# OV3D-CG: Open-vocabulary 3D Instance Segmentation with Contextual Guidance

# Supplementary Material

## 1. Implementation Details

This section details the implementation components of our approach, including the class-agnostic proposal generation and the semantic reasoning modules.

### 1.1. Class-agnostic Proposal Module

#### 1.1.1. Pre-trained 3D Instance Segmenter

To generate class-agnostic instance masks using the 3D pre-trained segmenter, we follow the setup described by [11]. Specifically, we adopt the Transformer-based Mask3D architecture [9], which is designed for open-vocabulary 3D instance segmentation. In our experiments, we employ the Mask3D model trained on the ScanNet200 dataset [8] for instance segmentation, keeping the mask proposal module weights frozen. This model produces binary instance masks that are not associated with specific class labels, aligning with the open-vocabulary paradigm. To ensure an adequate number of mask proposals, we set the number of queries to 150. Additionally, we apply the DBSCAN clustering algorithm [2] to subdivide non-contiguous instance masks into spatially contiguous clusters.

#### 1.1.2. SAM-Guided 3D Instance Segmenter

To generate class-agnostic instance masks using the SAM-based segmenter, we employ Semantic-SAM [5] as the 2D foundation model following the method proposed by [12]. To extract superpoints from the original point cloud, we adopt the method described in [3]. During the progressive region-growing process, we set the affinity threshold values to [0.9, 0.8, 0.7, 0.6, 0.5] for the ScanNet200 dataset to ensure optimal segmentation performance.

### 1.2. Semantic Reasoning Module

#### 1.2.1. Best Viewpoint Selection

After generating class-agnostic 3D instances, we determine the optimal projection viewpoint by evaluating the visibility of each instance from multiple camera poses. Leveraging the intrinsic matrix and pose transformation, we project the 3D instance mask onto 2D image coordinates. Visibility is then quantified by counting the number of visible pixels, where pixels are considered invisible if their depth values deviate beyond a small threshold $\delta$, designed to mitigate sensor noise. For the experiments on both ScanNet200 [8] and Replica [10] datasets, we set $\delta$ to 0.2.

To maximize instance clarity and robustness, we select the viewpoint that results in the highest number of visible pixels. While we explored using multiple views and aggregating predictions, the performance gain was marginal (less than 1% AP), and the computational cost increased significantly. Therefore, we adopt a single-view strategy as a practical trade-off between accuracy and efficiency.

#### 1.2.2. MLLM Chain-of-Thought Reasoning

We utilize multimodal large language models (MLLMs) for semantic reasoning, specifically Gemini 1.5 Flash, Gemini 1.5 Flash-8B, and LLaVA-OneVision-7B. The first two models are accessed via API calls through Gemini's online service, while the latter is deployed locally to support inference. To ensure the validity of the generated labels during evaluation, we employ the ViT-B/32 CLIP [7] text encoder to match the MLLM-generated output to the closest predefined category.

To enhance the model's reasoning capabilities, we design a Chain-of-Thought (CoT) prompt as described in the main manuscript. However, during dataset evaluations, the predefined label set may limit the open-ended nature of the model's responses. To address this, we tailor the CoT prompts specifically for each dataset, guiding the model to generate outputs that align with the predefined categories while still leveraging contextual reasoning.

Despite tailored prompting, large models are prone to hallucination, potentially generating labels that deviate from the predefined set. To mitigate this, we employ CLIP to verify the semantic validity of the predicted labels. Given CLIP's input length limitation, we also restrict the large model's output descriptions to a maximum of 50 words to ensure clarity and relevance.

The designed prompts for various context-aware representations are presented in Tab. 1. For our ablation study, we also develop prompts without CoT, which are shown in Tab. 2.

### 1.3. Datasets

To balance efficiency and performance, we sample data at fixed intervals across datasets. Specifically, for the ScanNet200 [8] and Replica [10] datasets, we extract one frame out of every 20 frames, including RGB images, depth frames, and pose information. This sampling strategy results in a 5% data selection rate.

