Chair of Computational Modeling and Simulation
TUM School of Engineering and Design
Technical University of Munich

Open World 3D Scene Understanding

Scientific work to obtain the degree

Master of Science (M.Sc.)

at the TUM School of Engineering and Design
of the Technical University of Munich.

Supervised by Prof. Dr.-Ing. Andre Borrmann
M.Sc. Fiona Collins
Chair of Computational Modeling and Simulation

Dr. Sai Manoj Prakhya
Huawei - Intelligent Cloud Technologies Lab

Submitted by Rafay Mohiuddin ([redacted])
[redacted]
e-Mail: rafay.mohiuddin@tum.de

Submitted on 01. April 2024

Abstract

This study introduces an innovative approach that offers incremental and scalable solutions for constructing open set, instance-level 3D scene representations, leading to an open world understanding of 3D environment. Current methodologies in open vocabulary 3D scene understanding are predominantly non-incremental requiring pre-constructed 3D scenes, and they rely on learning per point feature vectors, creating scalability issues for many practical use cases. Moreover, their efficacy in contextualizing and responding to complex queries is considerably limited. The proposed method, addresses these limitations by leveraging 2D foundation models to incrementally construct instance-level 3D scene representations. It efficiently tracks and aggregates corresponding instance-level details (such as masks, feature vectors, names, captions etc.) from 2D foundation models to 3D space. Furthermore, our work introduces fusion schemes for feature vectors that effectively integrate contextual information, significantly enhancing performance on complex queries. Additionally, this study explores methods to effectively utilize large language models for, robust automatic annotation and complex spatial reasoning tasks over the constructed open set 3D scene. The proposed method is evaluated on ScanNet [4, 41] and Replica [44] datasets, both quantitative and qualitative results demonstrate its zero-shot generalization capabilities that exceeds current state-of-the-art methods in open world 3D scene understanding tasks.

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Comprehension and reasoning within perceived 3D environments pose significant challenges for many applications. Recent advancements in AI have led to significant breakthroughs in open set understanding and reasoning within 2D imagery, primarily attributed to the development of pre-trained foundation models [20, 24, 29, 36] and the synergistic integration of vision and large language models [26, 27, 58]. These developments have revolutionized the field, driving forward capabilities in 2D image processing and interpretation.

However, translating these successes to 3D scene remains a formidable task. Current approaches designed for 3D environments [6, 11, 12, 17, 33], while innovative, have not yet reached a performance level comparable to their 2D counterparts. This disparity poses a compelling and complex challenge in the field. Addressing this performance gap is crucial for a broad spectrum of applications requiring interaction with three-dimensional spaces. Bridging this divide can unlock the potential for the development of next-generation engineering tools such as digital twins, revolutionizing the way we perceive, analyze, and interact with the three-dimensional world.

1.2 Limitations of Current Methods

Recent advancements [6, 11, 12, 17, 33] have pioneered methods to integrate 2D foundation models for open-world 3D scene understanding tasks and have demonstrated impressive results. Yet, these innovative approaches are not without their limitations, especially in the context of practical applications. Primarily, many of these methods are designed as non-incremental solutions, presuming the full dataset of the 3D scene is available from the onset. This assumption often clashes with the dynamic and unpredictable nature of real-world environments. Moreover, their focus largely lies in the generation of per point 3D feature vectors derived from 2D vision-language embeddings. While these methods demonstrates a degree of effectiveness, they fall short in offering a universally applicable strategy for 2D to 3D information extrapolation, thus limiting its versatility. Additionally, the tendency to construct dense, per-point feature vectors poses significant scalability challenges and complicates the critical task of isolating distinct entities within a scene, an essential component for practicable use cases. Most notably, the practicality of these representation methods is constrained, predominantly effective for simpler queries, lacking in depth and contextual understanding for more complex spatial inquiries.

1.3 Proposed Approach

In this work, a novel framework is introduced, designed to address the challenges in 3D scene representation for tasks such as open vocabulary instance recall, segmentation, annotation, and spatial reasoning. This work proposes a unique approach that leverages 2D foundation models to extract instance-level information, subsequently constructing a 3D segmentation map through a purely geometric method. This process involves aggregating and associating data from 2D images to 3D space and creating an instance-level 3D segmentation map.

Utilizing a sequence of RGBD images, the method initially extracts masks, bounding boxes, names, captions, and prediction scores using GroundedSAM [39] and Large vision language models GPT-4V [32]. For each instance in the RGB sequence, the method crops individual instances at multiple scales to obtain feature vectors from the CLIP [36] encoder; these vectors are then aggregated using a multiscale feature fusion scheme, discussed. Meanwhile, each instance is assigned a unique ID. The 2D segmentation masks, enriched with these IDs as per-pixel labels, depth, and global pose, are utilized to back-project and construct the 3D scene. The method innovatively updates and tracks each back-projected segmentation mask based on the number of common points in overlapping regions, thereby enabling efficient and scalable construction of a comprehensive instance-level 3D scene.

Distinctively, the method constructs the scene incrementally, adapting as the environment is explored. This focus on instance-level representations and a simple count-based approach for updating masks and tracking identifiers enhances both efficiency and scalability. Additionally, the feature fusion schemes, detailed in Section 3.4, incorporate local context information, which aids in distinguishing instances within the same class using relational queries.

For evaluation, a series of experiments across diverse environments were conducted, employing standard datasets like Scannet [4, 41] and Replica [44]. Both quantitative and qualitative analyses underscore the method's proficiency in open-world scene understanding tasks.

1.4 Key contributions

This study bring the following key contributions to the field of 3D scene understanding:

1. A scalable, incremental approach for 3D instance segmentation is introduced, seamlessly merging instance-level information from 2D foundational models into a unified 3D scene representation.
2. An innovative feature fusion formulation is developed, enabling the identification of instances within the same class through contextual queries.
3. The use of large language models, in conjunction with constructed scene representation, is explored for automatic annotation and 3D spatial reasoning.

Chapter 2

Literature Review

2.1 Foundation and Large Language Models

In the evolving landscape of artificial intelligence, pretrained foundation models have been instrumental in driving forward advancement due to large-scale, adaptable architectures. Models like CLIP [36], BLIP [24], BLIP2 [23], and Florence [57] Flamingo [1], blend visual and textual representations learned using contrastive learning, significantly elevating multimodal understanding. In the area of image segmentation, promptable segmentation models like SAM [20] and open vocabulary 2D segmentation models such as LSeg [21], OVSeg [25], and CLIPSeg [29] have broadened the horizons of image processing.

Grounding, another crucial aspect of these advancements, involves contextualizing model outputs by linking model results to verifiable information, enhancing the model's interpretative accuracy. This evolution towards grounding in foundation models GSAM [39], SEEM [60], GDINO [28], Semantic-SAM [22], and Caption-Anything [49] represents another leap offering nuanced understanding and application in complex scenarios.

Recent advancements in language models such as GPT3 [2], GPT4 [32], LLaMA [48], LLaMA2 [47], have demonstrated groundbreaking performance leaps in natural language understanding. The combination of large language models with vision in recent works like LLaVA [27], LLaVA2 [26], Grounded-LLaVA [58], has opened new doors in open-world understanding and human-machine interaction.

Given the rise in capabilities of foundation and large language models, this work explores a generalizable approach for extracting and linking information between 2D images and 3D spaces.

2.2 3D Semantic Segmentation

Semantic segmentation has emerged as a significant challenge in 3D computer vision. In the past, numerous semantic SLAM approaches have been introduced. SemanticFusion [31] merges the output of semantic segmentation network with SLAM, probabilistically generating a surfel-based scene representation. Kimera [40] integrates semantics from neural networks within a voxel grid, while Voxblox++ [8] organizes object instances in a volumetric map. Methods such as Hydra [15] and Scene Graph Fusion [52] construct scene graphs on top of built semantic maps to enhance scene understanding.

Among recent works, Incremental 3D Semantic Scene Graph [51] investigates the incremental construction of scene graphs over sparse point maps by identifying 3D instances through the assessment of overlap among back-projected sparse points attributed to 2D instances. SAM3D [55] generates fine-grained 3D masks via back-projected mask proposals from SAM, employing bidirectional merging for consecutive frames, and a bottom-up approach for iterative 3D mask aggregation. Both [55] and [51] recognize 3D instances through overlapping regions. However, SAM3D [55] produces fine-grained 3D masks in a non-incremental manner, with an overall complexity of $\mathcal{O}(\log_2 n)$. However, it still depends on input size of the entire map being processed and requires a KDTree search of the entire frame for each frame per bidirectional update, which increases per bottom up iteration. Whereas [51] offers an incremental alternative, creating a sparse 3D point map with fixed computation requirement per update.

A principal constraint of preceding approaches is their foundation on a closed vocabulary paradigm. This work proposes an incremental alternative that leverages 2D mask proposals from SAM and an overlap-based method to generate fine-grained, instance-level 3D masks with a constant computation requirement per incremental update. Additionally, it streamlines 2D and 3D transfer by efficiently tracking the ID of each 2D mask and its corresponding 3D counterpart.

2.3 3D Scene Understanding

The field of 3D scene understanding has evolved significantly, building upon the success of 2D vision-language models. The key idea for initial approaches was to map features from 2D foundation models onto 3D spaces to identify relevant objects or regions corresponding to an open vocabulary query.

Based on this principle, many methods have been proposed [6, 10, 34, 43], OpenScene [33] and ConceptFusion [17] are notable early examples in this domain. OpenScene [33] utilizes CLIPSeg [29] for feature extraction from 2D images, projecting these into 3D spaces combined with point cloud data. ConceptFusion [17], meanwhile, computes and maps pixel-aligned embeddings from CLIP [36] to 3D space using Grad-SLAM [18]. However, the per-point feature representation approach introduces significant computational demands and scalability challenges. Another method, PLA [5], offers a unique approach by integrating language-driven techniques, employing the GPTViT2 [30] model for scene understanding through detailed captioning, yet shares similar computational complexities due to constructing per-point feature vectors. OpenMask3D [46], one of the latest advancements in 3D scene understanding, uses the Mask3D [42] model for 3D masks proposal. For each mask proposal, it finds corresponding 2D instances for determining per-instance feature vectors. OpenMask3D [46], due to its instance-centric approach, doesn't face scalability constraints; however, it is still non-incremental and limited by the 3D segmentation capabilities of base model [42].

Additionally, studies such as ConceptFusion [17] and OpenMask3D [46] have explored advanced feature engineering techniques. These techniques involve the fusion of CLIP feature vectors from object-centric crops with those from larger image sections to achieve a more nuanced representation. However, the effects and implications of these methods require more comprehensive evaluation and detailed discussion.

While these methods mark progress in 3D scene understanding, they highlight ongoing challenges such as high computational load, scalability, and dependence on pre-existing 3D scenes, indicating a need for more efficient, scalable, and versatile solutions in this rapidly advancing field.

2.4 3D Spatial Reasoning

Global 3D spatial reasoning remains a formidable challenge in the domain of open-world 3D scene understanding. Recent studies have introduced various methodologies to address this issue, including 3DCLR [11], 3DLLM [12], and GroundedLLM [53], each proposing innovative approaches to tackle global spatial reasoning with large language models (LLMs). Despite these advancements, achieving accurate 2D spatial reasoning is still a challenging task; even state-of-the-art models like GPT-4V [32] face challenges in reasoning over 2D images [54]. Recent efforts, such as ViperGPT [45], have attempted to overcome these obstacles by integrating 2D object detection models with the code generation capabilities of LLMs like GPT-3 CODEX [3]. Another study, Set of Mark Prompting [54], explored prompting strategies to directly use large vision-language models for spatial reasoning on 2D images.

In this study, similar to the approach of [54], a prompting strategy aimed at directly utilizing large language models for 3D spatial reasoning tasks over constructed 3D scenes using the proposed method is explored.

2.5 Concurrent Work

Concurrently with this work, Segment3D [13], OpenIns3D [14], SayPlan [37], LangSplatt [35], and others have proposed different methods to address the challenges of 3D scene understanding. Among these, OVSG [25] and ConceptGraph [9] share the closest resemblance to the approach discussed here, as they also focus on constructing an incremental, scalable instance-based representation for scene understanding. In contrast to these methods, which rely on CLIP similarity for merging 3D segmentation masks, our proposed approach relies only on geometric principles. Additionally, while the aforementioned works emphasize scene graph creation for global spatial reasoning, this study explores leveraging Large Language Models with long context window and prompting strategies for intricate spatial reasoning tasks.

Chapter 3

Methodology

Our method, processes a sequence of RGBD images and their poses to create a open set 3D scene representation for open world scene understanding tasks like, open vocabulary object retrieval, segmentation, annotation and spatial reasoning. Illustrated in Fig. 3.1 the pipeline contains two main modules:

- **Feature Extraction:** Extracts instance-level details from images and assigns a unique ID to each instance for precise tracking.
- **2D to 3D Fuse and Track:** Creates a 3D semantic map from 2D masks, and associate 2D information into the 3D space by tracking the corresponding IDs.

These two modules construct a 3D scene representation, which is subsequently utilized for downstream open world scene understanding tasks.

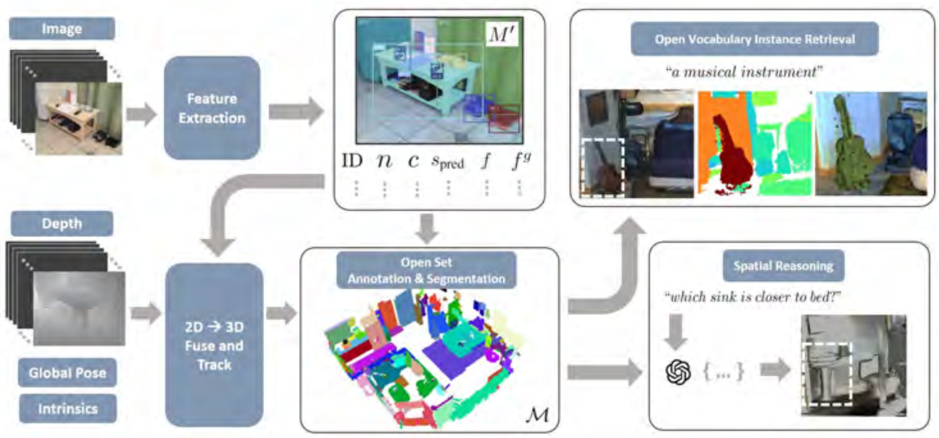


Figure 3.1: **Workflow of proposed method - Open World 3D Scene Understanding.** Given a sequence of RGB-D images, our method constructs a 3D scene representation for open vocabulary instance retrieval, open set annotation & segmentation, and spatial reasoning tasks.

3.1 Feature Extraction

The feature extraction process (Algo. 1) commences with a sequence of RGB images, denoted as $\mathcal{I} = \{I_0, I_1, I_2, ..., I_n\}$. We sample a subset $\mathcal{I}' = \{I_0, I_s, I_{2s}, ..., I_n\}$ from $\mathcal{I}$ with a stride s, which ensures a reasonable overlap to minimize computational redundancy. For each image $I' \in \mathcal{I}'$, we employ the groundedSAM [39] to obtain a set of 2D masks M, bounding boxes BB, and prediction scores S_{pred}. Concurrently, crop of each instance utilizing $bb \in BB$ are passed to a vision language model, GPT-4V [32], to get a precise list of names N and detailed captions C, describing each instance.

Each instance is assigned a unique identifier ID, leading to an updated set of 2D masks M', where each pixel label is modified to reflect the new ID. Additionally, a border of px pixels is added around each mask within the image to delineate distinct entities.

Feature vectors are extracted using the CLIP encoder in two stages:

1. A global feature vector f^g is derived for the entire image.
2. Instance-specific feature vectors F are computed by cropping the image multiple times, guided by a set of scaling ratios $S_r = \{s_r\}_k$ for crop sides and bounding boxes $bb \in BB$. These vectors are then integrated using a multiscale feature fusion scheme.

Finally, we store the image-level updated masks M' and instance-level data, including the identifiers ID, names $n \in N$, captions $c \in C$, prediction scores $s_{\text{pred}} \in S_{\text{pred}}$, feature vector after multiscale fusion $f \in F$ and global feature vector f^g, in a hash table for each image in $\mathcal{I}'$.

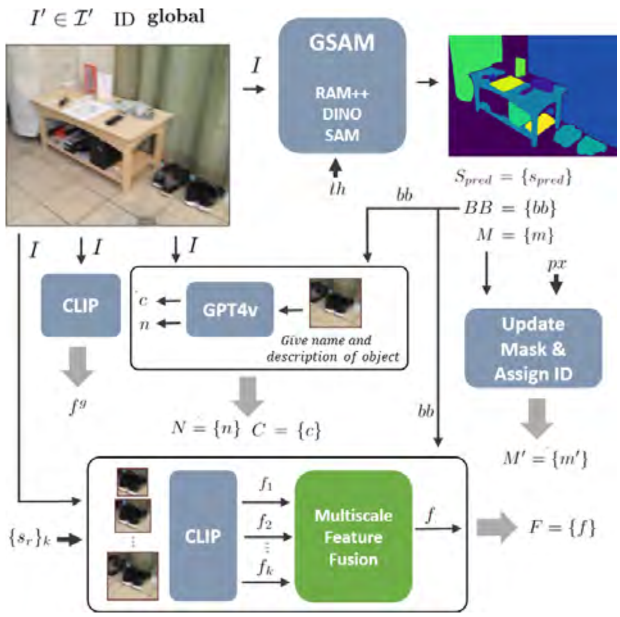


Figure 3.2: **Overview of feature extraction module.** Every image undergoes processing through a series of foundation models. For each instance within an image, a globally unique ID is assigned, accompanied by its name, bounding box, caption, prediction score, and associated CLIP feature vectors.

