OpenVoxel: Training-Free Grouping and Captioning Voxels for Open-Vocabulary 3D Scene Understanding

Supplementary Material

Figure A1. **Detail of the mask merging process.**

In this supplementary material, we provide additional method details in Sec. 1 and more results of semantic segmentation in Sec. 2, ablation studies in Sec. 4, visualization in Sec. 5. Finally, we discuss about our limitation in Sec. 6.

1. Details

Merging groups. In Sect. **??**, we mentioned merging small segmentation masks by re-prompting SAM2 [12] to reduce noise. Here, we visualize this process for a better understanding. As depicted in Fig. **??** and Fig. A1, the *"hand sculpture"* instance is separated into two different groups in M_2^{proj} since the SAM2 mask M_1 treats them as two different segment, where one group covers almost the entire hand and the other just representing a finger tip. To encourage each of our groups to be more instance-wise, we double-check if any two groups should be merged by prompting all groups that are observable under the current view with SAM2 again. Taking the hand sculpture under view ξ_2 as example in Fig. A1, we create the point prompts for SAM2 by treating several pixels sampled from the group in M_2^{proj} of interest as positive prompts, and also select several other groups and treat their center pixel in M_2^{proj} as negative prompts. As for the mask prompt, we treat the entire group of interest in M_2^{proj} as a positive prompt (mask value: 20), all the other known groups as a negative prompt (mask value:-20), and the rest of the region is treated as unknown (mask value:0). After using this setting as prompt for SAM2, if the output mask of one group (e.g., the finger tip in Fig. A1) is almost lie inside another group's mask (e.g., over 90% inside the hand sculpture), then we merge the smaller group into the larger group accordingly (i.e., merge the finger tip into the hand sculpture). As shown in our ablation study Table **??**, a slight improvement is observed by including this merging strategy.

MLLM prompt. In Sect. **??** and **??**, we leverage an MLLM model of QWen3-VL-8B to conduct canonical captioning, prompt refinement, and target retrieving. The system prompts are as List 1, List 1, and List 1.

We note that we do not spend much effort exploring different kinds of system prompt design. We simply use ChatGPT by describing our task with some examples and ask it to provide system prompts that suit Qwen-VL [1, 2, 15] models, and the same prompts are shared for both RES and OVS tasks. Therefore, these system prompts may not be the optimal ones, and users still have a chance to improve the performance by using our pipeline and simply changing system prompts.

Implementation. We now elaborate on the implementation details. Since our method is totally training-free after having the pre-trained SVR model of each scene, reducing the process time of the following processes (i.e., grouping, captioning, and retrieving) is a crucial problem. To achieve this goal, we do not always go through all the images from the training set for each scene in practice. Instead, taking the LeRF [7] dataset and corresponding subsets as an example, we uniformly sample the processed views to make sure the total processed view of each scene does not exceed 150. Also, we found that since the merging process requires additional SAM2 execution, conducting this merging process for all views would slow down the inference time. Hence, we conduct the merging (re-prompting SAM2) process for each scene per 1 to 5 steps to speed up the inference. Furthermore, when prompting the captioning model (i.e., DAM [9]), we only sample 8 frame-mask pairs (pad to 8 pairs if not enough) for each group to reduce the visual tokens for the model, making sure that the inference is fast. Also, for the usage of MLLM for Canonical Captioning, Query Refinement, and Text-to-text retrieving, we "*DO NOT*" provide any visual example for the MLLM as in-context examples since the inference time would be slowed down by doing so (although we acknowledge that conducting these information might bring better performance, we leave this as a future direction to explore the balance between adding visual demonstrations and inference time).