Following previous works [6, 11, 12], we exclude instances labeled as "wall" and "floor" from the ScanNet200 dataset to focus on object-centric analysis. This exclusion ensures that our evaluation emphasizes the recognition and segmentation of distinct object instances.

## 2. Experiments

In this section, we evaluate the performance of our proposed approach through extensive experiments on both synthetic benchmarks and real-world robotic platforms.

### 2.1. Text-driven 3D Instance Segmentation

To clarify the process of text-driven 3D instance segmentation as shown in Fig. 4 in the main manuscript, we provide additional explanation regarding how textual queries interact with our framework.

Our method first performs class-agnostic 3D instance segmentation, generating a set of instance masks. For each detected instance, a detailed caption is produced through CoT reasoning. This caption not only describes the instance itself but also includes information about its surrounding context. These instance-level captions effectively serve as natural language descriptions that uniquely characterize each object in the scene.

To support text-driven segmentation, user-provided query texts are matched against the generated instance captions using a text similarity measure (e.g., CLIP-based or other embedding comparisons). The instance whose caption best aligns with the query is identified as the target, and its corresponding 3D mask is returned as the segmentation result.

This approach allows the model to ground free-form textual descriptions to specific 3D instances, without requiring fixed category labels or predefined object classes. It enables flexible and intuitive interaction with the 3D scene through natural language.

### 2.2. Real-world Implementation

We constructed a dedicated embodied environment to evaluate our segmentation and navigation algorithms. For testing and validation, we employed the LoCoBot robot as our experimental platform. The LoCoBot is equipped with an RGB-D camera (Intel RealSense D435) for depth perception, an IMU for motion tracking, and wheel encoders for odometry. The robot is powered by an onboard computer and supports wireless communication for remote operation. Our experimental platform and environment are illustrated in Fig. 1.

Figure 1. Real-world experiment platform.

To enable 3D environmental reconstruction, we deployed the RTAB-Map SLAM algorithm [4] on the ROS platform, utilizing the RGB-D camera to generate a dense point cloud representation of the environment. Following this reconstruction, we applied our proposed OV3D-CG method for instance segmentation. OV3D-CG assigns open-vocabulary semantic labels and descriptions to each object, enhancing the semantic understanding of the scene.

This enriched semantic information serves as valuable input for downstream navigation tasks, enabling more context-aware and adaptive robotic behavior.

### 2.3. Additional Experiments

#### 2.3.1. Runtime Analysis

To provide a comprehensive evaluation of our method's computational efficiency, we conducted a detailed runtime analysis on the ScanNet200 dataset. The runtime was measured as the average processing time in seconds per scene ($s/scene$) across the entire validation set, comparing our approach with representative CLIP-based baselines.

Our analysis reveals that the primary computational bottleneck for current 3D open-vocabulary segmentation methods—including ours—is the 3D-to-2D modality alignment step ($T_{\mathbf{3D}to\mathbf{2D}}$). This step's significant time consumption is the main reason that existing methods struggle to achieve real-time performance.

Although our inference time is higher, it constitutes only a fraction of the total runtime ($T_{\mathbf{total}}$), which, as detailed in

| Representation | Caption Prompt | Identification Prompt |
|---|---|---|
| bounding box | "Describe the object inside the red bounding box. Consider the surrounding background context as well. Limit the description to 50 words." | "According to the description [PROMPT RESPONSE], identify what is the object from the available labels. Here is the list of possible labels: [LABELS]." |
| landmarks | "There are two images given. The first one is the original image. The second one is the original image with some green landmarks. Describe the object that the green landmarks mainly covers. Limit the description to 50 words." | |
| SAM mask | "There are two images given. The first one is the original image. The second one is the original image with a green mask. Describe the object that the green mask mainly covers. Limit the description to 50 words." | |

Table 1. CoT prompts for different context-aware representations on validation set.

| Representation | Identification Prompt |
|---|---|
| bounding box | "Identify what is the object inside the red bounding box from the available labels. Here is the list of possible labels: [LABELS]." |
| landmarks | "There are two images given. The first one is the original image. The second one is the original image with some green landmarks. Identify the object that the green landmarks mainly cover from the available labels [LABELS]." |
| SAM mask | "There are two images given. The first one is the original image. The second one is the original image with a green mask. Identify the object that the green mask mainly covers from the available labels [LABELS]." |

Table 2. Prompts without CoT for different context-aware representations on validation set.