3.2 2D to 3D Fusion & Tracking

We initiate the second module (Algo. 2) by initializing an empty 3D point cloud for the scene, represented as $\mathcal{P}_{\text{scene}} \in \mathbb{R}^{x,y,z,\text{ID}}$. A global hash table $\mathcal{Q}$ for tracking the unique IDs, defined as:

$$\mathcal{Q} : \mathcal{Q} \mapsto \left\{ \text{ID} \in \texttt{unique}(\text{ID} \in \mathcal{P}_{\text{scene}}) : \{\text{ID} \in \{M'\}\} \right\} \tag{3.1}$$

For the image I', associated elements including depth maps D, global poses T, updated masks M', and camera intrinsic K are retrieved. The back-projection of each pixel $(u, v) \in I'$ in the images into a 3D space is performed, assigning it a semantic label from the corresponding mask M' and aggregating into a point cloud $\mathcal{P}_{\text{frame}}$.

$$\mathcal{P}_{\text{frame}} = \left\{ \left(T \cdot \left(D(u,v) \cdot K^{-1} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \right), M'(u,v) \right) \right\} \tag{3.2}$$

Index pairs $\{(\mathbf{i}_{\text{frame}}, \mathbf{i}_{\text{scene}})\}$ are determined to identify corresponding points between $\mathcal{P}_{\text{frame}}$ and $\mathcal{P}_{\text{scene}}$ (Algo. 3). Using bounds of $\mathcal{P}_{\text{frame}}$, $\mathcal{P}'_{\text{scene}}$ is sampled from $\mathcal{P}_{\text{scene}}$, containing only points within the bounds of $\mathcal{P}_{\text{frame}}$. A $KDTree$ search is performed, utilizing Euclidean distance function, $d(\cdot,\cdot)$ to matches points $\mathbf{p} \in \mathcal{P}_{\text{frame}}$ with points $\mathbf{q} \in \mathcal{P}'_{\text{scene}}$. If $d(\mathbf{p}, \mathbf{q}) < \epsilon$, group indices corresponding to $\mathbf{p} \in \mathcal{P}_{\text{frame}}$ with $\mathbf{q} \in \mathcal{P}_{\text{scene}}$ to get respective index pairs $\{(\mathbf{i}_{\text{frame}}, \mathbf{i}_{\text{scene}})\}$ for all overlapping points. This search strategy limits the size of $KDTree$ search, thereby requiring a constant computation per update.

To update and track IDs, (Algo. 4), Similar to SAM3D's [55] approach, we begin by obtaining a list of unique IDs $\{\text{ID}_f\}$ for each segment, alongside a corresponding list denoting the total point count $\{c_{\mathcal{P}_f}\}$ of each segment of $\mathcal{P}_{\text{frame}}$.

For each segment in $\mathcal{P}_{\text{frame}}$ with $c_{\mathcal{P}_f} \in \{c_{\mathcal{P}_f}\}$, we utilize index pairs $\{(\mathbf{i}_{\text{frame}}, \mathbf{i}_{\text{scene}})\}$, to get chunk points from $\mathcal{P}_{\text{scene}}$ that overlaps $\mathcal{P}_{\text{frame}}$, and from these points we derive a list of unique segment IDs $\{\text{ID}_s\}$ and their corresponding total point counts $\{c_{\mathcal{P}_s}\}$. The overlap ratio is then evaluated as:

$$\text{OverlapRatio} = \frac{\max(\{c_{\mathcal{P}_s}\})}{\min(c_{\mathcal{P}_f}, \max(\{c_{\mathcal{P}_s}\}))} \tag{3.3}$$

If the overlap ratio satisfies a pre-defined threshold, i.e., OverlapRatio $\geq \rho$, we perform an ID replacement and update operation. Specifically, all $\text{ID}_f \in c_{\mathcal{P}_f}$ present in the frame $\mathcal{P}_{\text{frame}}$ are replaced with $\text{ID}_s \in \max(\{c_{\mathcal{P}_s}\})$ to get, $\mathcal{P}'_{\text{frame}}$ which is then concatenated to $\mathcal{P}_{\text{scene}}$. Additionally, chunk of previous points from $\mathcal{P}'_{\text{scene}}$ can also be deleted to retain constant sparsity, ensuring fixed computation requirement per update. The updated IDs are then appended to the $\mathcal{Q}$:

$$\mathcal{Q}[\text{ID}_s \in \max(\{c_{\mathcal{P}_s}\})] \leftarrow \mathcal{Q} \cup \{\text{ID}_f \in c_{\mathcal{P}_f}\} \tag{3.4}$$

Conversely, if the overlap ratio does not meet the threshold requirement, a new entry is added to $\mathcal{Q}$ directly from $c_{\mathcal{P}_f}$:

$$\mathcal{Q}, \mathcal{Q}[\text{ID}_f \in c_{\mathcal{P}_f}] \leftarrow \mathcal{Q} \cup \{\text{ID}_f \in c_{\mathcal{P}_f}\} \tag{3.5}$$

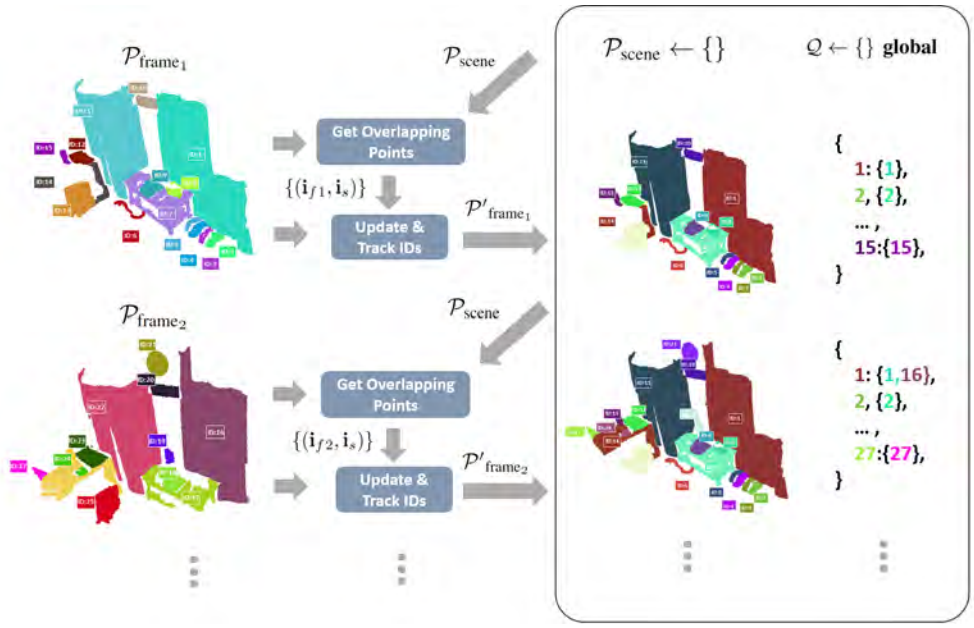


Figure 3.3: **Overview of 2D to 3D fusion & tracking module.** This module updates and tracks IDs for each back-projected semantic mask associated with an image by assessing the overlap with previously back-projected semantic masks. Tracked IDs are recorded, and updated projections are concatenated.

3.3 Post Processing

Upon obtaining the 3D point cloud $\mathcal{P}_{\text{scene}}$, with includes updated IDs and $\mathcal{Q}$ track of overlapping IDs, we engage in an post-processing stage (Algo. 5). This stage focuses on constructing an instance-centered map representation $\mathcal{M}$, defined as:

$$\mathcal{M} : \mathcal{M} \mapsto \{\{\mathcal{P}, n, c, f, bb_{3D}, (x_c, y_c)\}_i | i \in \texttt{unique}(\text{ID} \in \mathcal{P}_{\text{scene}})\} \tag{3.6}$$

For each distinct 3D object $\mathcal{P}_i$ within $\mathcal{P}_{\text{scene}}$, the initial step is to apply DBSCAN clustering to mitigate noise and split over merged segments. Subsequently, we compute the 3D bounding box $bb_{3D,i}$ and the centroid coordinates $(x_c, y_c)_i$ for each object. The corresponding 2D IDs $\mathcal{Q}[\text{ID} \in \mathcal{P}_i]$ are then used to retrieved corresponding names N', captions C', prediction scores S'_{pred} and feature vectors F'.

Using the maximum prediction score, we assign the name n_i and caption c_i as in $N'[\arg\max(S'_{\text{pred}})]$ and, $C'[\arg\max(S'_{\text{pred}})]$ respectively. Alternatively, we can leverage large language models to add redundancy by considering names for images with top m prediction scores $N'[\arg\max(S'_{\text{pred}}, m)]$ and pass it to the language model with prompt to `'assign a name to the object based on a given list of names'`, to get more precise names n'_i.

For feature fusion, we employ the multiview fusion scheme discussed in Sec. 3.4. To introduce redundancy, we consider only the feature vectors corresponding to the top m prediction scores $F'[\arg\max(S'_{\text{pred}}, m)]$ to get the feature vector corresponding to each instance f_i.

3.4 Feature Fusion

Provide a list of feature vectors $\{f\}_m$ corresponding to multiple view of images of a 3D instance and feature vectors $\{f\}_k$ from multiple scale crops of an instance in an image. Feature vectors can be aggregated by direct approach, as mentioned below:

$$f_{\text{MultiScale}} = \frac{1}{k}\sum_{i=0}^{k} f_i \tag{3.7}$$

$$f_{\text{MultiView}} = \frac{1}{m}\sum_{i=0}^{m} f_i \tag{3.8}$$

To understand the limitations of fusion schemes in Eq. 3.7 and Eq. 3.8, extensive ablation studies were conducted as discussed in Sec. 4.2.1. Similar to the findings of OpenMask3D [46], it was observed that adding multiscale crops makes the feature vector redundant and adds surrounding context, however, a larger crop size results in the deterioration of performance. To mitigate this, a modified multiscale fusion scheme, Eq. 3.9, is proposed. The key strategy of this approach is to downweight the influence of larger crops. Hence, in the updated approach, weight is assigned to the next coming feature vector from a crop with a cosine similarity score with respect to the best fit crop.

$$f_{\text{MultiScale}'} = \frac{1}{k}\sum_{i=0}^{k} \left(\frac{f_0 \cdot f_i}{\max(\|f_0\|_2 \cdot \|f_i\|_2)}\right) \cdot f_i \tag{3.9}$$

For the integration of multiview features, drawing parallels to the methodological approach outlined in ConceptFusion [17] which focuses to the fusion of 2D per-pixel features. We propose direct incorporation of a global feature vector f^g, whilst synthesizing multiview features vector for each instance, defined as:

$$f_{\text{MultiView}'} = \frac{1}{m} \sum_{i=0}^{m} \left(f_i + \left(\frac{f_i \cdot f_i^g}{\max(\|f_i\|_2 \cdot \|f_i^g\|_2)} \right) \cdot f_i^g \right) \tag{3.10}$$

3.5 Instance Retrieval & Segmentation

Given a map $\mathcal{M}$, the instance retrieval and segmentation (Algo. 7) operates in two stages. In the first stage, a given query $\mathcal{K}$ is processed using CLIP text encoder to obtain the corresponding feature vector $f_{\mathcal{K}}$. In the second stage, a similarity score is computed for all instance, and the segmentation mask corresponding to the instance with maximum score is retrieved, denoted as $\mathcal{M}[\text{argmax}(S_{\text{score}})]$ as the most likely response to the query $\mathcal{K}$, here S_{score} is defined as:

$$S_{\text{score}} = \left\{ \left(\frac{f_i \cdot f_{\mathcal{K}}}{\max(\|f_i\|_2 \cdot \|f_{\mathcal{K}}\|_2)} \right) \middle| f_i \in \mathcal{M} \right\} \tag{3.11}$$

Alternatively, multiview feature vector with top m images can also be arranged base on CLIP similarity score instead of prediction score S_{pred} from GroundedSAM [39], directly during query process instead of post-processing Sec. 3.3 (Algo. 6).

3.6 Spatial Reasoning

For queries involving complex spatial reasoning, the key idea is to leverage the long context window of Large Language Models like GPT-4 [32], to perform symbolic reasoning based on information available with in constructed scene representation $\mathcal{M}$. A simplified map $\mathcal{M}' := \mathcal{M} \setminus \{\mathcal{P}, f, f^g\}$ is parse along with system prompt fine-tuned, using the prompting strategy defined as:

```
- Use ‘‘Name’’ & ‘‘Description’’ to understand object.

- Use ‘‘ID’’ to refer object.
```

```
- Get ‘‘Centroid’ & ‘‘Bounding Box’’ information.

- Do ‘‘Symbolic Computation’’.

- Compute ‘‘Euclidean Distance’’ if necessary.

- Assume ‘‘Tolerance’’ if necessary.

- HINT: Rely on ‘‘Cartesian Coordinates’’.
```

If the information regarding instances of interest is mentioned in query $\mathcal{K}$, an efficient approach is to first use the LLM to identify instances $\{\mathcal{K}\}_n$ from query $\mathcal{K}$ and subsample those instances from the map using CLIP-based similarity scores to obtain $\mathcal{M}'$. This can then be parsed to the LLM as discussed earlier.

$$\mathcal{M}' : \mathcal{M}' \mapsto \{\mathcal{M}[argmax(S_{\text{score}_i})] \setminus \{\mathcal{P}, f, f^g\} | S_{\text{score}_i} \in \{S_{\text{score}}\}_n\} \tag{3.12}$$

Chapter 4

Results and Discussion

4.1 Experiment Setup

4.1.1 Datasets

For experimentation, we employed scenes from two datasets: the semi-synthetic dataset Replica [44] and the real-world dataset ScanNet [4, 41].

The **Replica** [44] dataset comprises evenly distributed, rendered images from pre-constructed, high-fidelity 3D models of real-world scenes using algorithmically generated poses. The experiments utilized the following scenes from Replica: *room0*, *room1*, *room2*, *office0*, *office1*, *office2*, *office3*, *office4*. This selection aligns with the scenes used in recent studies, facilitating direct comparison.

The **ScanNet200** [41] dataset is an augmented version of the **ScanNet** [4] dataset incorporating additional ground truth categories. We randomly extracted samples from five distinct categories and chose the following scenes for the analysis: *scene0000_00* (apartment), *scene0034_00* (bathroom), *scene0164_03* (kitchen), *scene0525_01* (office), and *scene0549_00* (lobby). Given the emphasis of the experiments is predominantly focused on manual human evaluation (Explained in Sec. 4.2.2), we limited the selection to specified scenes for a more focused assessment.

4.1.2 Implementation Details

Foundation and Large Language Models

In the feature extraction pipeline, GroundedSAM [39] is used, a method based on RAM++ [59] (`ram_plus_swin_large_14m`), GroundingDINO [28] (`groundingdino_swint_ogc`), and SAM [20] (`sam_vit_h_4b8939`) for the generation of instance-level mask proposals and bounding boxes. GPT-4V [32] (`gpt-4-vision-preview`) is employed for the generation of detailed captions and names. The CLIP encoder [36] (`ViT-H-14`), pre-trained on (`laion2b_s32b_b79k`), is used to obtain the feature vector. Lastly, GPT-4 [32] (`gpt-4-1106-preview`) is utilized as the large language model, for spatial reasoning and annotation tasks.

Hyperparameters

In this study, a consistent set of hyperparameters was employed across different datasets, guided by insights gained from ablation studies (Sec. 4.2.1), conducted on the Replica [44] scenes. Specifically, the top $m = 5$ images were selected, and $k = 3$ levels of crops were applied, with a scaling ratio S_r set to $[0.8, 1, 1.2]$, expanding at a factor of 0.2. The experiments were conducted with a stride of $s = 40$ to ensure adequate overlap for the method, recognizing that the optimal stride may vary with the data acquisition rate.

For GroundedSAM [39], the Intersection over Union (IoU) threshold was set to 0.4, with both bounding box and text thresholds at 0.25, denoted as $th = [0.4, 0.25, 0.25]$. A constant padding of $px = 20$ pixels was utilized to delineate the borders between instance masks. Overlap ratio evaluation was performed using a voxel size of $\epsilon = 0.02$ and a fixed overlap threshold of $\rho = 0.3$. Post-processing employed DBSCAN with an epsilon of 0.1 and a minimum cluster size of 20 points. A temperature setting of 0 was utilized with GPT-4 [32].

Considering the lower resolution of depth images in ScanNet [4], a statistical filter was applied for pre-filtering (with a minimum neighbor count of 20 and a standard deviation ratio of 0.2) to mitigate artifacts. This led to a revised parameter set for overlap ratio in ScanNet, with a voxel size of $\epsilon = 0.01$ and an overlap threshold $\rho = 0.05$.

Filter Implementation

During the feature extraction stage, large objects such as *wall*, *ground*, *roof*, and *ceiling* are excluded based on the names assigned to the 2D masks, along with objects whose bounding boxes occupy more than 95% of the image area. This is because their corresponding feature vectors from the CLIP [36] encoder are likely to exhibit more similarity to the object in front of them, thereby adversely affecting the overall recall performance and similarity score distribution. Additionally, during the post-processing of each segment with DBSCAN, if a cluster is at least 80% the size of the largest cluster in the segment, it is considered a separate instance. A unique ID and the same attributes as the largest cluster are then assigned to it.

Additional Implimentation Details

For KDTree search, `pyfnntw` [7], a high-performance parallel k-NN library, is utilized with configuration parameters set to `parallel_split_level_size=2` and `leaf_size=16`. For CLIP `open_clip` [16] library is utilized, with `ViT-L-14` pre-trained on `open-ai` and `ViT-H-14` & `ViT-G-14` pre-trained on `LAION-2B` as CLIP model variants. In situations where GPT-4 [32] enters safety mode, names from RAM++ [59], and simplified captions format as "`an {object} in a scene`" are assigned to the instance.

4.1.3 Quantitative Evaluation

For the quantitative evaluation of proposed method, standardized metrics widely adopted in instance and semantic segmentation tasks were utilized. These metrics include mean recall accuracy (mAcc), frequency-weighted intersection over union (F-mIoU), and average precision (AP) across an IoU range of $[0.5 : 0.05 : 0.95]$. Additionally, AP50 and AP25, which measure average precision at IoU thresholds of 50% and 25%, respectively, were reported. The definitions for AP scores adhere to the specifications outlined in ScanNet [4].

Given the objective to quantitatively evaluate open vocabulary performance, current approach was inspired by method outlined in OpenMask3D [46] and ConceptGraph [9]. For the evaluation of quantitative metrics relative to the ground truth mask, 3D masks from the constructed scene were called using the ground truth object label with a prompt '`an {object} in a scene`'. As the method does not provide direct point-to-point label correspondence with the ground truth, a strategy similar to that of ConceptGraph [9] for identifying intersecting points was adopted. The ground truth and retrieved mask were downsampled to a voxel size of 0.25cm, and a nearest neighbor search with a threshold of 0.25cm was applied to find intersecting points.

