Supplementary: Find Any Part in 3D

In the supplementary material, we provide more details and qualitative examples of our data engine output, more qualitative results of FIND3D, additional experimental details and quantitative results, our benchmark annotation protocol, and more discussions.

1. Additional details on the data engine

1.1. Additional data engine annotation examples

We provide additional examples of our data engine annotations, both for high-quality examples in Fig. 4 and lower-quality (but still useful) examples in Fig. 5. Upon manual inspection of 50 randomly sampled objects, we observe 76% high-quality examples. The annotations cover diverse objects and descriptions. For example, the body of a fire extinguisher is referred to both as "body" and "cylinder" from different views. The lower-quality examples are still useful for training – they might not have pronounced parts, or contain partial masks (e.g., the baguette example), but the supervision signal still pushes the point features close to the correct semantic embedding (e.g. bread-related concepts). The low-quality cartoon frog contains both correct and incorrect masks. When learning from millions of such labels, the incorrect labels can be "smoothed out" because it's unlikely that many frogs' bellies are all incorrectly labeled as "bowtie".

1.2. Addressing potential Gemini inconsistencies

Gemini can assign multiple valid names to the same part from different viewpoints (*e.g.* glass and window). The data engine retains all such labels, which are handled by the contrastive objective (discussed in the main text, L.219-243), allowing FIND3D to learn diverse part names. While Gemini occasionally mislabels due to challenging viewpoints, errors due to inconsistency account for 7.6% annotations upon human evaluation of 250 random samples, comparable to the 6% error rate on ImageNet [2].

1.3. Data engine prompts

Fig. 8 shows the prompt we use to obtain object orientations from Gemini. For a given orientation, we render the object in 10 different views, and pass the prompt along with 10 renderings to Gemini. We calculate the percentage of "yes" answers and choose the orientation with the highest "yes" percentage. Fig. 8 also provides some example objects with answers from Gemini. Fig. 9 shows the prompt we use to obtain part names from Gemini, along with some examples.

2. Additional results of FIND3D

2.1. Additional qualitative examples

We provide additional qualitative results of FIND3D in Fig. 6 and Fig. 7. Fig. 6 shows predictions on Objaverse-General from 4 views for each object. Fig. 7 shows predictions on PartObjaverse-Tiny [7] and iPhone photos (reconstructed to 3D via off-the-shelf single-image reconstruction method, Trellis [6])). FIND3D can segment diverse objects and parts, and can generalize to real-world objects, despite being trained on synthetic data.

2.2. Robustness to parts without edges

While the data engine leverages SAM [1], which relies quite heavily on edges, FIND3D, trained on large-scale data, is able to overcome the edge bias. Although SAM occasionally provides imperfect masks due to its edge preference, such data is helpful for training. Parts not delineated by edges (present in 8% of objects upon 100 random inspections) can be: (a) correctly labeled due to slight color variations (pomegranate crown in Fig. 1); (b) partially labeled (octopus tentacle); (c) not labeled (jacket sleeve). Partial

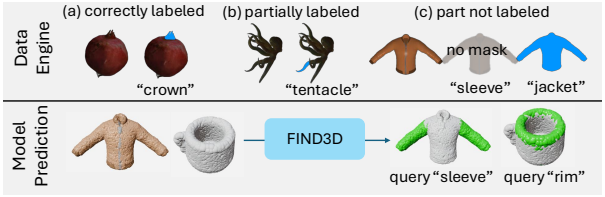


Figure 1. Parts without edges

labels in case (b) are useful at scale with contrastive training (main text, L.227-232); case (c) is also helpful since the holistic mask (*e.g.*, jacket) pushes the semantic features close to clothing-related concepts. As shown at the bottom of Fig. 1, FIND3D can segment out parts not delineated by edges, such as the sleeve of a jacket with the same color, or the rim of a monochromatic mug.

2.3. Robustness to fine-grained parts

We test FIND3D on the finer-granularity parts from PartNet-Level-3. As mentioned in [5], such parts can be challenging to open-world methods. Despite this, we observe good results with FIND3D even for finer parts in Fig. 2. For visualization purposes only, we remove the front part of the lamp shader to show the segmentation.