Tab. 3, remains dominated by the $T_{\mathbf{3D}to\mathbf{2D}}$ process. Therefore, the runtime overhead from our method is modest when considering the entire pipeline. We consider this performance trade-off acceptable given the substantial gains in flexibility, interpretability, and rich reasoning capabilities that our method provides.

Additionally, we specifically evaluated the runtime of using multiple SAM iterations for mask generation. This process is highly efficient, requiring only $\approx 0.4$ seconds to segment 10 images on average, confirming that its impact on the total runtime is negligible.

| Method | $T_{\mathbf{3Dmask}}$ | $T_{\mathbf{3D}to\mathbf{2D}}$ | $T_{\mathbf{infer}}$ | $T_{\mathbf{total}}$ | AP |
|---|---|---|---|---|---|
| OpenMask3D | 59 | 336 | 10 | 405 | 15.4 |
| Open3DIS | 87 | 415 | 12 | 514 | 23.7 |
| Ours | 95 | 375 | 120 | 590 | 25.4 |

Table 3. Detailed runtime analysis of OV-3DIS methods on the **ScanNet200** dataset.

### 2.3.2. Challenging Task Evaluation

We evaluated our approach on a particularly challenging task: functionality segmentation, using the SceneFun3D [1] dataset with ground truth masks to better evaluate semantic reasoning. The results demonstrate that our context-aware MLLM method significantly outperforms the CLIP-based baseline, particularly on small, functionally complex objects, as shown in Tab. 4. These findings highlight the strength of contextual reasoning and show that our approach excels not only in general tasks but also in complex, functionally demanding scenarios.

| Method | AP | Rot. | K.P. | T.P. | H.P. | P.P. | H.T. | F.P. | P.I. | Unp. |
|---|---|---|---|---|---|---|---|---|---|---|
| CLIP-based | 11.7 | 0.9 | 6.9 | 0.0 | 14.3 | 35.3 | **20.1** | 0.0 | 27.8 | 0.0 |
| **Ours** | **22.2** | **9.7** | **37.0** | **10.0** | **22.3** | **29.3** | 0.9 | **50.9** | **34.9** | **5.2** |

Table 4. Functionality segmentation results on the **SceneFun3D** dataset.

### 2.3.3. Qualitative Results

Our method demonstrates strong OV-3DIS capabilities on the ScanNet200 [8] and Replica [10] datasets. These datasets are characterized by a diverse range of scenes, encompassing both common and rare objects. Notably, our method excels in querying instance-level objects based on various attributes, such as color, shape, material, position, affordance, and state.

Compared to other CLIP-based methods, our approach is particularly effective in handling long-text descriptions, significantly enhancing the flexibility and expressiveness of our open-vocabulary capabilities. This advantage proves especially valuable when identifying uncommon or ambiguous objects, where our method leverages environmental context to improve semantic reasoning and ensure accurate categorization. Notably, several open-vocabulary cases demonstrating this capability are presented in Fig. 2-9, highlighting our method's ability to interpret complex descriptions and identify challenging object instances.

Beyond instance classification, our method provides detailed descriptions of the identified objects, enabling a richer and more comprehensive understanding of the scene's contents.

Furthermore, we demonstrate the practical utility of our method in real-world object-driven navigation scenarios, as illustrated in Fig. 10, where complex object queries are accurately interpreted and used to guide goal-oriented navigation behaviors.

