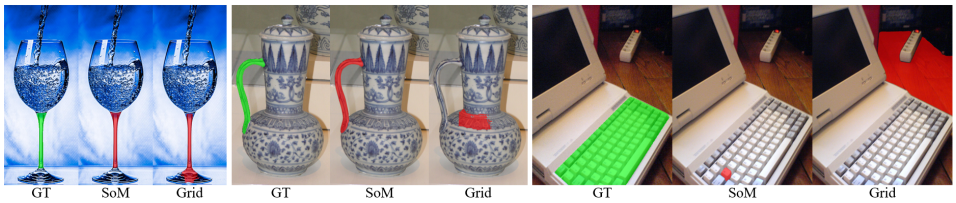

Figure D7: GPT4-V based methods.

Table E1: Index name for samples in Fig. E8, Fig. E9, and Fig. E10.

Fig. E8	**Fig. E9**	**Fig. E10**
1. 1009786005_d4a02fd811_o-faucet-handle 2. 2329134125_8a71be7470_o-kettle-handle 3. 3088942376_8681bb276f_o-spoon-handle 4. cup_000294-cup-handle 5. knife_000911-knife-handle 6. 410044558_6145ff0aaa_o-pot-handle 7. laptop_000445-laptop-keyboard 8. 538210619_c4def94c9b_o-scissors-handle 9. knife_002845-knife-handle 10. knife_000691-knife-handle	1. 4178009615_ed8921d0d1_k-kettle-spout 2. cup_000324-cup-handle 3. bottle_002805-bottle-body 4. knife_000568-knife-handle 5. knife_000953-knife-blade 6. 34465720_f8f20ee31a_c-scissors-handle 7. 381204305_e5e937fccc_h-pitcher-handle 8. bench_001273-bench-seat 9. fork_002954-fork-handle 10. knife_000154-knife-handle 11. shovel_1-shovel-blade 12. suitcase_001098-suitcase-telescopic_handle 13. wine_glass_001774-wine_glass-stem 14. dining_4-chair-seat	1. 2491323916_a05ac3648f_o-knife-handle 2. 4580224808_1194613deb_o-chair-seat 3. 4471021242_b9d855f193_k-bucket-handle 4. 8607578325_25221a7726_h-spoon-handle 5. bench_002898-bench-seat 6. cup_001798-cup-handle 7. cup_002055-cup-handle 8. knife_000530-knife-blade 9. scissors_001402-scissors-handle 10. cup_002062-cup-handle 11. 2939090254_2f01ebed6d_o-computer_mouse-scroll_wheel 12. 6217625873_411169d784_o-laptop-keyboard 13. cup_001104-cup-handle 14. fork_001529-fork-handle

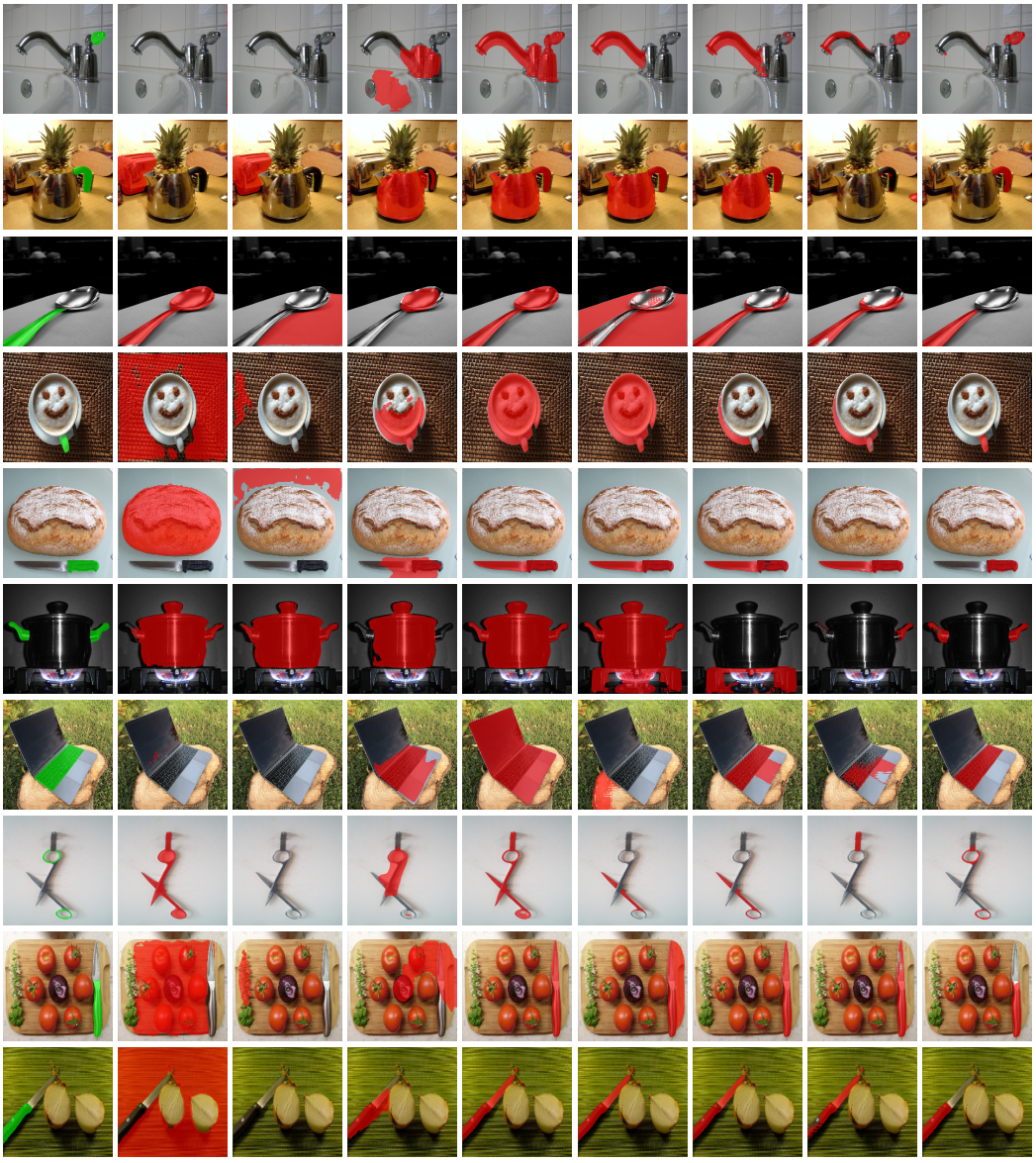

Figure E8: Qualitative comparison of different VLMs and the fine-tuned models. In these examples, the pre-trained LISA falls short of recognizing the correct part. After fine-tuning, PISA shows better potential for part understanding than LISA.

Figure E9: 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 E10: 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.

F More Annotation Samples

In addition to the annotation examples shown in Fig. B4, we include five more annotations for the samples presented in Fig. 3 of the main paper. The listed annotations correspond to the order of the samples shown in Fig. 3.


```
{
    "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"
            ]
        }
    ]
}
```


References

[1] 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. *arXiv preprint arXiv:2304.02643*, 2023. 2

[2] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 3

[3] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 1