Computation Consumption. We show the required time and GPU memory for each component of our OpenVoxel using a single RTX 5090 GPU in Table A1. We also include the consumption of the original SVR reconstruction for reference. Note that among all processes of our Open-

```
CANONICAL_CAPTION_SYSTEM_PROMPT= """You are a detail-focused visual caption
refiner for open-vocabulary segmentation and referring grounding.
INPUTS
- A short video where ONE region is masked (highlighted); the outside area
is darkened.
- A rough caption from another model (may be incorrect or misleading).
SCOPE
- Describe ONLY what lies INSIDE the masked region across frames.
- Never name or infer unmasked neighbors as the subject.
- Be factual; do not guess hidden details.
- If the original caption conflicts with the visual evidence, IGNORE it and
correct the errors.
REWRITE GOAL
Produce a precise, natural description with a clear class noun and
discriminative details that is easy to match with open-vocabulary queries.
CORE RULES
1) Class noun: Replace vague words ("object/thing/item/surface") with a
concrete class or fine-grained subtype.
2) Part-of decision (strict):
   - Use "part of <larger object>" ONLY if ALL are true:
     a) Visible physical continuity/attachment within the mask (seam/stitch/
     joint/hinge/fastener or continuous material/geometry),
     b) The region is an intrinsic component,
     c) The larger object's category is visibly identifiable.
   - Otherwise DO NOT use "part of". Prefer placement instead.
3) Placement vs background:
   - Use view-independent placement for surfaces/containers (\on table",
   \inside pouch", \on plate", \on shelf", \in tray", etc.).
   - If the region is background material/texture (floor/wall/ceiling/ground),
   begin with \background: <material/surface>".
4) Printed-content rule (critical):
   - Scan ALL frames within the mask for printed text, logos, labels,
   or characters.
   - If ANY are visible, include at least one cue:
     • Text: transcribe exact readable tokens (keep visible case/punctuation).
     If partial, include the visible substring.
     • Character/graphic: if identity is uncertain,
     describe visual attributes neutrally (e.g., \purple cartoon dinosaur");
     do not guess names.
5) Detail quota:
   - Include at least FOUR distinct cues chosen from: color; material;
   texture/pattern; shape/geometry; subtype/model; visible text/logo/
   printed character; state/condition; function/affordance;
   part-of (only if allowed); placement/relation (max 2).
6) Language hygiene:
   - View-independent wording only (no left/right/front/top; no camera terms).
   - Do not mention the highlight/red dot.
   - Forbidden words: object, thing, item (and similar generic fillers).
OUTPUT FORMAT (canonical; must be strictly followed)
- EXACTLY ONE line, 12{20 words, comma-separated phrases, no period.
- Start with the SUBJECT noun.
- **Order of phrases (strict):**
  1) <category noun> (table, chair, bottle, pouch, human character, cat,
  dog, rabbit, camera, spoon, door handle, floor, wall, etc.)
  2) <appearance details>  (color/material/texture/pattern/shape/
  subtype/text/logo)
  3) <function/affordance or part-of>  (\part of <larger object>"
  only if rule 2 allows; otherwise a concise function/affordance like
  \resealable pouch", \pour spout", \grip handle")
  4) <placement/relation>  (on/in/inside/attached to/against/between;
  max 2 relations)
- If unsure between \part of" and placement, choose placement.
- For background regions: \background: <material/surface>,
<appearance details>, <(optional) function if any>, <placement/relation>".
STRICTNESS
The form of the output must strictly follow the rules above and the
ordered template:
<category noun (no color)>,(comma here)
<appearance details (color here)><function/affordance or part-of>
<placement/relation>.
The original caption may be wrong; rely on visual evidence to correct it.
"""
```