Quantitative findings on the Replica [44] dataset were compared against results from OpenMask3D [46], Segment3D [13], ConceptFusion [17], and ConceptGraph [9], based on the metrics described above. To facilitate a direct comparison, identical prompts for mask retrieval and uniform foundation models were employed across respective baseline methods.

4.1.4 Qualitative Evaluation

The quantitative metrics detailed in Section 4.1.3 provide a limited perspective as recall performance is confined to a closed set of ground truth instances. Additionally, the recall queries are overly simplified and do not adequately capture the requirements for practical open vocabulary use cases. To facilitate a more comprehensive evaluation, an extensive qualitative assessment conducted by human evaluators is planned. This qualitative evaluation aims to measure the efficacy of open vocabulary instance retrieval, spatial reasoning, and annotation performance.

To assess **open vocabulary instance retrieval**, over 1,000 queries encompassing instances, affordances, properties, and relative queries across all test categories were asked. This approach aims to evaluate the open vocabulary instance recall performance, leveraging CLIP [36]. The efficacy of four distinct fusion scheme combinations was examined. In *"Scheme 1"*, both multiscale and multiview and multiscale features are aggregated directly according to Eq. 3.7 and Eq. 3.8 respectively, with parameters determined from the ablation studies Sec. 4.2.1. *"Scheme 2"* utilizes updated multiview features as delineated in Eq. 3.10, while *"Scheme 3"* employs updated multiscale features as described in Eq. 3.9, adopting x10 crop expansion ratios starting from the base image, specifically $S_r = [1, 2, 4]$. Finally, *"Scheme 4"* utilizes both updated multiview and multiscale features fusion formulations, Eq. 3.10 and Eq. 3.9 respectively. The principal aim here is to explore the potential of these fusion schemes to incorporate additional contextual information, thereby enhancing their performance on relative/contextual queries.

In the similar fashion, the performance of **open vocabulary annotation and segmentation** is meticulously evaluated through manual verification of label assignment and mask merging flaws.

To evaluate **spatial reasoning** abilities, a total of 70 spatial reasoning questions across all scenes were administered. These inquiries aimed to assess spatial reasoning capabilities using a large language model through the proposed methodology (Sec. 3.6), with assessments conducted by a human evaluator. The evaluation is primarily exploratory in nature, with the principal objective being to investigate the viability of the prompting strategy for directly using large language models over constructed scene representations for spatial reasoning tasks.

4.2 Results and Analysis

4.2.1 Ablation Studies

To assess the influence of hyperparameters, we conducted multiple ablation studies. We specifically assess the effect of Crop Level (number of crops) k, Top Images (images with the best prediction scores) m, and Crop Ratios (ratio for scaling crop sides) S_r using quantitative metrics.

Top Images m affects multiview feature fusion Eq. 3.10 as it specifies the feature vectors to be used for aggregation. While Crop Ratio S_r and crop level k affect multiscale feature fusion Eq. 3.7 as they respectively specify crop size and their quantity being used for getting feature vector for aggregation. It is important to point out that k amplifies the effect of S_r, as a large Crop Level with the same Crop Ratio would result in a much larger crop of the image.

Similar to the conclusions of OpenMask3D [46], we found that both lower and very high values of these hyperparameters would lead to deterioration of results.

A lower value of m would result in reduced redundancy, while a high value may include bad images (images with lower S_{pred}) in multiview feature fusion, as shown in Table C.3.

Top Images	mAcc	F-mIoU	AP	AP50	AP25
1.0	39.6	43.4	8.7	19.3	27.2
5.0	**40.8**	**44.7**	**8.9**	**19.6**	**27.7**
10.0	39.3	44.3	8.7	19.1	27.5

Table 4.1: **Ablation study of top images (m).** The total images m with top s_{pred} considered for multiview feature fusion, assessed on Replica [44] dataset.

As highlighted in Tables C.1 and C.2, lower values of S_r and k may not adversely affect the model, and could introduce redundancy. However, larger cropping sizes could result in a saturation of the similarity score distribution associated with the query. This occurs because larger S_r values introduce additional influence from the surrounding context. Ideally, larger crops should add more context without leading to saturation of the similarity score.

Crop Ratio	mAcc	F-mIoU	AP	AP50	AP25
0.1	39.9	44.4	**8.9**	19.4	**28.1**
0.2	**40.8**	44.7	**8.9**	**19.6**	27.7
0.3	39.9	**44.8**	**8.9**	19.4	27.3

Table 4.2: **Ablation study of crop scaling ratio (S_r).** Ratio S_r by which the crop sides are scaled for multiscale feature fusion, assessed on the Replica [44] dataset.

Crop Levels	mAcc	F-mIoU	AP	AP50	AP25
1.0	35.9	43.6	**9.1**	**19.6**	**27.7**
3.0	**40.8**	**44.7**	8.9	**19.6**	**27.7**
5.0	39.4	44.3	8.8	19.4	26.9

Table 4.3: **Ablation study of crop level (k).** The total number of crops k, considered for multiscale feature fusion, assessed on the Replica [44] dataset.

We also conducted an ablation study with different CLIP model variants. Quantitatively, as shown in Table 4.4, a larger CLIP variant resulted in improved overall performance. Furthermore, we provide an example in Fig. 4.1 where using a larger CLIP model led to better association of properties with respect to the queried object, enabling the model to distinguish between *"single sofa"* and *"double sofa"* within the scene.

CLIP Models	mAcc	F-mIoU	AP	AP50	AP25
ViT-L-14	38.9	48.2	9.3	20.5	30.3
ViT-H-14	**42.6**	40.9	9.9	**21.6**	**31.6**
ViT-G-14	**42.6**	**46.4**	**10.4**	21.1	28.4

Table 4.4: **Ablation study of CLIP model variants**. Influence of different CLIP Models on instance recall performance, assessed on Replica [44] dataset.

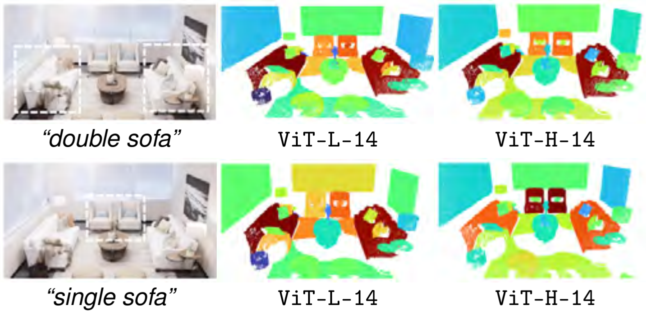


Figure 4.1: **Heatmaps representing similarity between text queries and scene-instances across CLIP models.** For a given text query (left), comparison of per-instance cosine similarity results for `ViT-L-14` (middle) and `ViT-H-14` (right) on scenes from Replica [44]. 'Dark red' represents maximum similarity, and 'dark blue' indicates minimum similarity.

4.2.2 Open Vocabulary Instance Retrieval

Quantitative Comparison with Baseline Methods

Under conditions similar to those outlined in baseline studies, the proposed method demonstrates comparable or better performance on quantitative metrics. We report results in Tables 4.6 and 4.5, comparing them to those reported in the respective original studies. The objective is to assess the accuracy and precision in segmentation masks retrieved in response to open vocabulary queries, with respect to the ground truth masks.

For comparisons with methods listed in Table 4.6, we employ `CLIP-ViT-H` as the CLIP encoder and invoke instances with the prompt "`an image of a {object}`", as described in ConceptGraph [9]. For the methods outlined in Table 4.5, we utilize the prompt "`an {object} in scene`" with `CLIP-ViT-L` as the CLIP encoder, in accordance with the original study OpenMask3D [46], further details regarding experimental setup are discussed in Sec. 4.1.3.

Additionally, detailed per-scene results and self-evaluation of the baseline method are reported in Tables C.6, C.5, and C.7. Overall, across all metrics and datasets, we found the performance to be comparable to or better than the baseline work.

Method	Replica [44]		
	AP	AP50	AP25
OpenMask3D [46]	**13.0**	18.4	24.2
OpenMask3D+Segment3D [13]	-	18.7	-
OpenSU3D (Ours)	8.9	**19.6**	**27.7**

Table 4.5: **Comparison of open-vocabulary segmentation results.** Quantitative evaluation and comparison of open vocabulary instance segmentation performance with counterpart methods, using experimental settings as proposed in OpenMask3D [46].

Method	Replica [44]	
	mAcc	F-mIoU
ConceptFusion [17]	24.2	31.3
ConceptFsuion+SAM [17]	31.5	38.7
ConceptGraph [9]	40.6	36.0
ConceptGraph-Detector [9]	38.7	35.4
OpenSU3D (Ours)	**42.6**	**40.9**

Table 4.6: **Comparison of open-vocabulary segmentation results.** Quantitative evaluation and comparison of open vocabulary instance segmentation performance with counterpart methods, using experimental settings as proposed in ConceptGraph [9].

Limitation of Quantitative Metrics

The quantitative methods employed were primarily designed for closed vocabulary assessments. Focusing on metrics that rely on recall accuracy corresponding to the ground truth label, which do not reflect real-world requirements for open vocabulary queries.

While these methods can provide a holistic overview of system performance, they may not accurately represent true performance due to their dependence on the quantity of mask proposals [13]. Specifically, for methods that generate a greater number of mask proposals, the uncertainty associated with achieving the highest recall does not necessarily align with ground truth labels.

Qualitative Comparison with Baseline Methods

To address the limitations of quantitative metrics pointed out in Sec. 4.2.2 we also provide comprehensive qualitative comparison with baseline works in Fig. 4.2, B.12, B.2 and B.1. The objective is to assess the ability to recall the correct segmentation mask corresponding to open vocabulary query, with high similarity scores assigned to relevant objects and lower to irrelevant objects.

Overall, we found performant across all methods to be comparable. However, in few cases, proposed method resulted in better 2D to 3D association and distribution of similarity of scores with direct feature fusion formulation (Eq. 3.10 and Eq. 3.9). As in Fig. 4.3 and 4.2 for query *"sideboard"* and *"an empty vase"* both of the baseline method recalled incorrect object. Whereas for queries *"a coffee table"* and *"fridge"* the proposed approch demonstrated better distribution of similarity score, i.e., assigning higher similarity to the relevant object and lower to others.

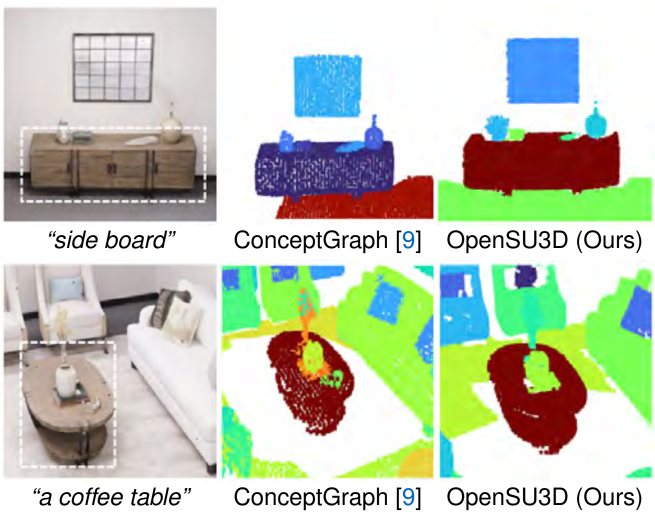


Figure 4.2: **Heatmaps representing similarity between text queries and scene-instances.** For a given text query (left), comparison of per instance cosine similarity scores for ConceptGraph [9] (middle) & OpenSU3D (right). 'Dark red' represents maximum similarity, and 'dark blue' indicates minimum similarity.

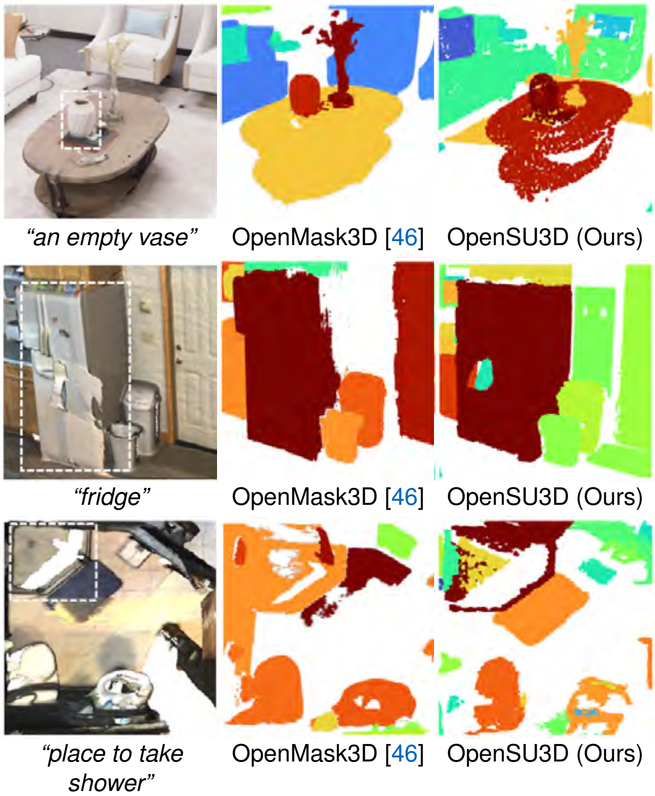


Figure 4.3: **Heatmaps representing similarity between text queries and scene-instances.** For a given text query (left), comparison of per instance cosine similarity scores for OpenMask3D [46] & OpenSU3D (right). 'Dark red' represents maximum similarity, and 'dark blue' indicates minimum similarity.

Assessment of Feature Fusion Schemes

Due to limitations of quantitative methods (Sec. 4.2.2), we conducted extensive qualitative evaluation of feature fusion schemes as defined in Sec. 4.1.4. Table 4.7 and Fig. 4.4 report an overview of qualitative performance on ScanNet [4] and Replica [44] scenes. The objective of the study is to assess the ability of feature fusion schemes (Sec. 3.4) to recall the correct segmentation mask of instances over a diverse set of open vocabulary queries, with high similarity scores assigned to relevant objects and lower scores to irrelevant objects.

Table 4.7 reports an overview of qualitative performance on ScanNet [4] and Replica [44] datasets. From Table 4.7, it can be inferred that, for *Instance*, *Property*, and *Affordance* query types, performance across all schemes remains comparatively same. However, for the *Relative* query type (query with more than one instance), the performance of *Scheme 2* and *Scheme 3* with updated feature fusion formulations Eq. 3.10 and Eq. 3.9, respectively resulted in better performance than *Scheme 1* with direct multiview Eq. 3.8 and multiscale Eq. 3.7 feature aggregation. Finally, *Scheme 4* with both updated formulations for multiview Eq. 3.10 and multiscale Eq. 3.9 feature fusion resulted in more accurate recall of instance segmentation masks.

To assess the similarity score distribution, we also report similarity score heatmaps, Fig. 4.4 and Fig. 3.3). From these results, two key observations can be drawn: for *Scheme 1*, the highest score is assigned to the instance of the largest size, regardless of the semantic meaning of the query. Secondly, with the incorporation of updated feature fusion formulations Eq. 3.10 and Eq. 3.9, we can observe improvement in the recall of instance masks along with similarity score distribution, with *Scheme 4* resulting in overall better performance.

Additionally, in Fig. 4.5 and Sec. B.3, we provide a comprehensive set of examples showcasing similarity score distribution obtained with *Scheme 4* over a diverse set of queries across all query types.

Feature Fusion	Replica [44]				ScanNet [4]			
	Inst.	Aff.	Prop.	Rel.	Inst.	Aff.	Prop.	Rel.
Scheme 1	0.8	0.7	0.7	0.3	0.8	**0.8**	0.7	0.4
Scheme 2	0.8	0.7	**0.9**	0.5	**0.9**	0.7	**0.8**	0.6
Scheme 3	**0.9**	**0.9**	**0.9**	**0.6**	**0.9**	**0.8**	0.7	0.6
Scheme 4	0.8	**0.9**	**0.9**	**0.6**	**0.9**	0.7	0.7	**0.7**

Table 4.7: **Qualitative evaluation of feature fusion schemes on open vocabulary instance retrieval performance**. Accuracy of feature fusion schemes for instance retrieval with “Inst.” (instance), “Aff.” (affordance), “Prop.” (Property). and “Rel.” (relative) text queries, as assessed by a human evaluator.