## References


[1] Alexandros Delitzas, Ayça Takmaz, Federico Tombari, Robert W. Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14531–14542, 2024. 3

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Knowledge Discovery and Data Mining*, pages 226–231, 1996. 1

[3] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59:167–181, 2004. 1

[4] Mathieu Labbé and François Michaud. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36:416 – 446, 2018. 2

[5] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint*, arXiv:2307.04767, 2023. 1

[6] Phuc D. A. Nguyen, T.D. Ngo, Chuang Gan, Evangelos Kalogerakis, Anh Dat Tran, Cuong Pham, and Khoi Nguyen. Open3DIS: Open-Vocabulary 3D Instance Segmentation with 2D Mask Guidance. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4018–4028, 2023. 2

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 1

[8] David Rozenberszki, Or Litany, and Angela Dai. Language-Grounded Indoor 3D Semantic Segmentation in the Wild. *European Conference on Computer Vision (ECCV)*, 13693: 125–141, 2022. 1, 2, 3

[9] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. *International Conference on Robotics and Automation.*, pages 8216–8223, 2022. 1

[10] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, et al. The Replica Dataset: A Digital Replica of Indoor Spaces. *arXiv preprint*, arXiv:1906.05797, 2019. 1, 2, 3

[11] Ayca Takmaz, Elisabetta Fedele, Robert Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 68367–68390, 2024. 1, 2

[12] Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. SAI3D: Segment any Instance in 3D Scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3292–3302, 2023. 1, 2

**Prompt**: *"A blue pillow on the bed"*

Figure 2. OV-3DIS result based on color and position prompt.

**Prompt**: *"A heptagonal table"*

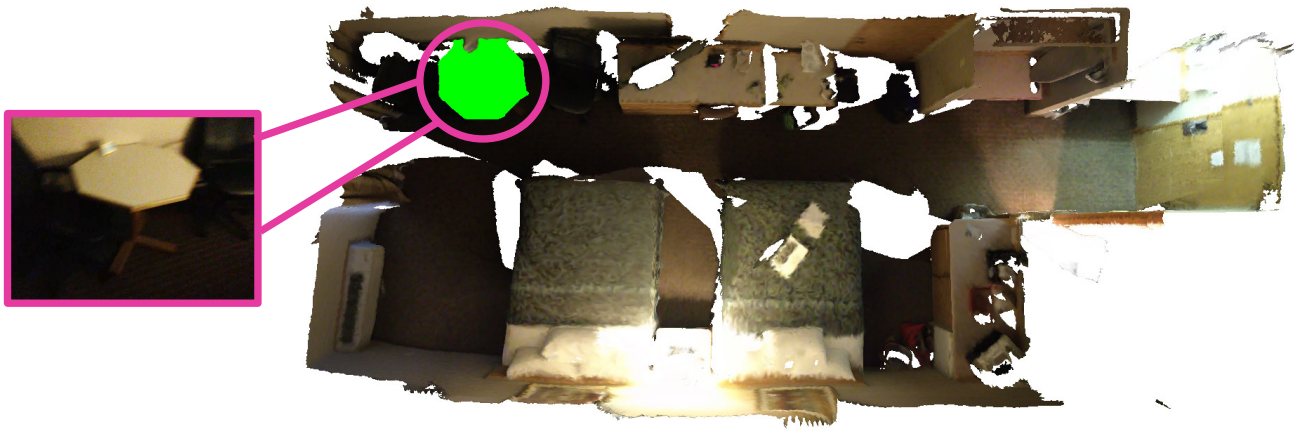

Figure 3. OV-3DIS result based on shape prompt.