Listing A1. System prompt of Canonical Captioning.

```
QUERY_REPHRASE_SYSTEM_PROMPT = """You are a helpful assistant.
You rewrite a short OVS query into ONE short canonical phrase
using ONLY the provided image and the raw query.
SCOPE
- Inputs:
(a) scene_map (JSON list of candidate objects with their
coordinates and caption),
(b) query (description about an object visible in the image),
(c) view_image.
- Keep the result SHORT and human-judgable from the query mainly.
- you must NOT include any spatial relations in the output if not explicitly
mentioned in the query.
CANONICAL FORM
- Output exactly ONE short phrase in the form:
<class noun> <appearance> <placement?>
- 2 to 6 words, lowercase, spaces only, no punctuation.
- class noun: singular, most specific common name that is visually supported
(e.g., "rubber duck", "paper bag").
- appearance: brief, image-supported attributes (color/material/texture/
state/text/logo/shape). If unsure, keep generic
(e.g., "plastic-like", "transparent").
- placement (OPTIONAL): view-INDEPENDENT, simple scene phrase
(e.g., "on table", "in bowl", "on shelf", "in bag").
Avoid left/right/front/behind/above/below.
REPHRASE PROTOCOL (follow strictly)
- Be conservative if uncertain; never hallucinate specifics you cannot see.
- Examples:
  • "toy car on the table" → "car on table"
  • "banana" → "banana" (no assumed color)
  • Materials: "plastic bag"→"plastic-like bag"; "nori"→"seaweed";
  "glass cup"→"transparent cup"; "porcelain"→"ceramic"
  • Common words: "gummy"→"gummy candy"; "ribeye beef"→"piece of meat";
  "toy car"→"car"; "stuffed bear"→"teddy bear";
    "paper napkin"→"napkin"; "kamaboko"→"small piece with pink swirl";
    "rubber duck with a bouy"→"rubber duck with pink lei"
  • Character names→descriptions:
  "pikachu"→"yellow character with long ears and possibly red cheek";
  "jake"→"yellow cartoon character big eyes slim legs";
    "miffy" → "rabbit character, possibly wearing garment";
    "waldo"→"character red-white striped shirt";
    "hello kitty"→"white cartoon cat red bow"
  • Brands→generic: "lays"→"potato chips"; "coca-cola"→"can";
  "nike shoes"→"sports shoes";
  "tesla door handle"→"metalic object, look like door handle"
  • Ambiguous placements: "in the bowl"/"on the plate"/"inside the pouch"
  → "in container"/"on surface"/"in bag"
OUTPUT (STRICT)
Return ONE JSON line only (no extra text, no code fences, no reasoning):
{"canonical": "<clear class noun>, (you must include this comma after
<clear class noun>) <appearance (color)>,
<placement (ONLY IF contained in query text)>"}
CONSTRAINTS
- No chain-of-thought or explanations.
- Do not use any information that cannot plausibly be inferred from
the image + query alone.
- You MUST NOT use ambiguous noun like "object", "thing", "item", "stuff",
"part", "area", "region", "section", "portion", "background", "foreground",
"surface", "area", "area of interest", etc.
- If the class noun is a general category
(e.g., "container", "food", "furniture"),
you MUST add more specific appearance to clarify.
- If the query does not contain any placement info,
DO NOT add any placement in the output.
"""
```

Listing A2. System prompt of Query Refinement.

Voxel, the highest peak memory is at the same order as the original SVR, and the accumulated time of all processes is much shorter, verifying the applicability and efficiency.

Table A1. **Detailed runtime and GPU memory cost of each stage of our OpenVoxel.** "Figurines" scene for example.

	SVR	Grouping	Caption	Refine	Inference
Mem (MiB)	9649	11380	8654	18322	18852
Runtime (sec)	305	64	33	24	<1

2. Semantic Segmentation

2.1. Approach

Different from OVS and RES, the task of semantic segmentation usually has a customized list of class candidates for each dataset. Therefore, instead of retrieving the matched groups for each class, we conduct semantic segmentation for OpenVoxel by choosing the best-matched class for each group.

2.2. Dataset and implementation details

Dataset. Following OpenGaussian [16], we conduct semantic segmentation on 10 different scenes on the Scannet [5] dataset. Each scene is represented as colored point clouds, with ground truth images and depth maps provided. Following the official setting, we conduct a 19 class semantic segmentation in our experiments.