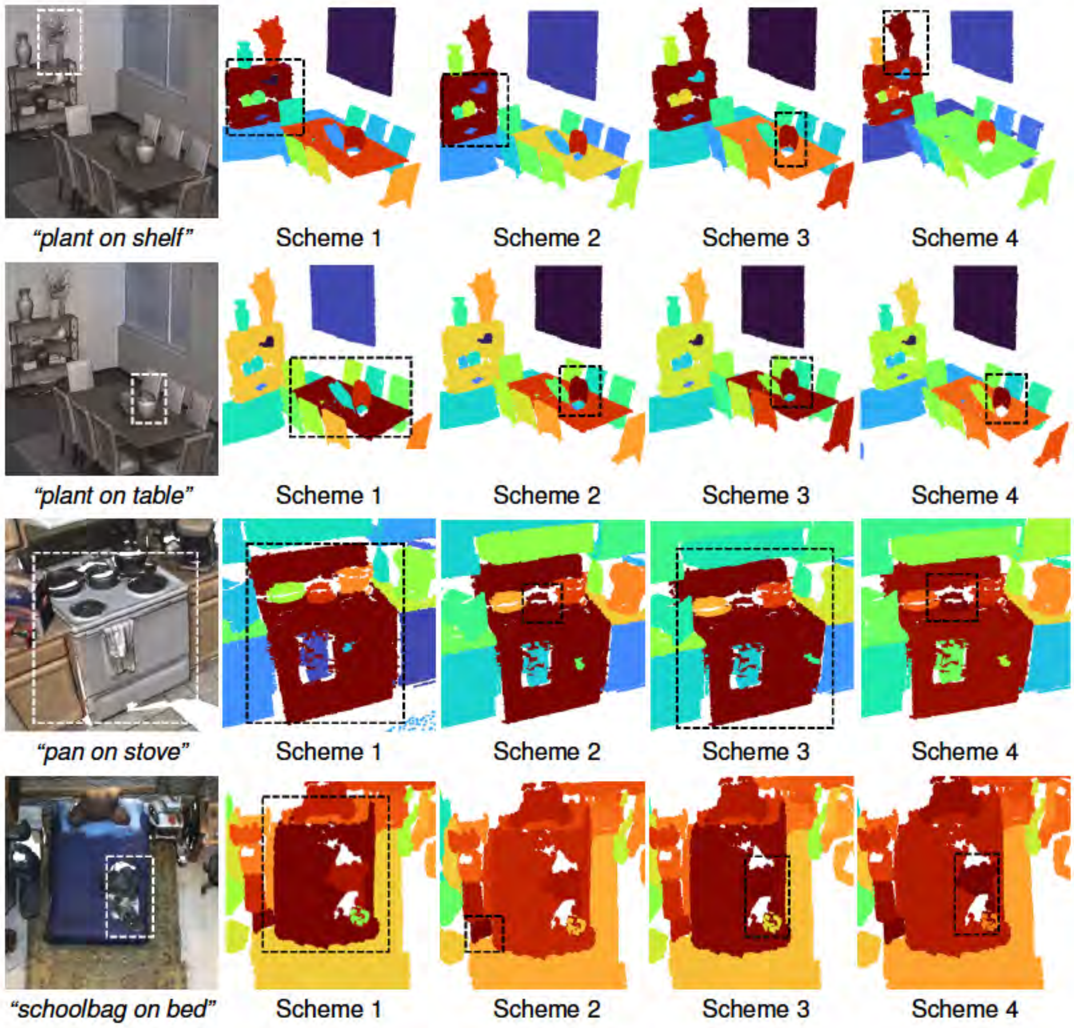


Figure 4.4: **Heatmap representing similarity between 'Relative' text queries and scene-instances for each feature fusion scheme.** For a given text query (left), a comparison of per instance cosine similarity scores for each feature fusion scheme. 'Dark red' represents maximum similarity, and 'dark blue' indicates minimum similarity. The 'white box' highlights the target object and the 'black box' highlights the object with maximum similarity score for the respective feature fusion schemes.

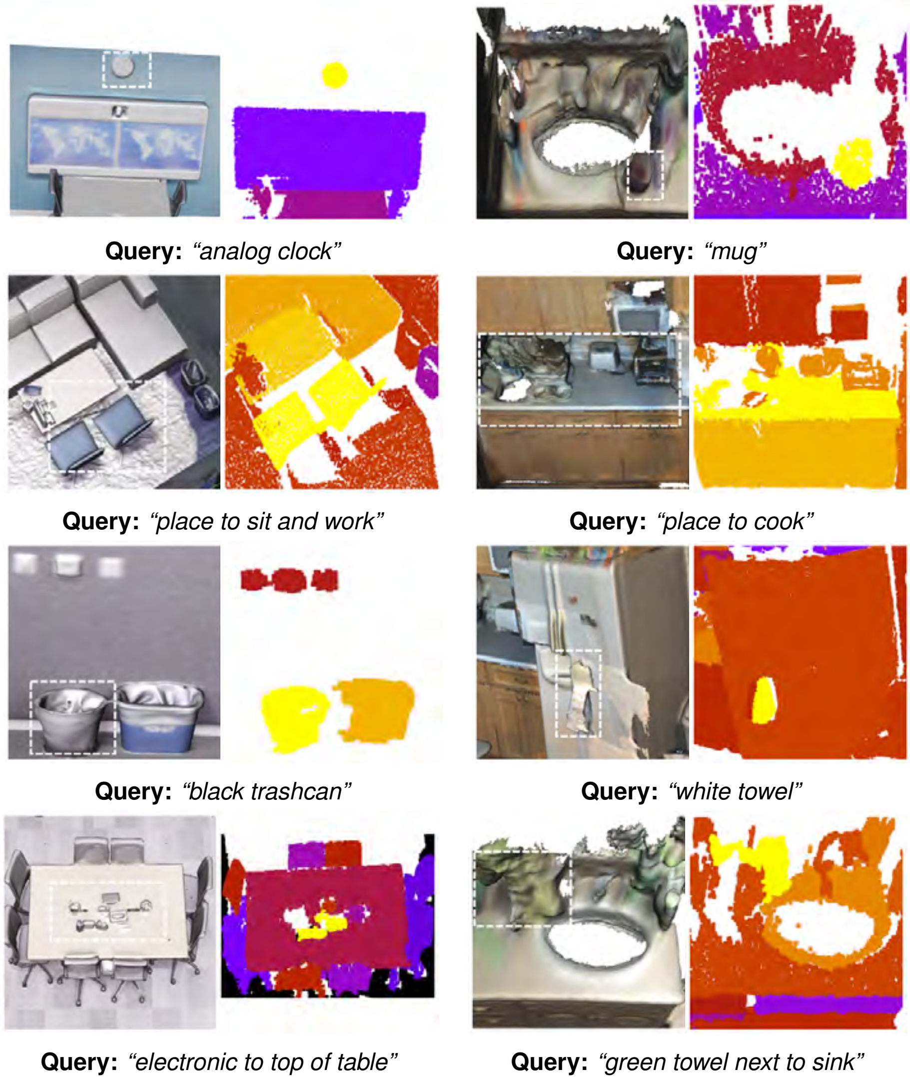


Figure 4.5: **Heatmasp representing similarity between text queries and scene-instances for different query type.** Visualization of cosine similarity score of each instance for given 'instance' (1st Row), 'affordance' (2nd Row), 'property' (3rd Row) and 'relative' (4th Row) text queries. 'Light yellow' represents maximum similarity and 'dark blue' indicates minimum.

4.2.3 Open Set Annotation and Segmentation

To assess annotation accuracy, the labels assigned to each mask, both MAX Label (n) and LLM Label (n') were manually verified across all scenes of Replica [44] and ScanNet [4] scenes. Similarly, to evaluate the performance of open set segmentation, under and over merged segments were manually counted and classified as faulty merges to assess merge accuracy. Table 4.8 presents the overall average of the results, with detailed analyses provided in TableC.12 and Table C.11.

From Table 4.8, it is evident that utilizing LLM with resulted in more accurate annotations n' than directly assigning the name corresponding to the best image n. Including more images introduces redundancy (as observed in Sec. 4.2.1), leading to better name assignment. This redundancy also results in improved filtering of undesired instances, Sec. 4.1.2, thus causing a slight improvement in merge accuracy. Furthermore, labels generated with LLM n' are more concise than directly assigned labels n, as illustrated in Fig. 4.7.

Labels	Replica [44]		ScanNet [4]	
	Label Acc.	Merge Acc.	Label Acc.	Merge Acc.
MAX Label (n)	0.83	0.87	0.75	0.85
LLM Label (n')	**0.87**	**0.88**	**0.84**	**0.87**

Table 4.8: **Qualitative evaluation of segmentation and annotation accuracy**. For MAX Label (n) and LLM Label (n'), the annotation and merge accuracy of segmentation masks, as assessed by a human evaluator.

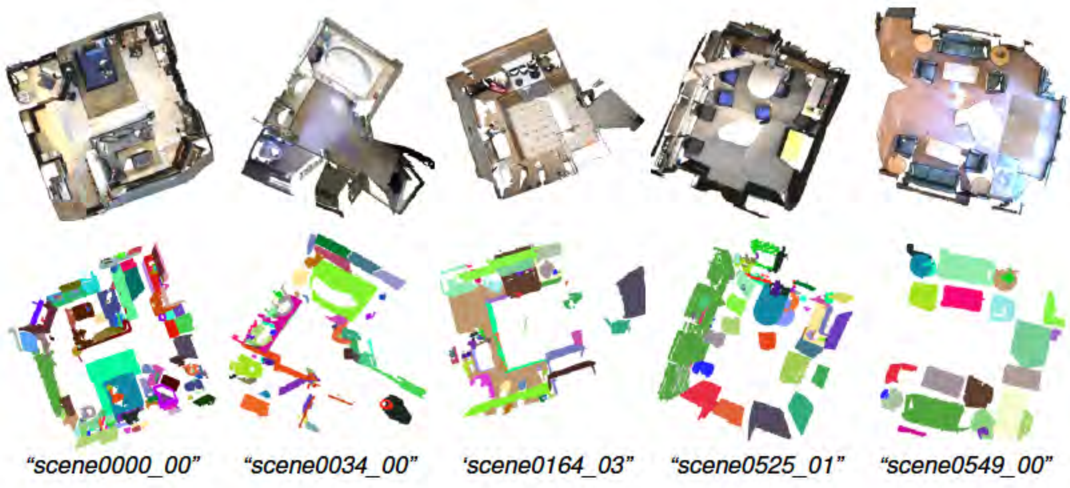


Figure 4.6: **Results for open set instance segmentation.** Bird-eye-view of ScanNet [4] scenes (top) and respective open set instance segmentation results (bottom).

Instance Segmentation

MAX Labels (n)

LLM Labels (n')

Figure 4.7: **Segmentation and annotation results.** Results for instance segmentation (top), MAX labels (n) (middle) and LLM labeled (n') (bottom) for Replica [44] *"room0"* scene.

4.2.4 Complex Spatial Queries

In order to assess the spatial reasoning capabilities, an exploratory study was conducted as discussed in Sec. 4.1.4. This study involved posing 70 complex spatial reasoning questions across scenes from both the Replica [44] and ScanNet [4] datasets. The results, as summarized in Table 4.9, indicate a trend where LLM exhibit significantly higher accuracy in smaller scenes, such as those found in Replica [44]. In contrast, performance tends to decline in larger scenes. This decrease in performance can be attributed to the higher incidence of faulty merges in larger scenes, as documented in Tables. C.12 and C.11.

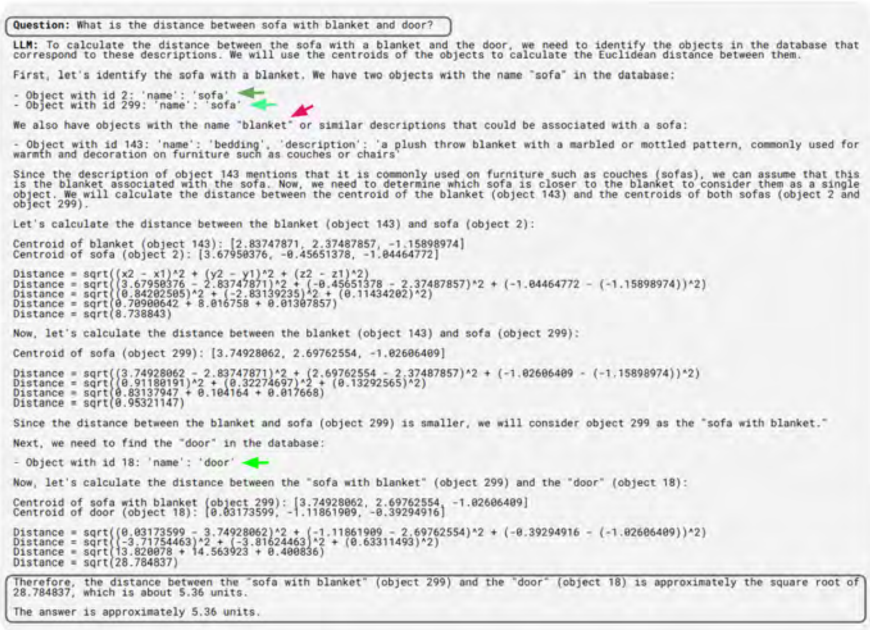
For smaller scenes, proposed strategy Sec. 3.6 showcased a marked improvement in handling complex spatial reasoning questions. An exemplary scenario depicted in Fig. 4.12 involves a question inquiring the distance between a sofa covered with a blanket and a door. The LLM employed a strategy as specified in Sec. 3.6. It first identifies sofas and blankets within the scene. Utilizing centroid calculations, the LLM discerns the sofa closest to the identified blanket. Even when the blanket was mislabeled as *"bedding"*, the caption provided descriptive and accurate information that facilitated the correct identification of the target object. Subsequently, the model locates the door and calculates the Euclidean distance between the door and the sofa with blanket.

This observation, along with additional examples in Sec. B.6 underscores the effectiveness of proposed prompting strategy for directly utilizing LLM for complex spatial reasoning tasks.

Dataset	Reasoning Acc.
Replica [44]	0.83
ScanNet [4]	0.68

Table 4.9: **Qualitative evaluation of spatial reasoning accuracy**. Overview of spatial reasoning accuracy with GPT4 [32] on the constructed scene, as assessed by human evaluator.

Figure 4.8: **Reference scene from Replica [44] dataset.** Reference image indicating multiple instances in Replica [44] *'room0'* scene, for spatial reasoning examples.



Question: What is the distance between sofa with blanket and door?

LLM: To calculate the distance between the sofa with a blanket and the door, we need to identify the objects in the database that correspond to these descriptions. We will use the centroids of the objects to calculate the Euclidean distance between them.

First, let's identify the sofa with a blanket. We have two objects with the name "sofa" in the database:

- Object with id 2: 'name': 'sofa'
- Object with id 299: 'name': 'sofa'

We also have objects with the name "blanket" or similar descriptions that could be associated with a sofa:

- Object with id 143: 'name': 'bedding', 'description': 'a plush throw blanket with a marbled or mottled pattern, commonly used for warmth and decoration on furniture such as couches or chairs'

Since the description of object 143 mentions that it is commonly used on furniture such as couches (sofas), we can assume that this is the blanket associated with the sofa. Now, we need to determine which sofa is closer to the blanket to consider them as a single object. We will calculate the distance between the centroid of the blanket (object 143) and the centroids of both sofas (object 2 and object 299).

Let's calculate the distance between the blanket (object 143) and sofa (object 2):

Centroid of blanket (object 143): [2.83747871, 2.37487857, -1.15898974]
Centroid of sofa (object 2): [3.67950376, -0.45651378, -1.04464772]

```
Distance = sqrt((x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2)
Distance = sqrt((3.67950376 - 2.83747871)^2 + (-0.45651378 - 2.37487857)^2 + (-1.04464772 - (-1.15898974))^2)
Distance = sqrt((0.84202505)^2 + (-2.83139235)^2 + (0.11434202)^2)
Distance = sqrt(0.70900642 + 8.016758 + 0.01307857)
Distance = sqrt(8.738843)
```

Now, let's calculate the distance between the blanket (object 143) and sofa (object 299):

Centroid of sofa (object 299): [3.74928062, 2.69762554, -1.02606409]

```
Distance = sqrt((3.74928062 - 2.83747871)^2 + (2.69762554 - 2.37487857)^2 + (-1.02606409 - (-1.15898974))^2)
Distance = sqrt((0.91180191)^2 + (0.32274697)^2 + (0.13292565)^2)
Distance = sqrt(0.83137947 + 0.104164 + 0.017668)
Distance = sqrt(0.95321147)
```

Since the distance between the blanket and sofa (object 299) is smaller, we will consider object 299 as the "sofa with blanket."

Next, we need to find the "door" in the database:

- Object with id 18: 'name': 'door'

Now, let's calculate the distance between the "sofa with blanket" (object 299) and the "door" (object 18):

Centroid of sofa with blanket (object 299): [3.74928062, 2.69762554, -1.02606409]
Centroid of door (object 18): [0.03173599, -1.11861909, -0.39294916]

```
Distance = sqrt((0.03173599 - 3.74928062)^2 + (-1.11861909 - 2.69762554)^2 + (-0.39294916 - (-1.02606409))^2)
Distance = sqrt((-3.71754463)^2 + (-3.81624463)^2 + (0.63311493)^2)
Distance = sqrt(13.820078 + 14.563923 + 0.400836)
Distance = sqrt(28.784837)
```

Therefore, the distance between the "sofa with blanket" (object 299) and the "door" (object 18) is approximately the square root of 28.784837, which is about 5.36 units.

The answer is approximately 5.36 units.

Figure 4.9: **Spatial reasoning process of LLM.** Response of the LLM (GPT4 [32]), to a spatial reasoning question. 'Black box' highlights question & answer, and 'colored arrows' indicated instances as referred to in Fig. 4.8.

4.3 Scope and Prospects

4.3.1 Limitations

The analysis of results reveals that the performance of the method is primarily hindered by two critical factors: model limitations and merging faults. The proposed approach is fundamentally reliant on two foundational models: CLIP [36], for image-text association, and GroundedSAM [39], for generating instance-level 2D masks. As depicted in Fig. 4.1, the performance of CLIP varies across different model sizes and architectures, indicating that the efficacy of image-text associations is contingent upon the specific characteristics of the foundation model employed. Thus, the open vocabulary accuracy is inherently bounded by the accuracy of the base 2D foundation models, with one notable failure case illustrated in Fig. 4.10. Similarly, 2D instance segmentation performance of GroundedSAM [39] is restricted by the capabilities of the base models [20, 28, 59] being used. Nonetheless, the method's model-agnostic nature implies that advancements in foundational models are likely to directly enhance the system's overall performance.

Figure 4.10: **Failure case, incorrect instance recall.** Example, indicating wrong instance recall for given relative text query. 'Light yellow' represents maximum similarity and 'dark blue' indicates minimum.

Merging faults present another challenge. During the testing phase, a consistent set of hyperparameters across all scenarios were employed. Given the method's reliance on an overlap-based merging strategy, varying external scene conditions necessitate different threshold requirements for effective overlap determination. Hence, exploring voxel based [8] or neural representations [19] could offer better handling of merging faults. Additionally, a post-processing-based CLIP similarity [9] can be employed to handle under-segmentation.

The repercussions of merging faults extend to a decrease in the accuracy of annotations and spatial reasoning. Fig. 4.11 highlights different cases of merging flaws. For example, in the case of over-merging where two throw pillows were incorrectly merged into a single entity, it leads to an inaccurate response from the language model regarding the total number of pillows, as shown in Fig. 4.10. This example underscores the intricate challenges posed by merging faults and highlights the necessity for refined merging strategies to bolster the reliability and accuracy of spatial reasoning and annotation in complex scenes.