**Prompt**: *"A device that can freeze food"*

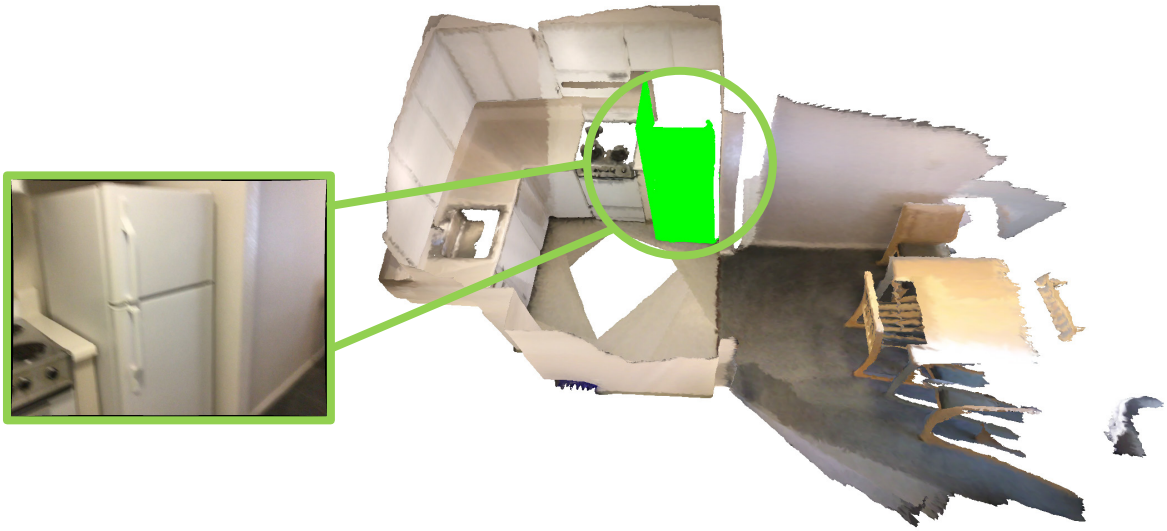

Figure 4. OV-3DIS result based on affordance prompt.

**Prompt**: *"Items that can be used for drinking water"*

Figure 5. OV-3DIS result based on affordance prompt.

**Prompt**: *"The trashcan with a plastic bag on it"*

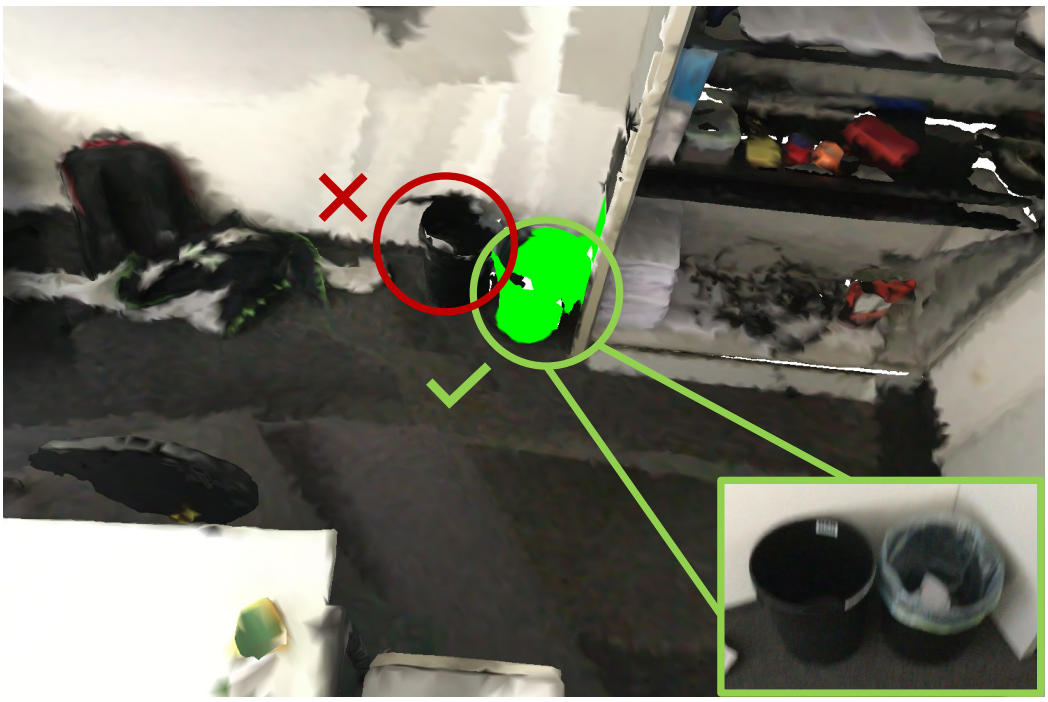

Figure 6. OV-3DIS result based on state prompt.