Implementation details. We note that in OpenGaussian, ground truth point clouds are directly utilized as initialization for the 3DGS, and they deactivate all the merging and splitting processes so that a perfect geometry alignment is naturally obtained for evaluation. However, it is not easy for our backbone (i.e., SVR [14]) to have such an initialization.

```
    SYSTEM_PROMPT_RETRIEVE = """You retrieve all matching targets using
    scene_map + view_image (optional) + a canonical short phrase.

INPUTS
1) scene_map: JSON of candidate voxel groups with fields:
   - id (integer, unique)
   - caption (short description; copy EXACTLY in output)
   - center: WORLD coordinates (use only for view-independent relations:
   near/far/between/closest/farthest)
   - optional: bbox/size/group/category
2) view_image (optional): one image for the current query instance
(targets may be occluded or off-frame).
3) canonical: the short canonical phrase from Stage 1
(e.g., "rubber duck, yellow", "paper bag, on table").

POLICY (caption-first, occlusion-robust)
- Primary signal: scene_map CAPTIONS (semantic match to the canonical phrase;
allow common synonyms/hypernyms).
- Ignore view-dependent relations (left/right/front/behind).
Use WORLD coords ONLY for near/far/between/closest/farthest
if such words appear.
- One real object may be split across multiple voxel groups (ids)
that are spatially adjacent and semantically consistent.
  If so, RETURN ALL ids for that instance.
- If multiple separate instances match the canonical phrase,
RETURN the best aligned one.
- Secondary signal: view_image (if provided) only to veto
obvious mismatches when visible; do NOT penalize occlusion.

INTERNAL STEPS (do not reveal):
1) Match captions to the canonical phrase|prioritize
exact/synonym class match, then attribute alignment.
2) Merge adjacent voxel groups that describe the same instance
(spatially close in WORLD coordinates and semantically consistent).
3) If canonical phrase includes near/far/between/closest/farthest,
apply these using WORLD centers/bboxes over the matched set.
4) Use the image (if provided) only to down-weight
clear visual contradictions when visible (do not discard due to occlusion).
5) Finalize ids and copy their captions EXACTLY from scene_map.

OUTPUT (STRICT)
Return EXACTLY one JSON line|no extra text,
no code fences, no reasoning:
{"ids": [<int>, ...], "captions": ["<EXACT caption>", ...]}
Rules:
- Include at least one id for every matching instance
(multi-instance allowed, but in most case only one).
- Sort ids ascending within each instance; overall order is arbitrary.
- Captions must be copied EXACTLY from scene_map, same order as ids.
- You cannot return empty ids or ids that are not in scene_map.
- If borderline: you MAY add
{"candidates": [{"id": a}, {"id": b}]}

CONSTRAINTS
- No chain-of-thought or explanations.
- Do not paraphrase any caption in the "captions" array;
copy exactly from scene_map.
- Use WORLD coordinates only for view-independent relations;
do not use image axes for left/right/front/behind.
"""
```

Listing A3. System prompt of target retrieval.

Table A2. **Quantitative evaluation on ScanNet semantic segmentation of 19 classes.**

Method	Uses GT Point	mIoU	mAcc
LEGaussian [13]	✓	3.8	10.9
LangSplat [11]	✓	3.8	9.1
OpenGaussian [16]	✓	24.7	41.5
Ours (Nearest)	-	30.0	41.1
Ours (Majority of 25-NN)	-	31.3	42.1
Ours (Majority of 50-NN)	-	**31.6**	**42.3**

To have a better geometry for the Scannet dataset, we utilize the provided depth map to guide the pre-training process of the SVR model.

2.3. Evaluation protocols and results

Evaluation protocols. Since we do not use the ground truth points for pre-training the SVR model, the constructed voxel number of each scene is very different from the number of ground truth points. Typically, the number of ground truth point clouds is about 50K to 350K, but the number of our voxels are about 5M to 10M. Therefore, we conduct several different protocols for evaluations to better showcase our OpenVoxel: **(1) Nearest**,**(2) Majority of 25-NN**, and **(3) Majority of 50-NN**. The **Nearest** protocol means that for each point in the ground truth point cloud, we find the spatially nearest voxel and take the voxel's class ID (obtained as described in 2.1) as our prediction; **Majority of 25-NN** and **Majority of 50-NN** indicates for each point in ground truth, we find 25 or 50 spatially nearest voxels and treat the majority of their labels as our predictions. After defining our prediction for each point, the rest are totally the same as the evaluation pipeline as proposed in OpenGaussian.