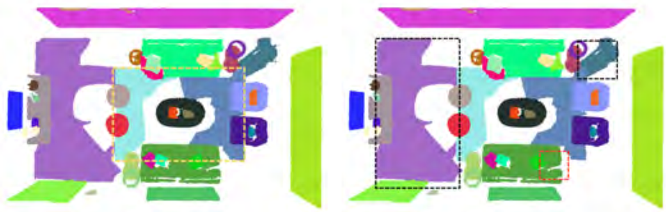
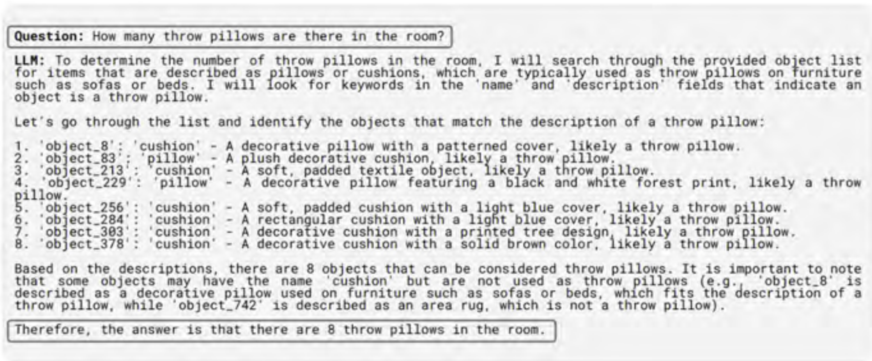


Figure 4.11: **Segmentation Flaws.** This figure highlights different cases of faulty merging. ‘Black box’ shows unfiltered/undesired masks, ‘yellow box’ and ‘red box’ indicates under segmentation and over segmentation respectively



Question: How many throw pillows are there in the room?

LLM: To determine the number of throw pillows in the room, I will search through the provided object list for items that are described as pillows or cushions, which are typically used as throw pillows on furniture such as sofas or beds. I will look for keywords in the 'name' and 'description' fields that indicate an object is a throw pillow.

Let's go through the list and identify the objects that match the description of a throw pillow:

1. 'object_8': 'cushion' - A decorative pillow with a patterned cover, likely a throw pillow.
2. 'object_83': 'pillow' - A plush decorative cushion, likely a throw pillow.
3. 'object_213': 'cushion' - A soft, padded textile object, likely a throw pillow.
4. 'object_229': 'pillow' - A decorative pillow featuring a black and white forest print, likely a throw pillow.
5. 'object_256': 'cushion' - A soft, padded cushion with a light blue cover, likely a throw pillow.
6. 'object_284': 'cushion' - A rectangular cushion with a light blue cover, likely a throw pillow.
7. 'object_303': 'cushion' - A decorative cushion with a printed tree design, likely a throw pillow.
8. 'object_378': 'cushion' - A decorative cushion with a solid brown color, likely a throw pillow.

Based on the descriptions, there are 8 objects that can be considered throw pillows. It is important to note that some objects may have the name 'cushion' but are not used as throw pillows (e.g., 'object_8' is described as a decorative pillow used on furniture such as sofas or beds, which fits the description of a throw pillow, while 'object_742' is described as an area rug, which is not a throw pillow).

Therefore, the answer is that there are 8 throw pillows in the room.

Figure 4.12: **Incorrect spatial reasoning.** Incorrect response by the LLM (GPT-4 [32]) to a spatial question. ‘Black box’ highlights question answer.

4.3.2 Future Research

As highlighted in Section 4.3.1, higher order representation, like neural [19, 35] or voxel based [8, 40], can enhance segmentation performance and improving downstream applications. The research area of open world 3D understanding, is still in early stages. The focus of the current study was to investigate foundational research questions. Further research into applying the findings of the study to engineering challenges, such as outdoor scenes and large-scale building modeling, presents a substantial opportunity for practical applications.

Utilizing 3D asset generation models such as Point-to-3D [56], proposed method can be used for novel application like 3D scan to digital twin. The constructed scene representation can provide grounding for these models by using information associated to each instance (point clouds, bounding boxes, captions, etc.). Additionally, integrating the proposed method into a SLAM pipeline could facilitate the creation of temporal record of instances. Combining this with a Large Language Model, featuring a large context window, such as Gemini 1.5 [38], could enable 3D spatio-temporal reasoning [50], thereby enhancing understanding of the dynamic 4D world.

4.3.3 Potential Use Cases

The proposed creates an open set representation of 3D environment with open vocabulary understanding capabilities. This offers significant benefits across various applications that require interaction with the 3D world. Key use cases include:

- **Digital Engineering Tools:** The method can uplift tools like Digital Twins, Building Information Models, 3D Scans etc. by enabling open vocabulary interactions, which opens up new possibilities for modeling and understanding physical spaces.
- **AR/VR Applications:** A detailed spatial record of instances, with open vocabulary features, can improve the realism and interactivity of augmented and virtual reality experiences.
- **Robotics:** The proposed approach enables open set spatial understanding and reasoning crucial for generalized robotic tasks such as manipulation, navigation, and localization. This allows robots to interact with a broader range of objects and settings, promoting greater autonomy and versatility in robotic systems.

4.4 Conclusion

In conclusion, this study presents a framework designed to address the current challenges in the domain of open world 3D scene understanding. By introducing an incremental and scalable approach, the method counters the limitations of current methodologies that are predominantly non-incremental, and struggle with scalability and contextualization in responding to complex queries. The proposed innovative approach capitalizes on the strengths of 2D foundation models, utilizing them to progressively build detailed instance-level open set 3D scene representations. The method links 2D space to 3D, by efficiently tracking and associate instance-specific information, such as feature vectors, names, captions etc.

Moreover, this study proposes feature fusion schemes, which significantly bolster the model's ability to contextualize and interpret complex relational queries. The application of large language models for automatic annotation and advanced spatial reasoning tasks further exemplifies the versatility and robustness of the proposed method. Through comprehensive evaluations conducted on scenes from ScanNet [4, 41] and Replica [44] datasets, the proposed method has demonstrated zero-shot generalization capabilities, that outperforming current state-of-the-art solutions.

Future research may explore further downstream applications based on the proposed representation, including, establishing temporal record of instances for spatio-temporal reasoning and application on large scale outdoor environments.

Appendix A

Algorithms

Algorithm 1: 2D Feature Extraction

Require: $\mathcal{I} = \{I_0, I_1, I_2 \ldots, I_n\}$, a set of RGB images; Parameters: $S_r = \{s_r\}_k$ (Scaling ratios), px (Border Pixels), th (GSAM threshold)

Output: Per instance details ID : $\{n_{ij}, c_{ij}, bb_{ij}, s_{pred,ij}, f_{ij}, f_i^g\}$; Per image updated mask M_i'

$\mathcal{I}' = \{I_0', I_s', I_{2s}' \ldots, I_n\} \leftarrow$ sampleImages$(\mathcal{I}, s)$
ID $\leftarrow 0$
for all $I_i' \in \mathcal{I}'$ **do**
  $M_i, BB_i, S_{pred,i} \leftarrow$ groundedSAM(I_i', th)
  $f_i^g \leftarrow$ extractFeatures(I_i')
  $M_i' \leftarrow \{\}$
  for all $\{(m_{ij}, bb_{ij}, s_{pred,ij}) \mid m_{ij} \in M_i, bb_{ij} \in B_i, s_{pred,ij} \in S_{pred,i}\}$ **do**
    $n_{ij}, c_{ij} \leftarrow$ getName&Caption(crop(I_i', bb_{ij}))
    $M_i' \leftarrow M_i' \cup$ updateMasks$(m_{ij}$, ID$, px)$.
    ID $\leftarrow$ ID $+ 1$
    $F_{ij} \leftarrow \{$extractFeatures(crop$(I_i'$, scale$(bb_{ij}, s_r))) \mid s_r \in S_r\}$
    $f_{ij} \leftarrow$ multiScaleFeatureFusion(F_{ij})
    output for ID : $n_{ij}, c_{ij}, bb_{ij}, s_{pred,ij}, f_{ij}, f_i^g$
  end for
  output for M_i'
end for

Algorithm 2: $2D \rightarrow 3D$ Fuse & Track

Require: $\mathcal{I}' = \{I'_0, I'_s, I'_{2s} \dots, I_n\} \leftarrow$ sampleImages$(\mathcal{I}, s)$ set of sampled RGB images; Set of corresponding Depth map $\{D\}_n$; Global poses $\{T\}_n$; Updated masks $\{M'\}_n$; Camera intrinsic K.

Output: 3D Point Cloud $\mathcal{P}_{\text{scene}} \in \mathbb{R}^{x,y,z,\text{ID}}$ with updated ID; Track of overlapping id $\mathcal{Q} : \mathcal{Q} \mapsto \{\text{ID} \in \texttt{unique}(\text{ID} \in \mathcal{P}_{\text{scene}}) : \{\text{ID} \in \{M'\}\}\}$

$\mathcal{Q} \leftarrow \{\}$ **global**
$\mathcal{P}_{\text{scene}} \leftarrow \{\}$
for all $I'_i \in \mathcal{I}'$ **do**
  $K, T_i, D_i, M'_i \leftarrow$ Retrive(I'_i)
  $\mathcal{P}_{\text{frame}} = \{\}$
  for each pixel (u, v) in I_i **do**
    $\mathbf{p} = T_i \cdot (D_i(u,v) \cdot K^{-1} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix})$, $\text{ID} = M_i(u, v)$.
    $\mathcal{P}_{\text{frame}} \leftarrow \mathcal{P}_{\text{frame}} \cup \{(\mathbf{p}, \text{ID})\}$
  end for
  $\{(\mathbf{i}_{\text{frame}}, \mathbf{i}_{\text{scene}})\} =$ overlappingPointsPairs$(\mathcal{P}_{\text{frame}}, \mathcal{P}_{\text{scene}})$
  $\mathcal{Q}, \mathcal{P}'_{\text{frame}} =$ update&TrackIDs$(\mathcal{P}_{\text{frame}}, \mathcal{P}_{\text{scene}}, \{(\mathbf{i}_{\text{frame}}, \mathbf{i}_{\text{scene}})\}, \mathcal{Q})$
  $\mathcal{P}_{\text{scene}} = \mathcal{P}_{\text{scene}} \cup \mathcal{P}'_{\text{frame}}$
end for

return Final Point Cloud: $\mathcal{P}_{\text{scene}}$; Track of ID's: $\mathcal{Q}$

Algorithm 3: Overlapping Point Pairs

Require: Point clouds $\mathcal{P}_f$, $\mathcal{P}_s$: $\mathcal{P}_s > \mathcal{P}_f$ & $\mathcal{P}_f, \mathcal{P}_s \in \mathbb{R}^{x,y,z,\text{ID}}$; Parameters: ϵ (search distance)

Output: Set of index pairs of overlapping points $\{(i, j)\}$

Define $d(\cdot, \cdot)$ as the Euclidean distance function.

Bounds $\leftarrow [\min(\mathcal{P}_{f,x}) - \epsilon, \max(\mathcal{P}_{f,x}) + \epsilon] \times [\min(\mathcal{P}_{f,y}) - \epsilon, \max(\mathcal{P}_{f,y}) + \epsilon] \times [\min(\mathcal{P}_{f,z}) - \epsilon, \max(\mathcal{P}_{f,z}) + \epsilon]$
Filter $\mathcal{P}_s$ to $\mathcal{P}'_s$ where $\mathcal{P}'_s = \{\mathbf{p}_i \in \mathcal{P}_s | \mathbf{p}_i \in \text{Bounds}\}$
Construct: KDTree $\leftarrow \mathcal{P}'_s$
$\mathcal{L} = \{\}$
for $\mathbf{p}_i \in \mathcal{P}_f$ **do**
  Find $\mathbf{q}$ in $\mathcal{P}'_s$ using KDTree such that $d(\mathbf{p}_i, \mathbf{q}) < \epsilon$
  if $\mathbf{q}$ exists **then**
    $\mathcal{L} = \mathcal{L} \cup (i, j)$, where j corresponds to the index of $\mathbf{q} \in \mathcal{P}_s$
  end if
end for
return Set of index pairs $\mathcal{L} = \{(i, j)\}$

Algorithm 4: Update and Track IDs

Require: Point clouds $\mathcal{P}_f$, $\mathcal{P}_s$: $\mathcal{P}_s > \mathcal{P}_f$ & $\mathcal{P}_f, \mathcal{P}_s \in \mathbb{R}^{x,y,z,\mathrm{ID}}$; Set of index pairs of overlapping points $\mathcal{L} = \{(i,j)\}$; Parameters: ρ (overlap ratio threshold)

Output: 3D Point Cloud $\mathcal{P}_\mathrm{f} \in \mathbb{R}^{x,y,z,\mathrm{ID}}$ with updated IDs; Track of overlapping IDs $\mathcal{Q} : \mathcal{Q} \mapsto \{\mathrm{ID} \in \texttt{unique}(\mathrm{ID} \in \mathcal{P}_{\mathrm{scene}}) : \{\mathrm{ID} \in \{M'\}\}\}$

Initialization:
Get list of unique ids with their counts
$U_{\mathcal{P}_s}, C_{\mathcal{P}_s} \leftarrow$ getUniqueCount(ID $\in \mathcal{P}_s$)
$U_{\mathcal{P}_f}, C_{\mathcal{P}_f} \leftarrow$ getUniqueCount(ID $\in \mathcal{P}_f$)
$\mathcal{G}_{\mathrm{overlap}} \leftarrow \{\}$
$\mathcal{Q} \leftarrow$ **global**

Record Counts of Overlapping Points in a Mask:
for each $(i_{\mathcal{P}_f}, i_{\mathcal{P}_s})$ in $\mathcal{L}$ **do**
  $\mathrm{id}_{\mathcal{P}_f} \leftarrow \mathrm{ID} \in \mathcal{P}_f[i_{\mathcal{P}_f}]$
  $\mathrm{id}_{\mathcal{P}_s} \leftarrow \mathrm{ID} \in \mathcal{P}_s[i_{\mathcal{P}_s}]$
  if $\mathrm{id}_{\mathcal{P}_f} \notin \mathcal{G}_{\mathrm{overlap}}$ **then**
    $\mathcal{G}_{\mathrm{overlap}}[\mathrm{id}_{\mathcal{P}_f}] \leftarrow \{\}$
  end if
  if $\mathrm{id}_{\mathcal{P}_s} \notin \mathcal{G}_{\mathrm{overlap}}[\mathrm{id}_{\mathcal{P}_f}]$ **then**
    $\mathcal{G}_{\mathrm{overlap}}[\mathrm{id}_{\mathcal{P}_f}][\mathrm{id}_{\mathcal{P}_s}] \leftarrow 0$
  end if
  $\mathcal{G}_{\mathrm{overlap}}[\mathrm{id}_{\mathcal{P}_f}][\mathrm{id}_{\mathcal{P}_s}] \leftarrow + 1$
end for

Point Cloud ID Update with History Tracking:
for each $\mathrm{id}_{\mathcal{P}_f}$ in $\mathcal{G}_{\mathrm{overlap}}$ **do**
  $\mathrm{id}^*_{\mathcal{P}_s} \leftarrow \underset{\mathrm{id}_{\mathcal{P}_s} \in U_{\mathcal{P}_s}}{\operatorname{argmax}} \mathcal{G}_{\mathrm{overlap}}[\mathrm{id}_{\mathcal{P}_f}][\mathrm{id}_{\mathcal{P}_s}]$
  $\rho \leftarrow \frac{\mathcal{G}_{\mathrm{overlap}}[\mathrm{id}_{\mathcal{P}_f}][\mathrm{id}^*_{\mathcal{P}_s}]}{\min(C_{\mathcal{P}_s}[\mathrm{id}^*_{\mathcal{P}_s}], C_{\mathcal{P}_f}[\mathrm{id}_{\mathcal{P}_f}])}$
  if $\rho >$ threshold **then**
    $\forall\, \mathrm{ID} \in \mathcal{P}_f : \mathrm{ID} = \mathrm{id}_{\mathcal{P}_f} \implies \mathrm{ID} \leftarrow \mathrm{id}^*_{\mathcal{P}_s}$
    if $\mathrm{id}^*_{\mathcal{P}_s}$ do not exists in $\mathcal{Q}$ **then**
      $\mathcal{Q} \leftarrow \mathrm{id}^*_{\mathcal{P}_s}$
      $\mathcal{Q}[\mathrm{id}^*_{\mathcal{P}_s}] \leftarrow \{\mathrm{id}^*_{\mathcal{P}_s}\}$
    end if
    $\mathcal{Q}[\mathrm{id}^*_{\mathcal{P}_s}] \leftarrow \mathcal{Q}[\mathrm{id}^*_{\mathcal{P}_s}] \cup \mathrm{id}_{\mathcal{P}_f}$
    continue
  end if
  $\mathcal{Q} \leftarrow \mathrm{id}_{\mathcal{P}_f}$
  $\mathcal{Q}[\mathrm{id}_{\mathcal{P}_f}] \leftarrow \{\mathrm{id}_{\mathcal{P}_f}\}$
end for

return Updated point cloud $\mathcal{P}_f$; Updated tracked ID's $\mathcal{Q}$

Algorithm 5: Post Process Point Cloud

Require: 3D Point Cloud $\mathcal{P}_{\text{scene}} \in \mathbb{R}^{x,y,z,\text{ID}}$ with updated ID; Track of overlapping ID's $\mathcal{Q} : \mathcal{Q} \mapsto \{\text{ID} \in (\text{ID} \in \mathcal{P}_{\text{scene}}) : \{\text{ID} \in \{M'\}\}\}$; Set of features associated with each ID : $\{n, c, bb, s_{\text{pred}}, f, f^g\}$.

Output: $\mathcal{M} : \mathcal{M} \mapsto \{\{\mathcal{P} \in \mathbb{R}^{x,y,z,\text{ID}}, n, c, f, bb_{3D}, (x_c, y_c)\}_i | i \in \{\texttt{unique}(\text{ID} \in \mathcal{P}_{\text{scene}})\}\}$.