**Prompt**: *"The bag on the floor"*

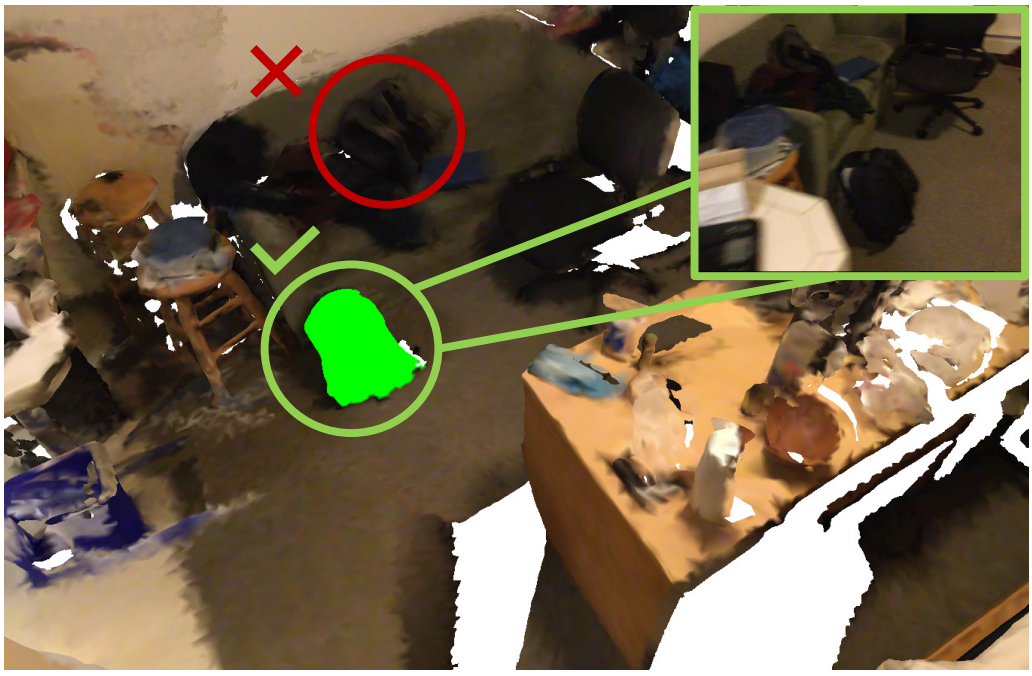

Figure 7. OV-3DIS result based on position prompt.