Table A3. **Results (mIoU) on 3D-OVS [33] dataset.**

	bed	bench	room	lawn	sofa	avg.
ObjectGS [76]	98	**96.4**	95.1	97.2	95.4	96.42
ReferSplat [15]	93.2	94.8	94.6	96.5	85.6	92.94
Ours	**98**	95.8	**97.4**	**97.4**	**96.5**	**97.02**

Table A4. **Ablation studies on different segmentation models for RES task on Ref-LeRF [6] subset.**

Method	Segmentation Model	mIoU
A	SAM	30.5
B	SAM2	42.4

Results. The results are shown in Table A2. We can see that even without using the ground truth points as prior, our OpenVoxel still outperforms all baselines in terms of mIoU, and is comparable in mAcc. This shows the potential of OpenVoxel on diverse tasks instead of just OVS and RES for being a training-free approach.

3. Open-Vocabulary Segmentation

To further verify our OpenVoxel, we conduct additional OVS experiments on the 3D-OVS [10] dataset, which contains five different scenes with 6 to 7 classes in each scene. The results are shown in Table A3, where we can see that although all current SOTAs are producing mIoU with over 90 %, our OpenVoxel still outperforms them, showing the robustness across different datasets.

4. Ablation study.

Being a training-free approach, it is essential to investigate how different prior models affect the performance of our OpenVoxel. Therefore, we conduct ablation studies in three main prior models: the segmentation model for grouping, the captioning model for generating raw captions, the MLLM for canonical captioning, query refinement, and target retrieval.

Table A5. **Ablation studies on different models for captioning of RES task on Ref-LeRF [6] subset.**

Method	Captioning Model	mIoU
A	Osprey (Yuan, et al, 2024)	29.3
B	Qwen3-VL-8B-Instruct	33.3
C	DAM [9]	42.4

Table A6. **Ablation studies on different MLLMs for RES task on Ref-LeRF [6] subset.**

Method	MLLM	mIoU
A	Qwen2.5-VL-7B-Instruct	23.4
B	Qwen3-VL-2B-Instruct	10.0
C	Qwen3-VL-4B-Instruct	35.6
D	Qwen3-VL-8B-Instruct	42.4

Different segmentation model. Table A4 shows our OpenVoxel using different versions of SAM [8, 12] as a segmentation prior model for our grouping stage for RES task. In our experiments, we observe that SAM tends to segment small fragments that are over-detailed, and therefore, the grouping results are slightly noisier than our original version using SAM2. As a result, we can see that there is about 10% performance drop on the RES task.

Different captioning model. Table A5 shows our OpenVoxel using different captioning model to obtain the original caption for each group on the RES task. For the setting using the Osprey (Yuan, et al, 2024) captioning model, we caption each frame and then ask Qwen3-VL-8B-Instruct model to summarize them into one sentence since Osprey is not suitable for taking video as input. As for the setting using Qwen3-VL-8B-Instruct as captioning model, we direct bypass the DAM captioning stage and ask Qwen3-VL-8B-Instruct to generate caption purely from the visual input (with darkened background and red dot as visual prompt) since it is not trained for taking separated video-mask pair as input. From Table A5 we can see that although Qwen3-VL-8B-Instruct is not trained specially for captioning masked region captioning task, it can still produce reasonable results that are feasible for the RES task. As for Osprey, although it is a captioning specialized model, the per-frame prediction property lead to inconsistent caption for the same group from different views, confusing the Qwen3-VL-8B-Instruct model for summarization and hindering the performance of the RES task. However, we note that using either captioning model achieves better mIoU than ReferSplat [6] (29.2% for the original reported number and 24.5% for our reproduced results) on the RES task, showing the robustness of our designed pipeline.