$\mathcal{M} \leftarrow \{\}$
for all $i \in \{\texttt{unique}(\text{ID} \in \mathcal{P}_{\text{scene}}\}$ **do**
  $\mathcal{P}_i \leftarrow \{p \in \mathcal{P}_i \mid \text{ID}(p) \in \text{ID}(\mathcal{P}_{\text{scene}})$ and $\text{ID}(p) = i\}$
  $\mathcal{P}_i \leftarrow$ filterDBSCAN$(\mathcal{P}_i)$
  $(x_c, y_c)_i \leftarrow$ getCentroid$(\mathcal{P}_i)$
  $bb_{3Di} \leftarrow$ getBoundingBox$(\mathcal{P}_i)$
  $N_i, C_i, F_i, S_{pred,i} \leftarrow \{\}, \{\}, \{\}, \{\}$
  for all $j \in \mathcal{Q}[ID_i \leftarrow i]$ **do**
    $n_{ij}, c_{ij}, s_{\text{pred}_{ij}}, f_{ij}, f^g_{ij} \leftarrow$ retrive$(ID_j \leftarrow j)$
    $N_i \leftarrow N_i \cup n_{ij}$
    $C_i \leftarrow C_i \cup c_{ij}$
    $S_{\text{pred}_i} \leftarrow S_{\text{pred}_i} \cup s_{\text{pred}_{ij}}$
    $F_i \leftarrow F_i \cup f_{ij}$
  end for
  $c_i \leftarrow C_i[argmax(S_{\text{pred}_i})]$
  $n_i \leftarrow N_i[argmax(S_{\text{pred}_i})]$
  $f_i \leftarrow$ multiviewFusion(F_i, f^g_i)
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathcal{P}_i, n_i, c_i, f_i, bb_{3Di}, (x_c, y_c)_i\}$
end for

Algorithm 6: Instance Retrieval Direct

Require: $\mathcal{M} : \mathcal{M} \mapsto \{\{\mathcal{P} \in \mathbb{R}^{x,y,z,\mathsf{ID}}, n, c, F, f^g, bb_{3D}, (x_c, y_c)\}_i | i \in \{\texttt{unique}(\mathsf{ID} \in \mathcal{P}_{\mathsf{scene}})\}\}$; Text query $\mathcal{K}$; Parameters: m (no of top images)

Output: $\{\mathcal{P} \in \mathbb{R}^{x,y,z,\mathsf{ID}}, n, c, F, bb_{3D}, (x_c, y_c)\}_i$ corresponding to text $\mathcal{K}$

$f_{\mathcal{K}} \leftarrow$ textEncoder$(\mathcal{K})$
$S_{\mathsf{score}} \leftarrow \{\}$
for all $F_i \in \mathcal{M}$ **do**
  $S'_{\mathsf{score}} \leftarrow \{\}$
  for all $f_{ij} \in F_i$ **do**
    $f'_{ij} \leftarrow \left(f_{ij} + \left(\frac{f_{ij} \cdot f_i^g}{\max(\|f_{ij}\|_2 \cdot \|f_i^g\|_2)}\right) \cdot f^g\right)$
    $S'_{\mathsf{score}} \leftarrow S'_{\mathsf{score}} \cup (\frac{f'_{ij} \cdot f_{\mathcal{K}}}{\max(\|f'_{ij}\|_2 \cdot \|f_{\mathcal{K}}\|_2)})$
  end for
  $S_{\mathsf{score}} \leftarrow S_{\mathsf{score}} \cup \frac{1}{m} \sum_{i=0}^{m} (S'_{\mathsf{score}}[argmax(S'_{\mathsf{score}}, m)])$
end for
return $\mathcal{M}[argmax(S_{\mathsf{score}})]$

Algorithm 7: Instance Retrieval

Require: $\mathcal{M} : \mathcal{M} \mapsto \{\{\mathcal{P} \in \mathbb{R}^{x,y,z,\mathsf{ID}}, n, c, f, bb_{3D}, (x_c, y_c)\}_i | i \in \{\texttt{unique}(\mathsf{ID} \in \mathcal{P}_{\mathsf{scene}})\}\}$; Text query $\mathcal{K}$

Output: $\{\mathcal{P} \in \mathbb{R}^{x,y,z,\mathsf{ID}}, n, c, f, bb_{3D}, (x_c, y_c)\}_i$ corresponding to text $\mathcal{K}$

$f_{\mathcal{K}} \leftarrow$ textEncoder$(\mathcal{K})$
$S_{\mathsf{score}} \leftarrow \{\}$
for all $f_i \in \mathcal{M}$ **do**
  $S_{\mathsf{score}} \leftarrow S_{\mathsf{score}} \cup (\frac{f'_i \cdot f_{\mathcal{K}}}{\max(\|f'_i\|_2 \cdot \|f_{\mathcal{K}}\|_2)})$
end for
return $\mathcal{M}[argmax(S_{\mathsf{score}})]$

Appendix B

Additional Results

B.1 Comparison with Baseline Methods

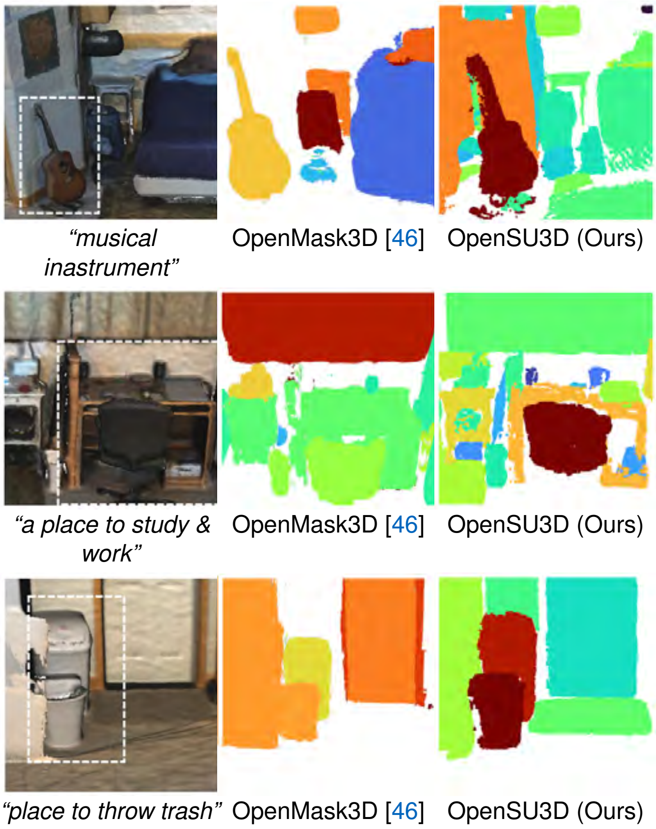


Figure B.1: **Heatmaps representing similarity between text queries and scene-instances.** For a given text query (left), comparison of per instance cosine similarity scores for OpenMask3D [46] (middle) & OpenSU3D (right) on ScanNet [4] scenes. ‘Dark red’ represents maximum similarity, and ‘dark blue’ indicates minimum similarity.

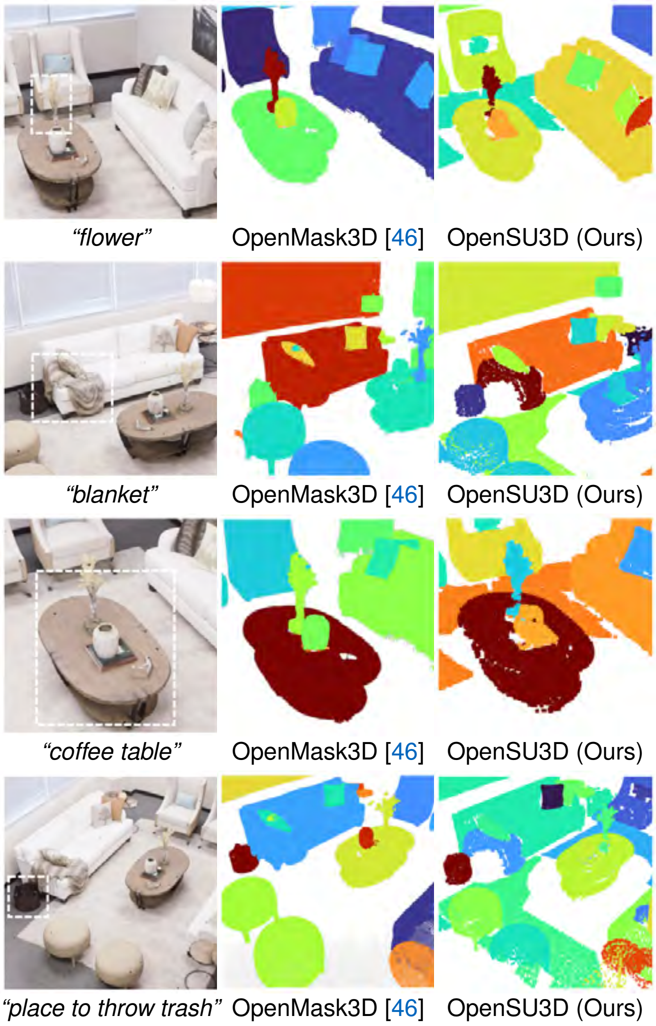


Figure B.2: **Heatmaps representing similarity between text queries and scene-instances.** For a given text query (left), comparison of per instance cosine similarity scores for OpenMask3D [46] (middle) & OpenSU3D (right) on replica [44] scenes. 'Dark red' represents maximum similarity, and 'dark blue' indicates minimum similarity.

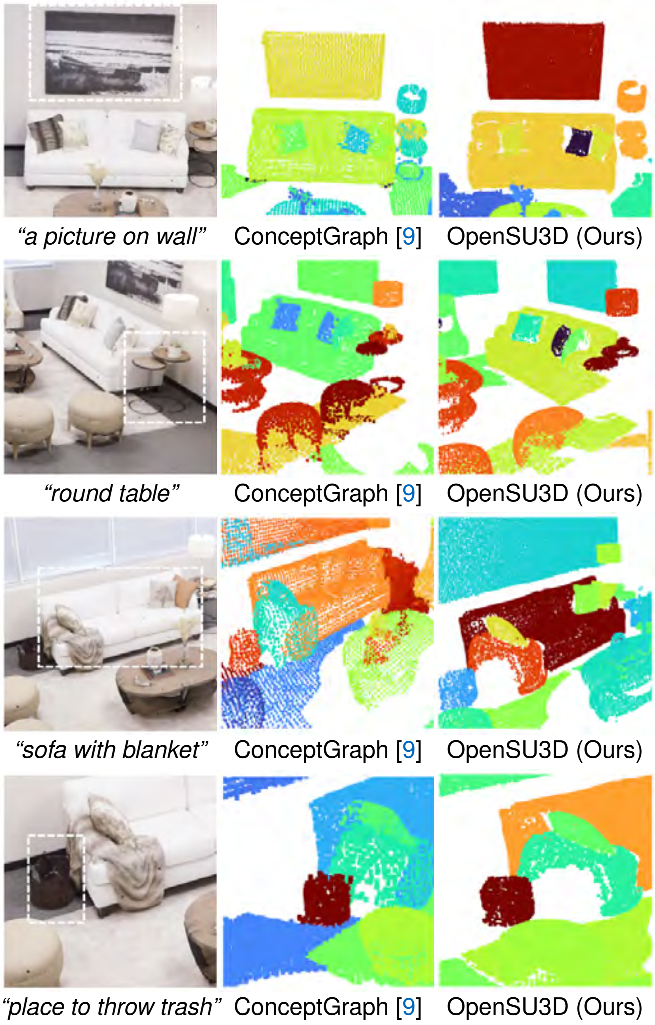


Figure B.3: **Heatmaps representing similarity between text queries and scene-instances.** For a given text query (left), comparison of per instance cosine similarity scores for ConceptGraph [9] (middle) & OpenSU3D (right) on replica [44] scenes. 'Dark red' represents maximum similarity, and 'dark blue' indicates minimum similarity.

B.2 Analysis of Feature Fusion Schemes

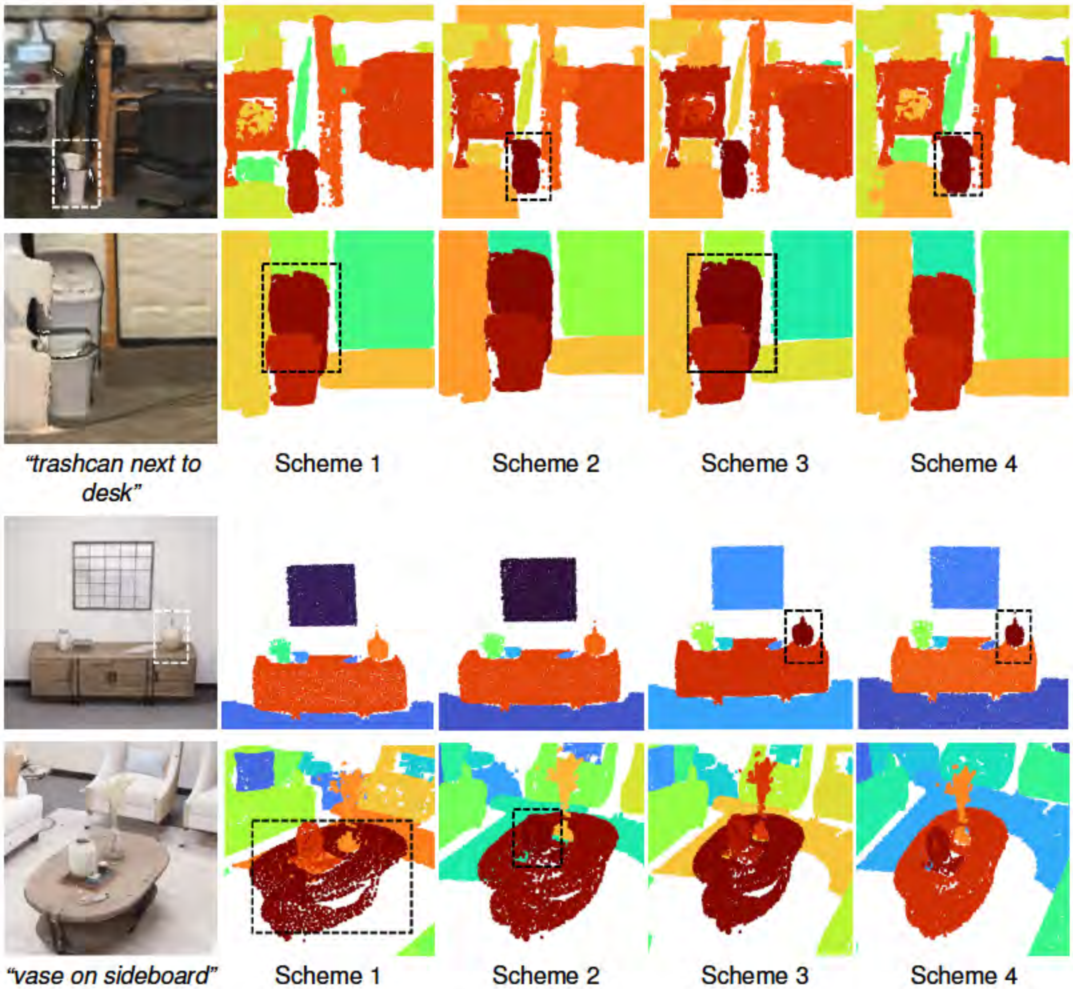


Figure B.4: **Heatmap representing similarity between 'Relative' text queries and scene-instances for each feature fusion scheme.** For a given text query (left), a comparison of per instance cosine similarity scores for each feature fusion scheme. 'Dark red' represents maximum similarity, and 'dark blue' indicates minimum similarity. The 'white box' highlights the target object and the 'black box' highlights the object with maximum similarity score for the respective feature fusion schemes.

B.3 Open Vocabulary Queries

B.3.1 Results for Instance Queries

Figure B.5: **Heatmaps representing similarity between text queries and scene-instances for different instance queries.** Visualization of cosine similarity score of each instance for a given ‘instance’ text queries, for Replica [44] scenes. ‘Light yellow’ represents maximum similarity and ‘dark blue’ indicates minimum.

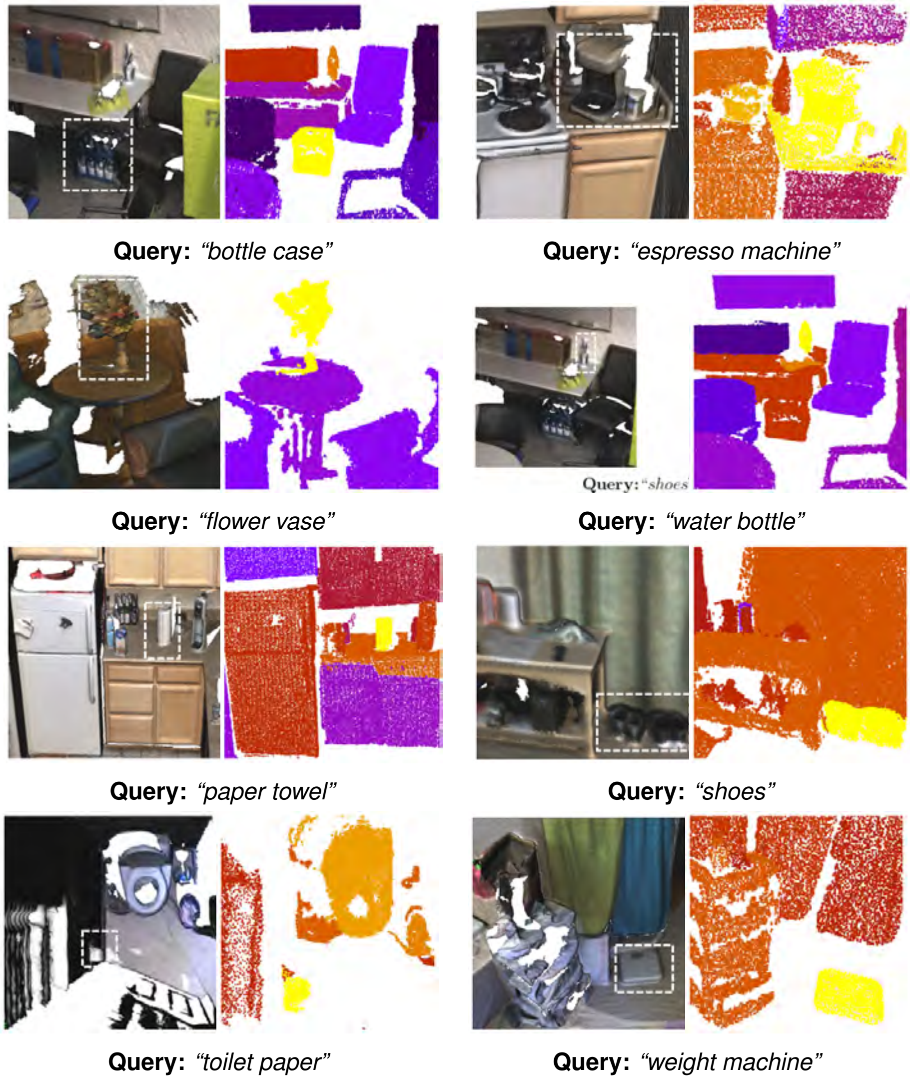


Figure B.6: **Heatmaps representing similarity between text queries and scene-instances for different instance queries.** Visualization of cosine similarity score of each instance for given ‘instance’ text queries, for ScanNet [4] scenes. ‘Light yellow’ represents maximum similarity and ‘dark blue’ indicates minimum.