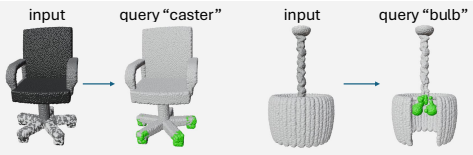


Figure 2. Fine-grained results on PartNet-Level-3

2.4. Additional quantitative results

In Tables 2,3,4 of the main paper, in order to evaluate all methods on the exact same data, we had to report results on subsets of ShapeNet-Part and PartNet-E because methods like PartSLIP++ and OpenMask3D are slow and infeasible to evaluate on the full test sets (*e.g.*, OpenMask3D would take 628 hours on PartNet-E). Here we provide full-set results for methods that are feasible for full-set evaluation in Tab. 2 and Tab. 3. The ranking of methods on the full sets and the subsets are the same. The subset indices for ShapeNet-Part can be found at `model/evaluation/benchmark/benchmark_reproducibility/shapenetpart/subset_idxs.json` of https://github.com/ziqi-ma/Find3D/tree/main, and indices for PartNet-E can be found at `model/evaluation/benchmark/benchmark_reproducibility/shapenetpart/subsetidxs.json`. The random rotations used for evaluation are saved in the same folders.

ShapeNet-Part. Tab. 1 compares all methods with various prompts, orientations, and data sources (ShapeNet-Part vs. ShapeNetPart-V2, a benchmark of the same object classes as ShapeNet-Part but sourced from Objaverse that we constructed, similar to ImageNetV2 [3]). PointCLIPV2 is trained on this dataset, and other methods are evaluated zero-shot. FIND3D performs the best in 8 out of 9 configurations, despite being zero-shot. While Tab. 1 reports metrics on the subset of ShapeNet-Part so that all methods can be evaluated strictly on the same dataset, for methods that are fast enough to evaluate on the full test set (FIND3D and PointCLIPV2), we also report the full-set evaluation results in Tab. 2. The full-set metrics are very close to the subset metrics. On the full set, we also see that FIND3D performs better in 5 out of 6 settings.

On both the full set and the subset, FIND3D, despite being zero-shot on this dataset, is the best-performing method in all configurations except for one—the canonical orientation with test-time top-k prompt searching. In this setting, PointCLIPV2, a method trained on this dataset and designed with test-time prompt searching in mind, performs slightly better. We note that this searching takes over an hour on an A100, which is unrealistic to perform in real applications. Our method is not designed for test-time prompt searching but clearly outperforms all baselines when doing direct inference.

PartNet-E. Tab. 3 shows results on PartNet-E, both on the subset (for all methods) and on the full set (for methods that are fast enough to evaluate on the full set). PartSLIP++, trained on this dataset, achieves the highest performance with the "`{part}`" prompts, yet is very sensitive to prompt variation. We note that PartSLIP++ also releases category-specific checkpoints, but we use the cross-category checkpoint for fairness of comparison. This dataset is more challenging for our method because many objects contain small parts that are not geometrically or colorfully prominent, such as buttons on a surface with the same color. Nevertheless, we see our method to be more robust to rotation and prompt variation, and clearly outperforms the other baselines that are not trained on this dataset. Furthermore, PartSLIP++ is a slow 2D-3D aggregation method, taking up to 3 minutes per object. Our method is over 30× faster.

3. Benchmark Annotation Protocol

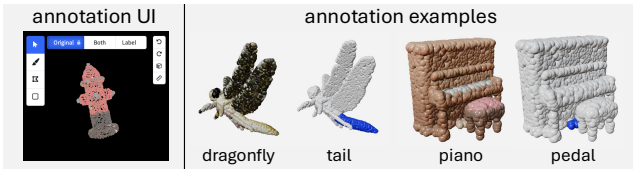


Figure 3. Benchmark annotation UI and examples

The Objaverse-General benchmark is annotated by humans from scratch using Segments.ai with a freeform paintbrush tool. Rather than enforcing a rigid taxonomy or granularity level, we intentionally allow semantic diversity in the annotations, which reflects the variability of the real world. The annotation guidelines are: (1) identify named semantic parts that do not overlap; (2) points that cannot be easily named can be unlabeled. A second round of human review is performed to assess annotation quality. Objects flagged as incorrect are re-labeled. As shown in Fig. 3, the annotations contain different granularity levels, including fine-grained parts like the piano pedal, and parts not delineated by edges, like the dragonfly tail.