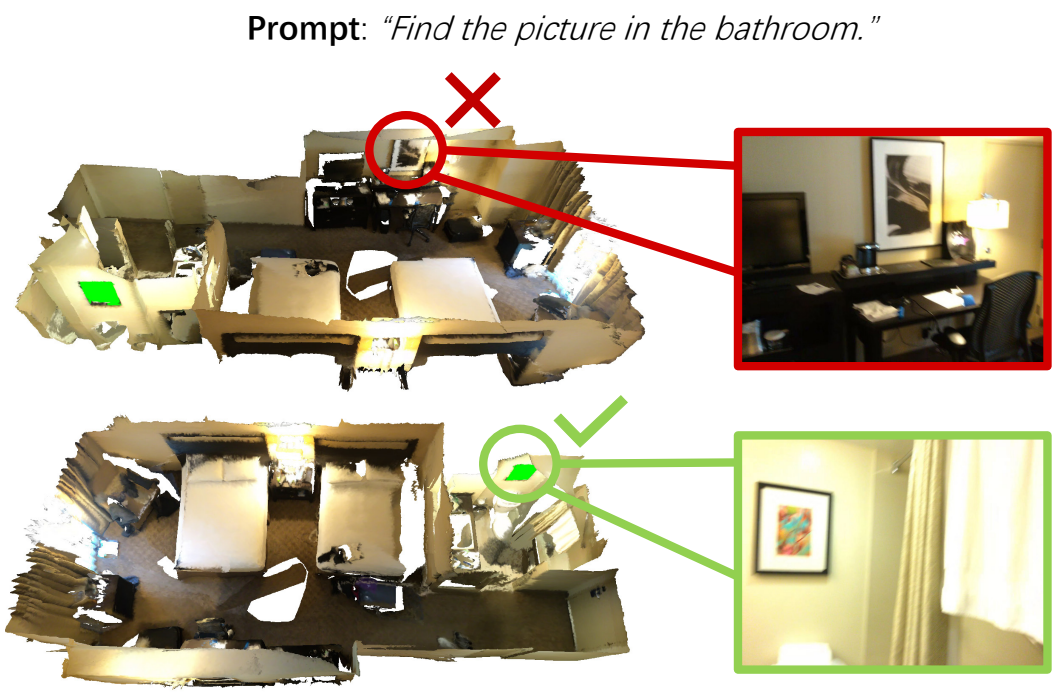


Figure 8. OV-3DIS result based on position prompt.

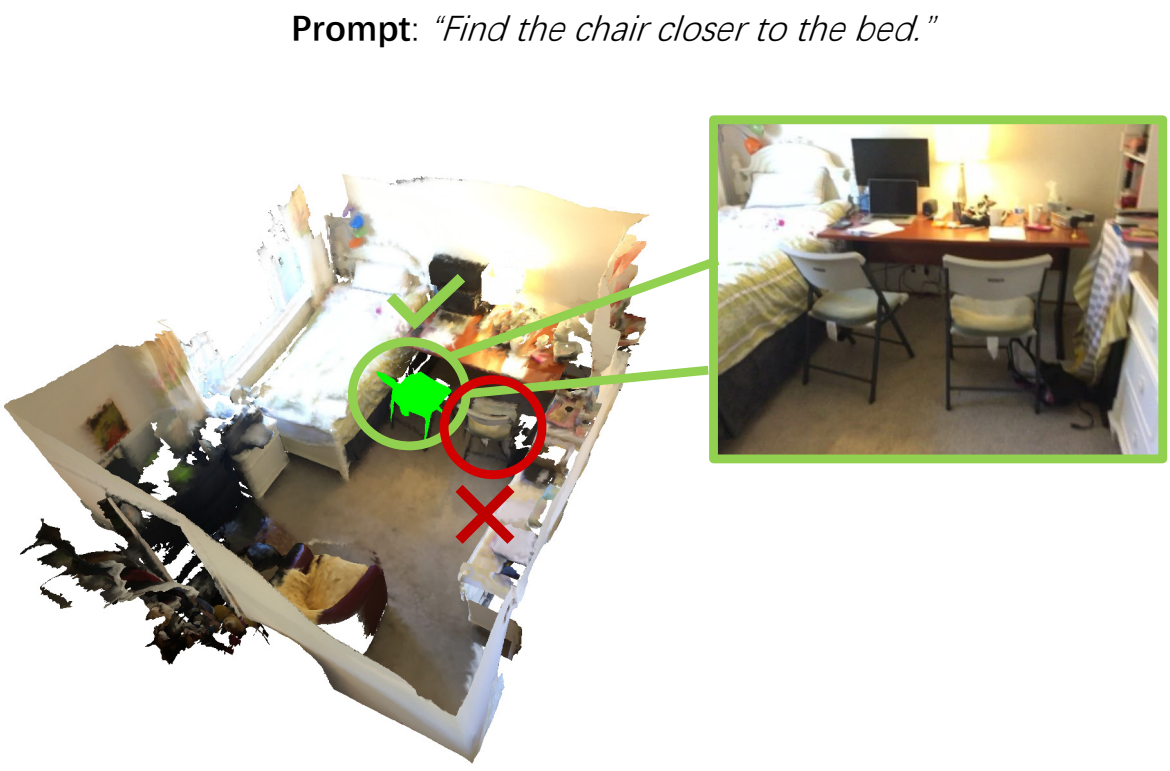


Figure 9. OV-3DIS result based on position prompt.

**Instruction:** *"Navigate to where the books are"*

**Instruction:** *"Find somewhere to sit"*

Figure 10. OV-3DIS result based on real-world instruction