B.3.2 Results for Affordance Queries

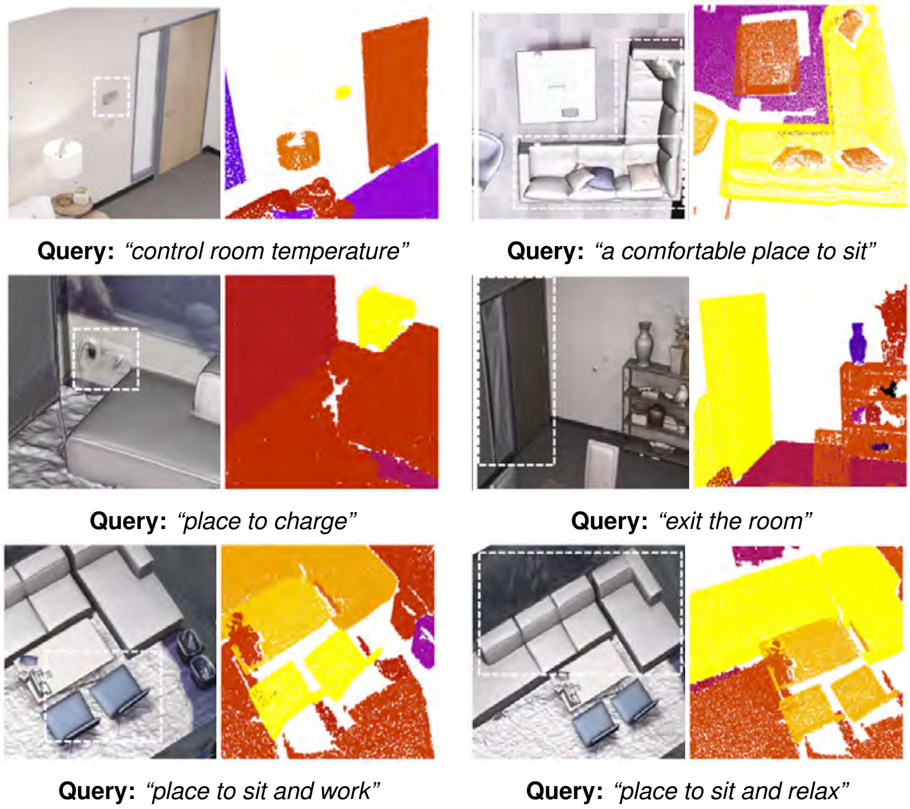


Figure B.7: **Heatmaps representing similarity between text queries and scene-instances for different affordance queries.** Similarity score for each instance (right) for given 'affordance' text query (left) for Replica [44] scenes. 'Light yellow' represents maximum similarity and 'dark blue' lowest.

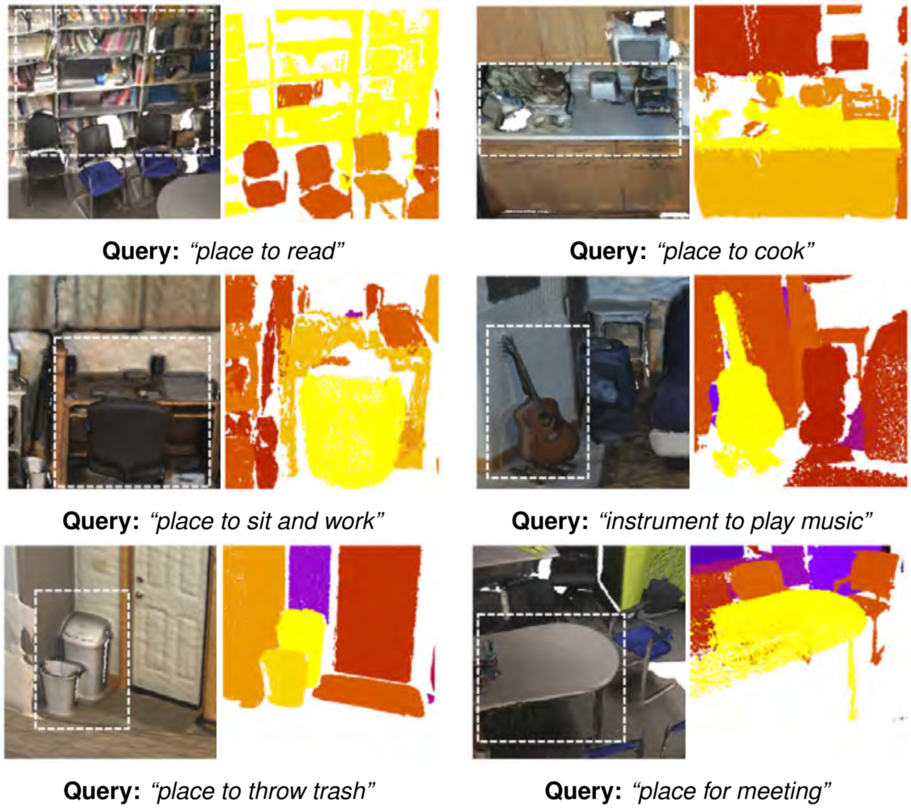


Figure B.8: **Heatmaps representing similarity between text queries and scene-instances for different affordance queries.** Visualization of cosine similarity score of each instance for given 'affordance' text queries, for ScanNet [4] scenes. 'Light yellow' represents maximum similarity and 'dark blue' indicates minimum.

B.3.3 Results for Property Queries

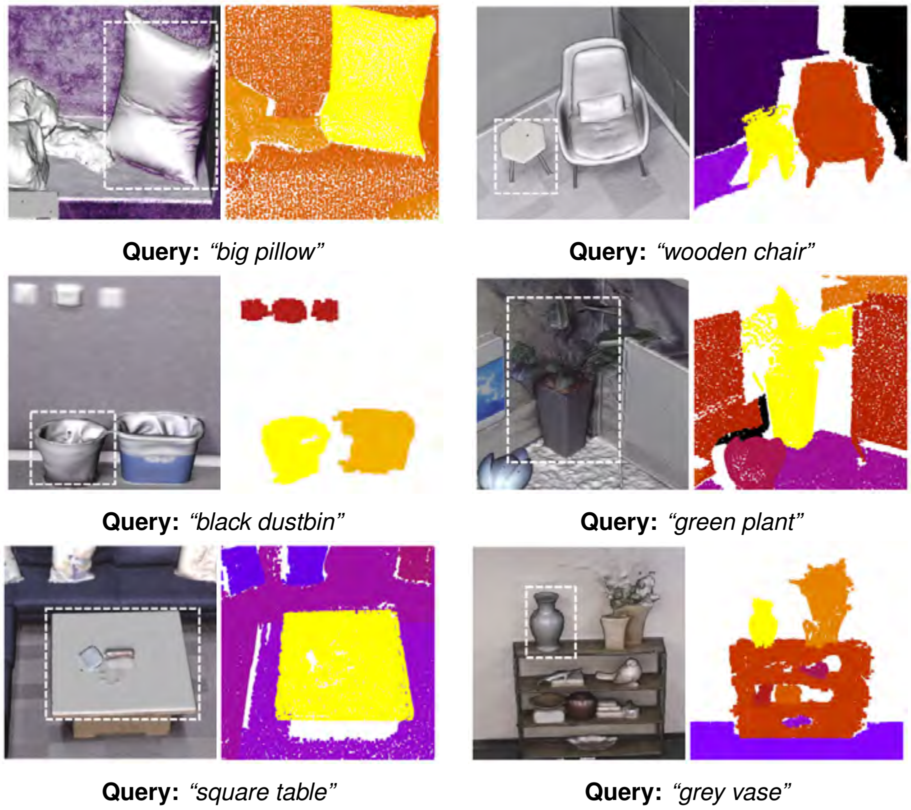
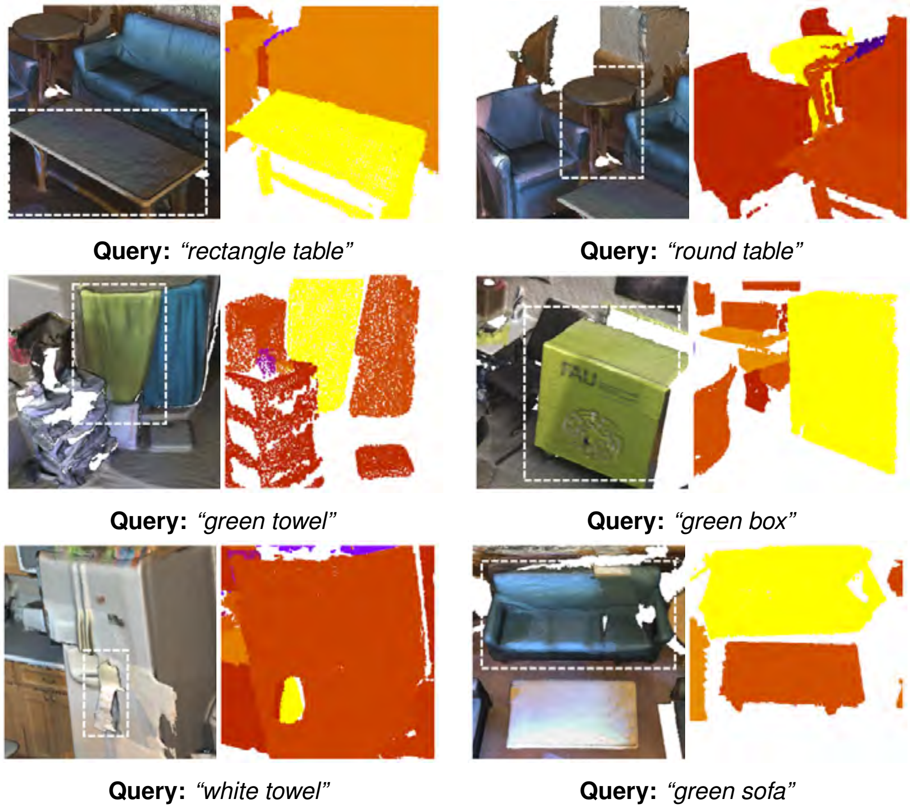


Figure B.9: **Heatmaps representing similarity between text queries and scene-instances for different property queries.** Visualization of cosine similarity score of each instance for given 'property' text queries, for Replica [44] scenes. 'Light yellow' represents maximum similarity and 'dark blue' indicates minimum.



Query: *"rectangle table"*

Query: *"round table"*

Query: *"green towel"*

Query: *"green box"*

Query: *"white towel"*

Query: *"green sofa"*

Figure B.10: **Heatmaps representing similarity between text queries and scene-instances for different property queries.** Visualization of cosine similarity score of each instance for given 'property' text queries, for ScanNet [4] scenes. 'Light yellow' represents maximum similarity and 'dark blue' indicates minimum.

B.3.4 Results for Relative Queries

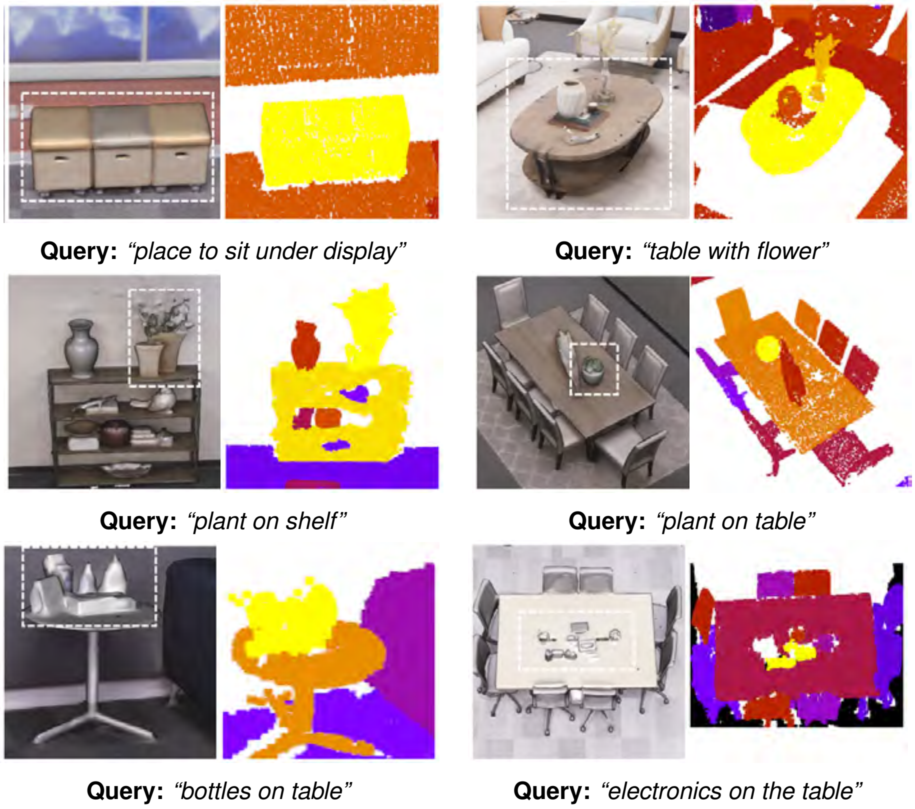


Figure B.11: **Heatmaps representing similarity between text queries and scene-instances for different relative queries.** Visualization of cosine similarity score of each instance for given 'relative' text queries, for Replica [44] scenes. 'Light yellow' represents maximum similarity and 'dark blue' indicates minimum.

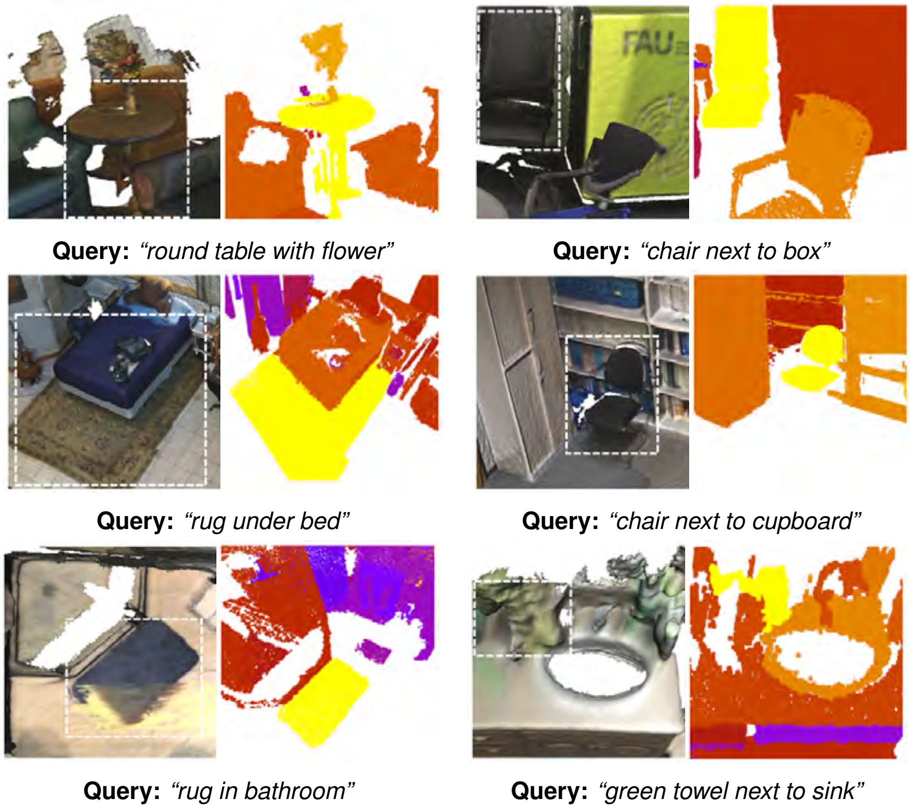


Figure B.12: **Heatmaps representing similarity between text queries and scene-instances for different relative queries.** Visualization of cosine similarity score of each instance for given 'relative' text queries, for ScanNet [4] scenes. 'Light yellow' represents maximum similarity and 'dark blue' indicates minimum.

B.4 Open Set Segmentation

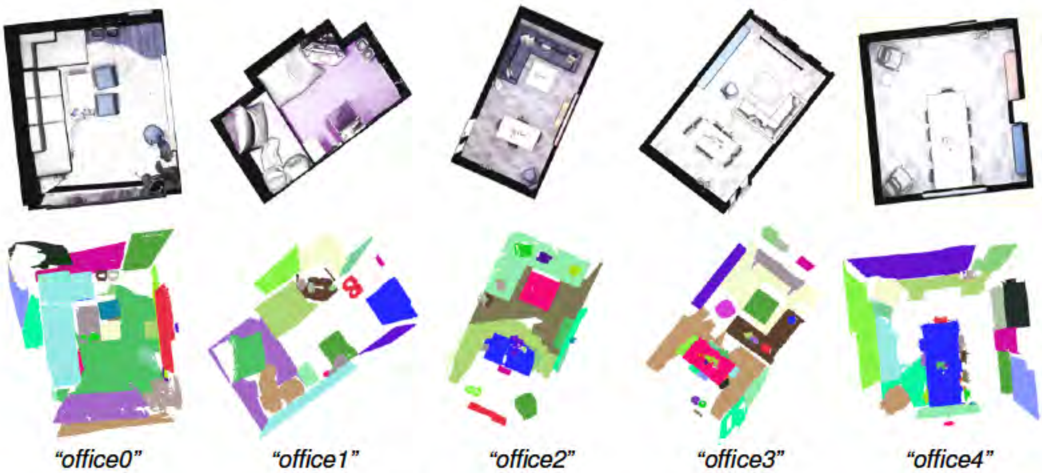


Figure B.13: **Results for open set instance segmentation.** Bird-eye-view of Replica [44] *"office"* (top) and respective open set instance segmentation results (bottom).