4. Additional discussions

4.1. Definition of parts

Ambiguity of granularity is a key challenge in part segmentation. Our definition of object part is any part that occurs in the common vocabulary. An open part vocabulary naturally encompasses many granularities (arm vs. hand) without requiring an explicit taxonomy. It can also handle other ambiguities, such as naming a part by function, material, etc. (window vs. glass), which are difficult to capture with a single taxonomy. Additionally, shared names in language can signal part similarities across categories (leg of chair and human both imply a thin, supportive structure).

4.2. Discussion on architecture vs. training data

FIND3D's performance comes from both the large training dataset and an architecture that enables learning at scale. The baselines cannot directly scale up to our dataset due to their design, such as extensive inference-time search (PointCLIPV2 [9] would take over 32 days on an A100 on our dataset), category-specific fine-tuning (PartSLIP++ [8]), or the training-free design (OpenMask3D [4]). In contrast, FIND3D is a feedforward model that naturally scales with the amount of training data available. In our scaling analysis, when we train on the same data scale as PointCLIPv2 [9] and PartSLIP++ [8] (16 & 45 categories, respectively), we achieve 4%, 5% higher mIoU (Fig.6, main paper).

4.3. Extension to meshes

FIND3D starts with the point cloud representation due to their general applicability (*e.g.*, robotic sensors, smartphone RGBD captures), and modeling scalability (shown by FIND3D). Meshes contain more geometric cues due to their connectivity, but are more expensive to process. Investigating whether similar scaling behavior applies to training FIND3D with additional mesh information is an interesting future direction.

5. Code

Code is provided at https://github.com/ziqi-ma/Find3D/tree/main.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1

[2] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021. 1

[3] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 2

[4] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

[5] Anh Thai, Weiyao Wang, Hao Tang, Stefan Stojanov, James M Rehg, and Matt Feiszli. $3\times$ 2: 3d object part segmentation by 2d semantic correspondences. In *European Conference on Computer Vision*, pages 149–166. Springer, 2024. 2

[6] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 1

[7] Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024. 1

[8] Yuchen Zhou, Jiayuan Gu, Xuanlin Li, Minghua Liu, Yunhao Fang, and Hao Su. Partslip++: Enhancing low-shot 3d part segmentation via multi-view instance segmentation and maximum likelihood estimation. *arXiv preprint arXiv:2312.03015*, 2023. 3