Different MLLM. Table A6 shows the results of our OpenVoxel using different MLLMs during the canonical scene map construction and the inference stage with the totally same system prompts and user prompts. Since we do not specially design different prompt for each model, we observe that the Qwen3-VL-2B-Instruct is incapable of canonicalize the captions generated from DAM. Instead, it tends to repeat some of the words in the original caption as the refined caption. As a result, the incorrect refined captions are hard for the Qwen3-VL-2B-Instruct model to locate the correct target object in the inference stage, leading to a catastrophic 9.98% mIoU. In contrast, the newest and largest model for this ablation, Qwen3-VL-8B-Instruct is obviously performing best.

5. Qualitative results

Referring Segmentation. We provide the qualitative results of the other two scenes (i.e., teatime and kitchen) in Ref-LeRF [6] subset for RES task in Fig. A2 and Fig. A3. Similar to our observation in Sect. **??**, for the first column (i.e., "A smooth container placed next to the sheep doll, near the apple" as query) in Fig. A2, ReferSplat [6] tends to capture only part of the query (i.e., "container") and hence segmenting both the coffe mug and the glass of tea, neglecting other spatial clues in the query. Similarly, for the last column in Fig. A3 with "A countertop with a vibrant yellow color provides plenty of space for preparing cooking ingredients." as query, ReferSplat only capture the color information of "vibrant yellow" and segment both the counter top and the wall. In contrast, our OpenVoxel successfully locates the ideal target in both cases.

We additionally showcase the qualitative results of RES on the "*Teatime*" scene using our created natural language that are not included in the Ref-LeRF subset as query to demonstrate the capability of our OpenVoxel compared with ReferSplat [6]. We can see that since ReferSplat requires training on all objects using human annotated sentences, it is not able to locate objects that are not annotated during their training. In contrast, our training-free approach is not depending on the training annotations at all and is able to locate the correct targets. We note that for the last two columns where the input query contains view dependent descriptions, although our OpenVoxel retrieve two targets instead of the only one matched, the results are still including the correct target, showing the potential capability of solving view-specific tasks

To better showcase the multi-view consistency of our RES result, we demonstrate rendered views of the same query from different views in Fig. A5. From this figure, we can see that the segmentation masks are multi-view consistent and accurate according to the input query, showing the robustness of our OpenVoxel.