[9] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2639–2650, 2023. 3

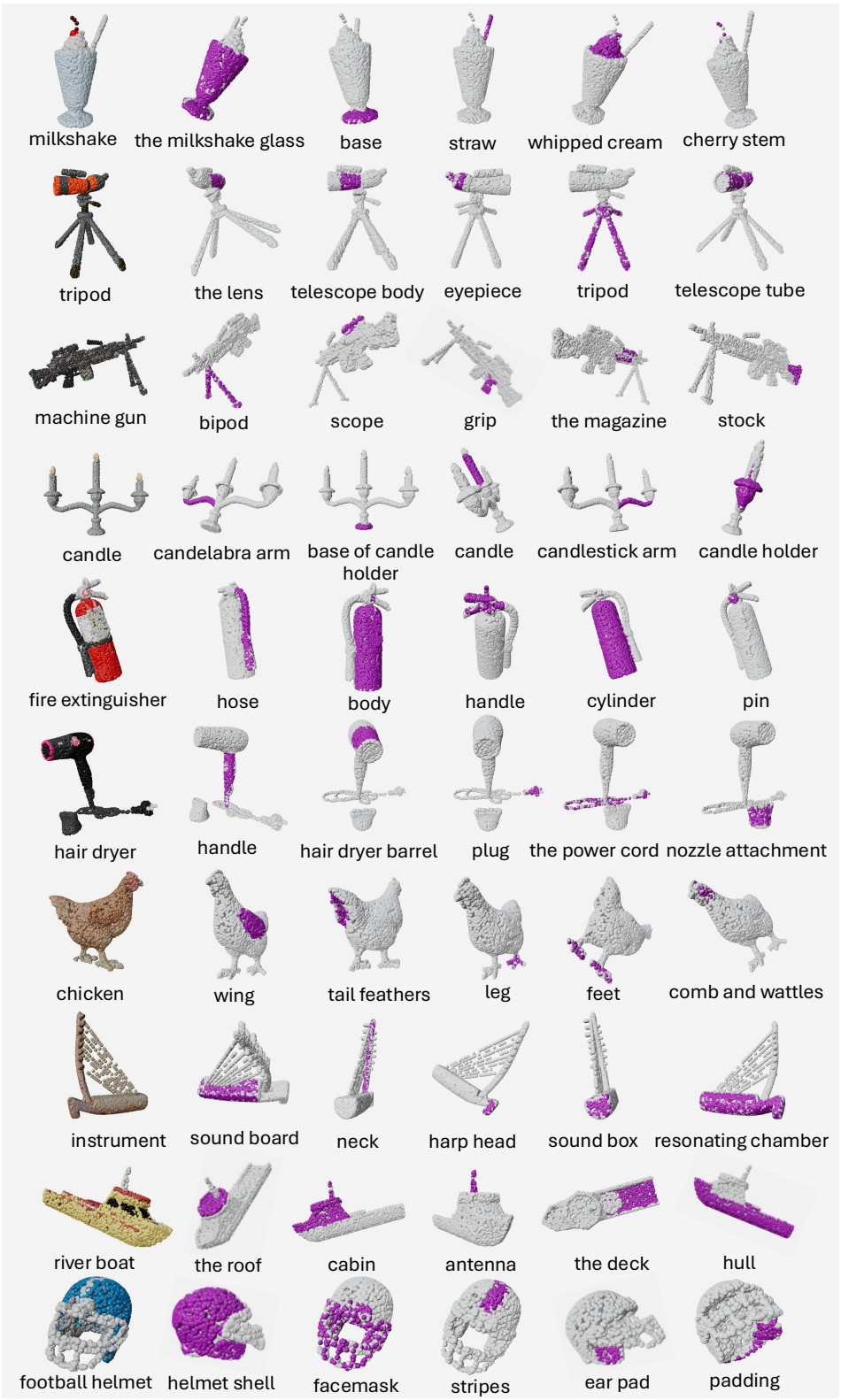


Figure 4. High-quality examples of data engine annotations. The LVIS label (from Objaverse) is shown below each input object. Our data engine annotates diverse objects and parts, including multiple captions for the same parts, such as "candelabra arm" and "candlestick arm", and multiple levels of granularity, such as "helmet shell" and "ear pad".

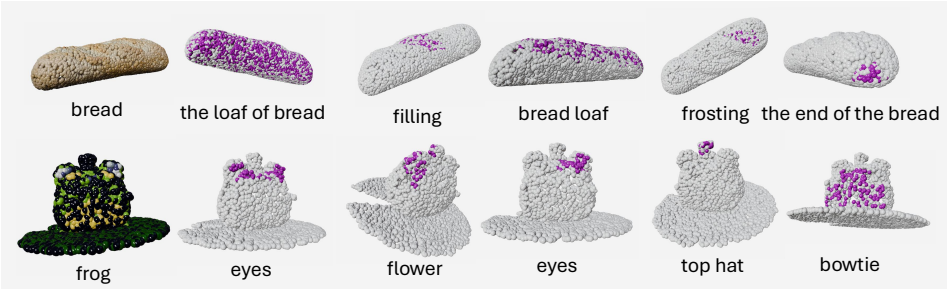


Figure 5. Lower-quality examples of data engine annotations. The LVIS label (from Objaverse) is shown below each input object. Some objects do not have pronounced parts, such as the baguette, and get partial part labels due to texture/lighting change on surfaces. Some objects are low quality, such as the cartoon frog, which results in incorrect labels.