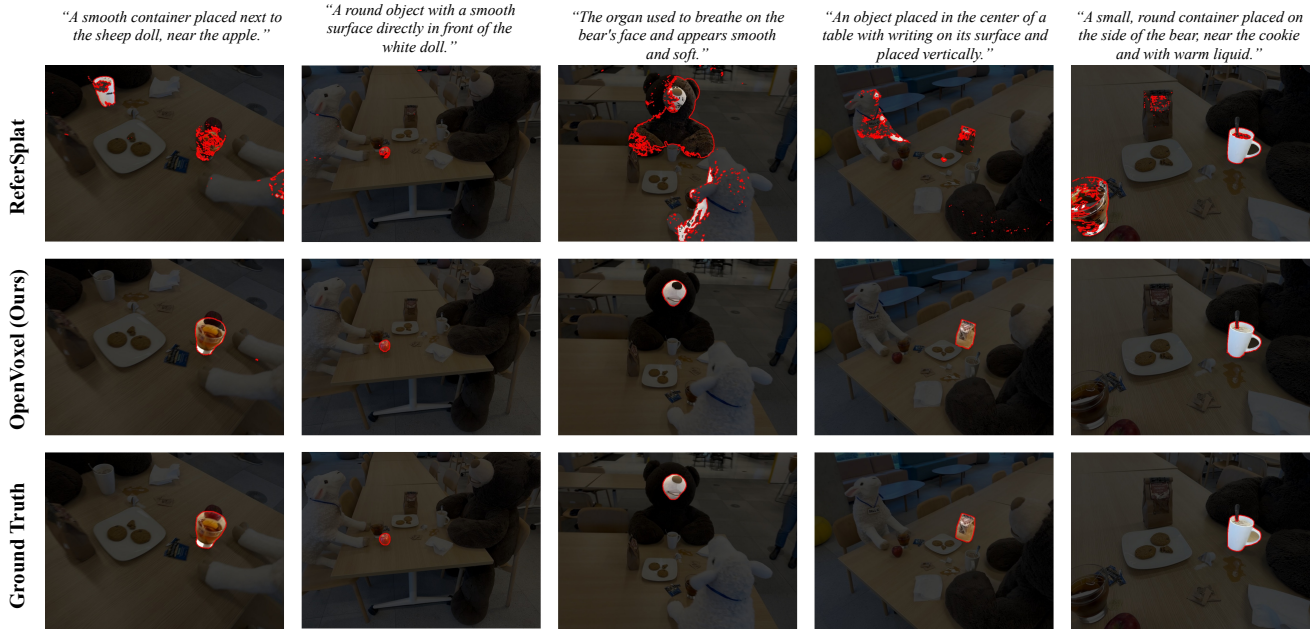

Figure A2. **Qualitative results of RES task on the *Teatime* scene.**

Figure A3. **Qualitative results of RES task on the *Kitchen* scene.**

6. Discussions and Limitations.

We now discuss the potential limitation of our OpenVoxel. As a training-free approach, the grouping process of OpenVoxel is relatively sensitive to parameters comparing to the end-to-end generalizable ones [4]. As described in Sect. 1, the sampling rate of frames and merging frequency are customized for each scene. And since how well-separated for different instances largely affects the performance for both OVS and RES (semantic segmentation is less affected), the

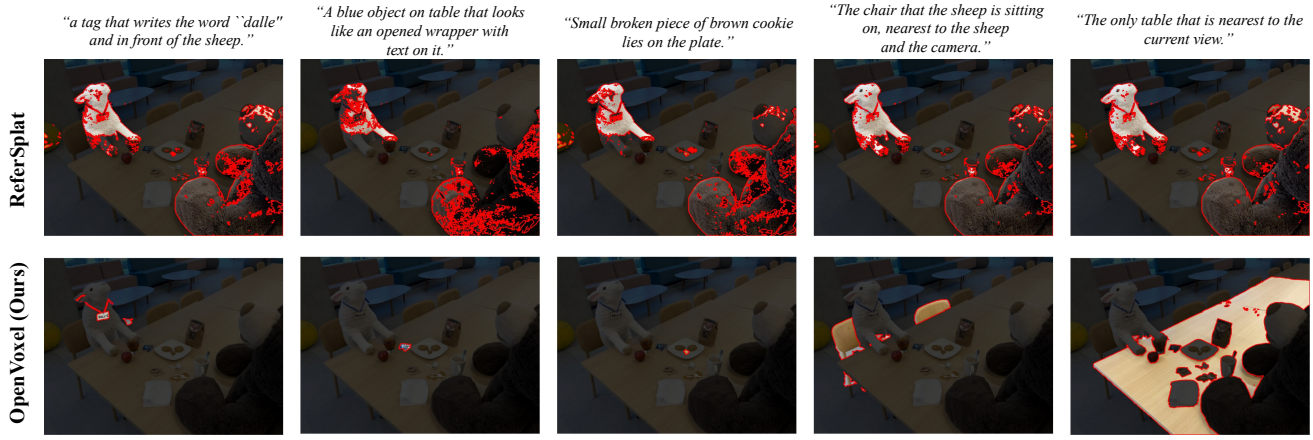

Figure A4. **Qualitative results of RES task on the *Teatime* scene with other queries.** Note that these queries are created additionally and all of them are not appeared in the original annotations from Ref-LeRF subset (so there are no ground truth mask for them). We can see that ReferSplat [6] struggles to recognize unseen target objects even in the same scene it is optimized, showing that it tends to overfit on annotated objects from the dataset.