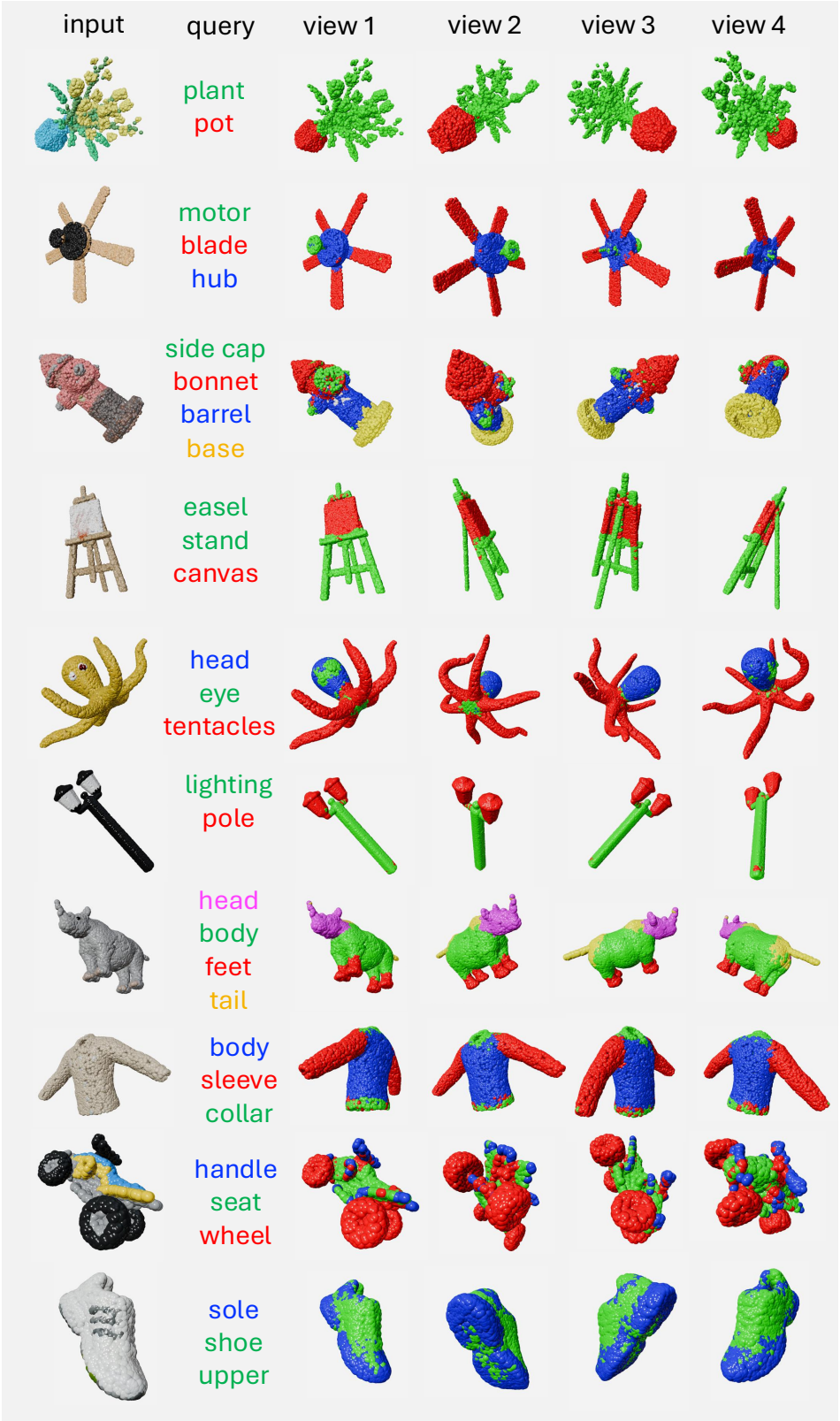


Figure 6. Multiple views of FIND3D predictions on Objaverse-General examples.

Figure 7. Multiple views of FIND3D predictions on PartObjaverse-Tiny examples and iPhone photos (reconstructed to 3D with off-the-shelf method).

mIoU(%)	Canonical Orientation			Rotated			Objaverse-ShapeNetPart		
	top-k	{part} of a {object}	{part}	top-k	{part} of a {object}	{part}	top-k	{part} of a {object}	{part}
PointCLIPV2	**48.666**	16.912	20.215	26.111	16.878	18.193	21.177	15.136	17.110
PartSLIP++	–	1.432	6.460	–	0.937	6.034	–	1.542	11.622
OpenMask3D	–	8.938	10.373	–	6.748	14.556	–	15.870	13.768
FIND3D (Ours)	43.613	**28.386**	**24.085**	**43.781**	**29.637**	**23.712**	**50.002**	**42.151**	**30.018**

Table 1. Detailed results on ShapeNet-Part subset. Shaded cells mean the method is trained on the same dataset (expected higher than white cells), and white cells mean zero-shot evaluation. We evaluate different orientations, query prompts, and data domains (ShapeNet-Part vs. ShapeNetPart-V2). We evaluate on 3 types of prompts: "{part} of a {object}", "{part}", and top-k. Top-k prompt reproduces the PointCLIPV2 paper, which runs an iterative search over $1400 \times n_{\text{parts}}$ prompts per object category to choose the best query text prompts. For fairness of comparison, we follow the same procedure to get top-k prompt metrics, although our method is not designed with prompt searching in mind, and it is not realistic to conduct this slow ($>$ 1 hour on A100) searching process at inference time. Our method, despite being zero-shot on this dataset, has the best performance in 8 out of 9 configurations—all configurations except for the canonical orientation with top-k prompt searching.

mIoU(%)	Canonical Orientation			Rotated		
	top-k	{part} of a {object}	{part}	top-k	{part} of a {object}	{part}
PointCLIPV2	**48.472**	17.471	20.157	26.337	17.034	18.021
FIND3D (Ours)	41.517	**28.532**	**23.569**	**42.734**	**29.966**	**23.794**

Table 2. Detailed results on ShapeNet-Part full test set. Shaded cells mean the method is trained on the same dataset (expected higher than white cells), and white cells mean zero-shot evaluation. PartSLIP2 and OpenMask3D are too slow and thus infeasible to evaluate on the full test set. The metrics are very close to the subset results in the previous table. Our method, despite being zero-shot on this dataset, has the best performance in 5 out of 6 configurations—all configurations except for the canonical orientation with top-k prompt searching. This searching process takes over an hour on an A100 and our method is not designed for test-time prompt searching.

mIoU(%)	Canonical Orientation				Rotated			
	Full		Subset		Full		Subset	
	{part} of a {object}	{part}	{part} of a {object}	{part}	{part} of a {object}	{part}	{part} of a {object}	{part}
PointCLIPV2	11.619	9.647	11.275	9.700	10.943	10.261	10.317	10.216
PartSLIP++	–	–	5.123	**32.705**	–	–	3.866	**23.033**
OpenMask3D	–	–	12.538	11.242	–	–	11.933	11.673
FIND3D (Ours)	**17.143**	**16.211**	**16.861**	16.384	**17.703**	**16.819**	**17.620**	17.164

Table 3. Detailed results on PartNet-E test set. Shaded cells mean the method is trained on the same dataset (expected higher than white cells), and white cells mean zero-shot evaluation. Cells with "-" denote that the method is too slow to be evaluated on the full test set. We evaluate with 2 types of prompts: "{part} of a {object}" and "{part}". PartSLIP++ achieves the highest performance with the "{part}" prompts, yet the performance drops 84% when we vary the query prompt. This dataset is more challenging for our method due to the sparsity of labels and the presence of small parts that are not geometrically or colorfully prominent (*e.g.*, buttons on a surface with the same color). Nevertheless, our method is more robust to rotation and prompt variation, and clearly outperforms the other baselines not trained on this dataset.

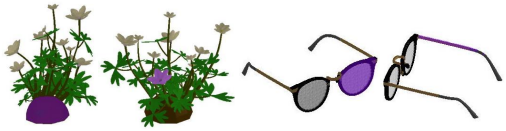
Prompt: For each image, is the object in an orientation that is usually seen? Please answer yes or no for each image.

Gemini: Yes Yes No No

Figure 8. The prompt used to query Gemini for object orientation. The car and the Christmas tree are in common orientations (and thus will yield higher-quality annotations), whereas the camel and the parasol are not.

Prompt: What is the name of the part of the object that is masked out as purple? If you cannot find the part or are unsure, say unknown. Please only output the part name as one word or phrase.

Gemini:



flower pot flower lens temple

spout handle gemstone ring

Figure 9. The prompt used to query Gemini for object part names. We show 2 example masks from different views for a potted plant, a pair of glasses, a teapot, and a ring.