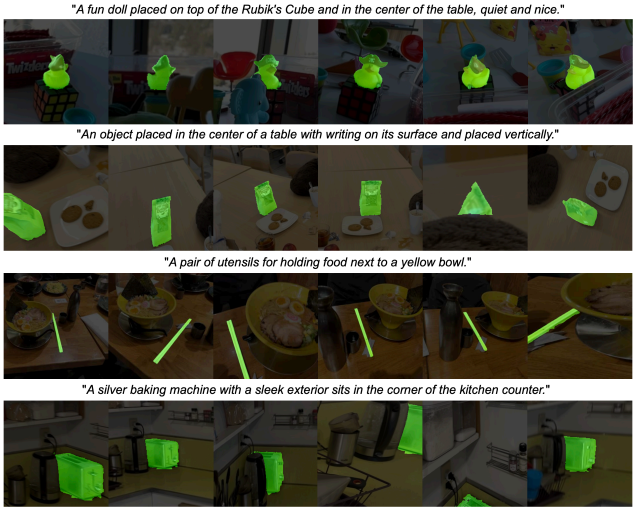

Figure A5. **Consistency of rendered RES masks.**

SAM2 parameters are needed to be adjust carefully. However, we note that other optimization-based grouping methods [17, 18] also share similar limitation, as their results heavily rely on the quality of view-consistent video segmentation models such as DEVA [3]. Fortunately, our heuristic of sampling one frame per 3-5 frames and conduct merging once per 3 steps generally work well. In case the result is unsatisfactory, our work is totally training-free so it would be easy to adjust the parameters and re-run the grouping process with a tolerable time (about 1 minute per-scene, taking Ref-LeRF subset as example.)

Also, since we conduct the instance-level grouping process before captioning/retrieval, if the user gives a query to indicate some part of a larger object (e.g., flash light of the camera in Fig. **??**), our OpenVoxel would still segment the whole object (i.e., the whole camera) since the small parts of the same object is bundled together. One possible solution to solve this issue is to curate the groups as small as possible while still keeping them semantically reasonable (requires 2D segmentation maps that includes those small part). Additionally, instead of just build the Scene Map S by storing center locations of each group, construct a complex scene graph for all the groups to indicate the spatial relations (e.g., on top of, between) or ownership (e.g., belongs to, part of) explicitly. We truly believes that this would help improving the performance and robustness of our OpenVoxel and leave it as a possible future direction.

Another limitation of our OpenVoxel lies in the usage of MLLM models. As shown in the Sect. 1, we turn off the chain-of-thought process of the MLLM model for fast inference (less than one second per query). However, by doing so the capability of reasoning for the MLLM is also limited, and hence both the canonicalized captions and the retrieving process are having room to be improved. Also, as shown in Table A6, the results of using different open-source MLLM differs. We believe that if better open-source MLLM appear with faster thinking/reasoning ability, our OpenVoxel can be benefited from them.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 2

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin

Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2

[3] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 7

[4] Zi-Ting Chou, Sheng-Yu Huang, I Liu, Yu-Chiang Frank Wang, et al. Gsnerf: Generalizable semantic neural radiance fields with enhanced 3d scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 6

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[6] Shuting He, Guangquan Jie, Changshuo Wang, Yun Zhou, Shuming Hu, Guanbin Li, and Henghui Ding. Refersplat: Referring segmentation in 3d gaussian splatting. *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. 4, 5, 7

[7] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 2

[8] 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 International Conference on Computer Vision (ICCV)*, 2023. 5

[9] Long Lian, Yifan Ding, Yunhao Ge, Sifei Liu, Hanzi Mao, Boyi Li, Marco Pavone, Ming-Yu Liu, Trevor Darrell, Adam Yala, et al. Describe anything: Detailed localized image and video captioning. *arXiv preprint arXiv:2504.16072*, 2025. 2, 5

[10] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 4

[11] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[12] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 5

[13] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[14] Cheng Sun, Jaesung Choe, Charles Loop, Wei-Chiu Ma, and Yu-Chiang Frank Wang. Sparse voxels rasterization: Real-time high-fidelity radiance field rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[15] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2

[16] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3, 4

[17] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 7

[18] Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. Objectgs: Object-aware scene reconstruction and scene understanding via gaussian splatting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2025. 7