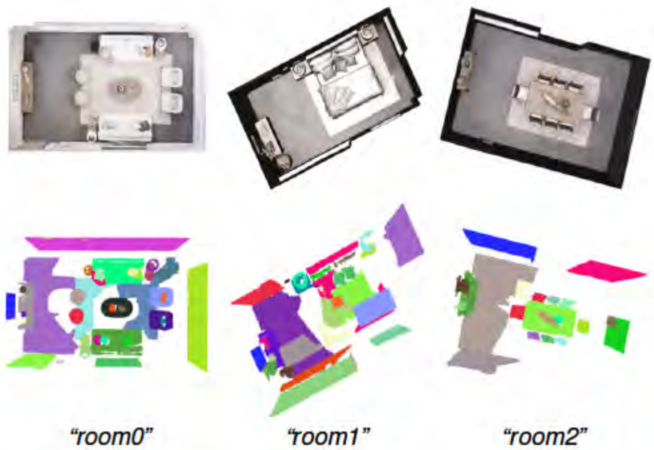


Figure B.14: **Results for open set instance segmentation.** Bird-eye-view of Replica [44] *"room"* (top) and respective open set instance segmentation results (bottom).

B.5 Open Set Annotation

Instance Segmentation (Reduced Masks)

MAX Labels (n)

LLM Labels (n')

Figure B.15: **Segmentation and annotation results.** Results for instance segmentation (top), MAX labels (n) (middle) and LLM labeled (n') (bottom) for ScanNet [4] *"scene0000_00"* scene. (Note: For visualization, segmentation masks are reduced.)

B.6 Spatial Reasoning

Question: Is there a vase on coffee table?

LLM: To determine if there is a vase with flowers on the coffee table, we need to identify objects that are described as a vase with flowers and a coffee table, and then check if the vase is spatially located on the coffee table. We will use the centroids of the objects to determine their spatial relationship.

First, let's identify the objects:

1. Vase with flowers:
   - Object ID 253: 'vase', description: 'a vase of yellow artificial flowers, likely made of silk or a similar fabric, designed to mimic the appearance of real flowers for decorative purposes', centroid: [3.82234383, 0.99759667, -0.66574706]

2. Coffee table:
   - Object ID 261: 'table', description: 'a round wooden coffee table with a smooth finish, featuring a decorative vase and tray on top, typically used in a living room setting', centroid: [3.71326733, 1.02602131, -1.17538255]

Now, let's analyze the spatial relationship between the vase (object 253) and the coffee table (object 261) using their centroids.

The centroid of the vase is at [3.82234383, 0.99759667, -0.66574706], and the centroid of the coffee table is at [3.71326733, 1.02602131, -1.17538255]. We will consider the x, y, and z coordinates separately to determine if the vase is on the coffee table.

For the vase to be on the coffee table, its x and y coordinates should be within the bounds of the coffee table, and its z coordinate should be just above the maximum z bound of the coffee table.

Let's calculate the approximate distance between the centroids in the x, y, and z dimensions:

- Δx (difference in x): |3.82234383 - 3.71326733| = 0.1090765
- Δy (difference in y): |0.99759667 - 1.02602131| = 0.02842464
- Δz (difference in z): |-0.66574706 - (-1.17538255)| = |-0.66574706 + 1.17538255| = 0.50963549

The Δx and Δy are relatively small, suggesting that the vase is close to the coffee table in the horizontal plane. The Δz indicates that the vase is above the coffee table, as expected for an object placed on top of another.

Given the description of the coffee table includes "featuring a decorative vase and tray on top," and the spatial analysis shows that the vase's centroid is close to the coffee table's centroid, especially in the x and y dimensions, and is above the coffee table in the z dimension, it is reasonable to conclude that the vase with flowers (object 253) is indeed on the coffee table (object 261).

Therefore, the answer is yes, there is a vase with flowers on the coffee table.

Question: What is the distance between sofa with blanket and door?

LLM: To calculate the distance between the sofa with a blanket and the door, we need to identify the objects in the database that correspond to these descriptions. We will use the centroids of the objects to calculate the Euclidean distance between them.

First, let's identify the sofa with a blanket. We have two objects with the name "sofa" in the database:

- Object with id 2: 'name': 'sofa'
- Object with id 299: 'name': 'sofa'

We also have objects with the name "blanket" or similar descriptions that could be associated with a sofa:

- Object with id 143: 'name': 'bedding', 'description': 'a plush throw blanket with a marbled or mottled pattern, commonly used for warmth and decoration on furniture such as couches or chairs'

Since the description of object 143 mentions that it is commonly used on furniture such as couches (sofas), we can assume that this is the blanket associated with the sofa. Now, we need to determine which sofa is closer to the blanket to consider them as a single object. We will calculate the distance between the centroid of the blanket (object 143) and the centroids of both sofas (object 2 and object 299).

Let's calculate the distance between the blanket (object 143) and sofa (object 2):

Centroid of blanket (object 143): [2.83747871, 2.37487857, -1.15898974]
Centroid of sofa (object 2): [3.67950376, -0.45651378, -1.04464772]

```
Distance = sqrt((x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2)
Distance = sqrt((3.67950376 - 2.83747871)^2 + (-0.45651378 - 2.37487857)^2 + (-1.04464772 - (-1.15898974))^2)
Distance = sqrt((0.84202505)^2 + (-2.83139235)^2 + (0.11434202)^2)
Distance = sqrt(0.70900642 + 8.016758 + 0.01307857)
Distance = sqrt(8.738843)
```

Now, let's calculate the distance between the blanket (object 143) and sofa (object 299):

Centroid of sofa (object 299): [3.74928062, 2.69762554, -1.02606409]

```
Distance = sqrt((3.74928062 - 2.83747871)^2 + (2.69762554 - 2.37487857)^2 + (-1.02606409 - (-1.15898974))^2)
Distance = sqrt((0.91180191)^2 + (0.32274697)^2 + (0.13292565)^2)
Distance = sqrt(0.83137947 + 0.104164 + 0.017668)
Distance = sqrt(0.95321147)
```

Since the distance between the blanket and sofa (object 299) is smaller, we will consider object 299 as the "sofa with blanket."

Next, we need to find the "door" in the database:

- Object with id 18: 'name': 'door'

Now, let's calculate the distance between the "sofa with blanket" (object 299) and the "door" (object 18):

Centroid of sofa with blanket (object 299): [3.74928062, 2.69762554, -1.02606409]
Centroid of door (object 18): [0.03173599, -1.11861909, -0.39294916]

```
Distance = sqrt((0.03173599 - 3.74928062)^2 + (-1.11861909 - 2.69762554)^2 + (-0.39294916 - (-1.02606409))^2)
Distance = sqrt((-3.71754463)^2 + (-3.81624463)^2 + (0.63311493)^2)
Distance = sqrt(13.820078 + 14.563923 + 0.400836)
Distance = sqrt(28.784837)
```

Therefore, the distance between the "sofa with blanket" (object 299) and the "door" (object 18) is approximately the square root of 28.784837, which is about 5.36 units.

The answer is approximately 5.36 units.

Figure B.16: **Spatial reasoning process of LLM.** Response of the LLM (GPT4 [32]), to a spatial reasoning question. 'Black box' highlights question & answer, and 'colored arrows' indicated instances as referred to in Fig. 4.8.

Figure B.17: **Reference scene from ScanNet [4] dataset.** Reference image indicating multiple instances in Replica [44] *"scene0000_00"* scene, for spatial reasoning examples.

```
Question: Which sink is closer to bed, kitchen sink or bathroom sink?

LLM: To determine which sink is closer to a bed, we need to find the centroids of the sinks and the bed,
then calculate the Euclidean distances between the bed and each sink. Let's locate the relevant objects in
the provided database.

First, we find the centroid of the bed. There are multiple objects named "bed," but we will use the
centroid of one of them as they are likely to be in similar positions if they are indeed the same object.
Let's use the first "bed" object we find:

Bed (object_60):
- Centroid: [4.51357574, 2.12990937, 0.57735379]

Next, we find the centroids of the kitchen sink and the bathroom sink:

Kitchen Sink (object_530):
- Centroid: [5.79942242, 8.03256244, 1.02961013]

Bathroom Sink (object_179):
- Centroid: [7.66126835, 3.4697668, 0.51416306]

Now, we calculate the Euclidean distance between the bed and each sink using the formula:

Distance = √((x2 - x1)² + (y2 - y1)² + (z2 - z1)²)

For the kitchen sink:
Distance_bed_kitchen = √((5.79942242 - 4.51357574)² + (8.03256244 - 2.12990937)² + (1.02961013 -
0.57735379)²)
Distance_bed_kitchen = √((1.28584668)² + (5.90265307)² + (0.45225634)²)
Distance_bed_kitchen = √(1.653752 + 34.841314 + 0.204534)
Distance_bed_kitchen = √(36.6996)
Distance_bed_kitchen ≈ 6.0575

For the bathroom sink:
Distance_bed_bathroom = √((7.66126835 - 4.51357574)² + (3.4697668 - 2.12990937)² + (0.51416306 -
0.57735379)²)
Distance_bed_bathroom = √((3.14769261)² + (1.33985743)² + (-0.06319073)²)
Distance_bed_bathroom = √(9.9076 + 1.7955 + 0.00399)
Distance_bed_bathroom = √(11.70709)
Distance_bed_bathroom ≈ 3.4216

Comparing the two distances:
- Distance_bed_kitchen ≈ 6.0575
- Distance_bed_bathroom ≈ 3.4216

The bathroom sink (object_179) is closer to the bed (object_60) than the kitchen sink (object_530) because
3.4216 < 6.0575.
```

Figure B.18: **Spatial reasoning process of LLM.** Response of the LLM (GPT4 [32]), to a spatial reasoning question. 'Black box' highlights question & answer, and 'colored arrows' indicated instances as referred to in Fig. B.17.

Question: Where is kitchen?

LLM: To determine the location of the kitchen, we need to look for objects that are typically found in a kitchen. Common kitchen-related items include appliances like refrigerators, toasters, coffee makers, and dishwashers, as well as kitchenware like pots, pans, and spatulas. We should also consider countertops and cabinets, which are essential components of a kitchen.

Let's identify objects from the provided list that are associated with a kitchen:

1. Refrigerator: 'object_211' with centroid at [6.21471516, 6.38892669, 0.96774697]
2. Toaster: 'object_501' with centroid at [6.36788507, 7.63667654, 1.08366181] and 'object_511' with centroid at [6.40961496, 7.18687732, 1.07924889]
3. Coffee Maker: 'object_568' with centroid at [4.10857615, 7.91081418, 1.33341652]
4. Dishwasher: 'object_235' with centroid at [6.03521413, 8.05214656, 1.13970931]
5. Kitchenware: 'object_504' with centroid at [6.09647436, 8.18516962, 1.2594524], 'object_643' (likely the same as 'object_504') with centroid at [6.09647436, 8.18516962, 1.2594524], and 'object_530' with centroid at [5.79942242, 8.03256244, 1.02961013]
6. Spatula: 'object_506' with centroid at [6.00750469, 8.676282, 1.20321975]
7. Countertop: 'object_244' with centroid at [6.05640545, 7.77063349, 0.96815944] and 'object_570' with centroid at [4.57643288, 6.94316232, 0.92714557]

By examining the centroids of these objects, we can infer that the kitchen is likely located in the region where these objects are clustered. The centroids have x-coordinates ranging from approximately 4 to 6 and y-coordinates ranging from approximately 6 to 8. This suggests that the kitchen is in the area where these coordinates overlap.

To narrow down the location further, we can calculate the average centroid of the identified kitchen-related objects. However, since we have multiple objects with the same name and similar descriptions (like 'object_504' and 'object_643'), we should consider them as a single object to avoid skewing the average.

Let's calculate the average centroid for the unique kitchen-related objects:

```
Average centroid x-coordinate = (6.21471516 + 6.36788507 + 6.40961496 + 4.10857615 + 6.03521413 + 6.09647436 + 6.00750469 + 6.05640545 + 4.57643288) / 9
Average centroid y-coordinate = (6.38892669 + 7.63667654 + 7.18687732 + 7.91081418 + 8.05214656 + 8.18516962 + 8.676282 + 7.77063349 + 6.94316232) / 9
Average centroid z-coordinate = (0.96774697 + 1.08366181 + 1.07924889 + 1.33341652 + 1.13970931 + 1.2594524 + 1.20321975 + 0.96815944 + 0.92714557) / 9
```

Calculating these averages:

```
Average centroid x-coordinate ≈ (6.21471516 + 6.36788507 + 6.40961496 + 4.10857615 + 6.03521413 + 6.09647436 + 6.00750469 + 6.05640545 + 4.57643288) / 9 ≈ 5.764824
Average centroid y-coordinate ≈ (6.38892669 + 7.63667654 + 7.18687732 + 7.91081418 + 8.05214656 + 8.18516962 + 8.676282 + 7.77063349 + 6.94316232) / 9 ≈ 7.638832
Average centroid z-coordinate ≈ (0.96774697 + 1.08366181 + 1.07924889 + 1.33341652 + 1.13970931 + 1.2594524 + 1.20321975 + 0.96815944 + 0.92714557) / 9 ≈ 1.106762
```

Therefore, the kitchen is likely located around the average centroid coordinates [5.764824, 7.638832, 1.106762]. This is a rough estimate based on the centroids of kitchen-related objects in the database.

Question: Is there a toilet paper near, toilet?

LLM: To determine if there is toilet paper near the toilet, we need to locate the objects labeled as "toilet paper" and "toilet" in the database and then calculate the distance between their centroids. If the distance is reasonably small, we can conclude that the toilet paper is near the toilet.

First, let's find the objects with the names "toilet paper" and "toilet":

- Toilet paper: 'object_437': {'id': 437, 'name': 'bathroom accessories', 'description': 'a cylindrical roll of toilet paper, typically made from paper pulp, which is used for personal hygiene in restrooms it appears to be mounted on a spindle or holder, which facilitates the unrolling and tearing off of individual sheets', 'centroid': array([7.69067717, 2.12205892, 0.68665118])}

- Toilet: 'object_181': {'id': 181, 'name': 'bathroom fixture', 'description': 'a white ceramic toilet with a closed lid, typically used for human waste disposal and sanitation in a bathroom setting', 'centroid': array([7.98375043, 2.66959259, 0.3178026])}

Now, let's calculate the Euclidean distance between the centroids of the toilet paper (object_437) and the toilet (object_181):

```
Centroid of toilet paper (object_437): (7.69067717, 2.12205892, 0.68665118)
Centroid of toilet (object_181): (7.98375043, 2.66959259, 0.3178026)
```

```
Euclidean distance (d) = √[(x2 - x1)² + (y2 - y1)² + (z2 - z1)²]
```

Plugging in the values:

```
d = √[(7.98375043 - 7.69067717)² + (2.66959259 - 2.12205892)² + (0.3178026 - 0.68665118)²]
d = √[(0.29307326)² + (0.54753367)² + (-0.36884858)²]
d = √[0.085892 + 0.299792 + 0.136098]
d = √[0.521782]

d ≈ √0.521782
d ≈ 0.7222
```

The Euclidean distance between the toilet paper and the toilet is approximately 0.7222 units. Given that this is a relatively small distance, we can conclude that the toilet paper (object_437:bathroom accessories) is near the toilet (object_181:bathroom fixture).

Figure B.19: **Spatial reasoning process of LLM.** Response of the LLM (GPT4 [32]), to a spatial reasoning question. 'Black box' highlights question & answer, and 'colored arrows' indicated instances as referred to in Fig. B.17.

Appendix C

Evaluation Details

C.1 Details on Ablation Studies

Scenes	Crop Ratio 0.1					Crop Ratio 0.2					Crop Ratio 0.3				
	mAcc	F-mIoU	AP	AP50	AP25	mAcc	F-mIoU	AP	AP50	AP25	mAcc	F-mIoU	AP	AP50	AP25
room0	50.4	74.7	9.2	20.8	35.8	49.8	76.6	9.2	20.9	32.7	50.7	77.8	9.3	20.9	29.4
room1	47.9	42.7	18.5	29.3	35.3	49.1	43.6	18.3	29.6	35.3	47.9	42.7	18.5	29.3	35.3
room2	43.3	54.0	11.6	25.3	29.6	42.6	51.5	11.2	21.7	29.8	43.3	54.0	11.6	25.3	29.6
office0	39.2	30.0	5.4	20.9	25.7	39.3	30.6	5.9	21.1	26.1	39.2	30.0	5.4	20.9	25.7
office1	38.0	22.7	8.3	16.0	28.5	37.1	23.3	9.4	21.1	33.1	38.0	22.7	8.3	16.0	28.5
office2	38.4	38.6	9.0	21.8	26.3	38.5	38.9	9.0	21.8	26.3	38.4	38.6	9.0	21.8	26.3
office3	32.6	44.6	5.1	14.0	20.4	32.3	45.9	4.6	13.2	17.6	32.6	44.6	5.1	14.0	20.4
office4	29.2	47.7	3.7	7.4	23.3	37.7	47.6	3.7	7.5	20.7	29.2	47.7	3.7	7.4	23.3
avg room	47.2	57.1	**13.1**	25.1	**33.6**	47.2	57.2	12.9	24.1	32.6	**47.3**	**58.2**	**13.1**	**25.2**	31.4
avg office	35.5	36.7	6.3	16.0	**24.8**	**37.0**	**37.2**	**6.5**	**16.9**	**24.8**	35.5	36.7	6.3	16.0	**24.8**
avg scenes	39.9	44.4	**8.9**	19.4	28.1	**40.8**	44.7	**8.9**	**19.6**	**27.7**	39.9	**44.8**	**8.9**	19.4	27.3

Table C.1: **Details on ablation study of crop scaling ratio (S_r).** Ratio S_r by which the crop sides are scaled for multiscale feature fusion, assessed per scene on the Replica [44] dataset.

Scenes	Crop Levels 1					Crop Levels 3					Crop Levels 5				
	mAcc	F-mIoU	AP	AP50	AP25	mAcc	F-mIoU	AP	AP50	AP25	mAcc	F-mIoU	AP	AP50	AP25
room0	51.0	74.1	9.5	21.1	36.4	49.8	76.6	9.2	20.9	32.7	49.4	79.2	9.3	20.9	35.4
room1	47.2	42.1	18.5	29.3	33.5	49.1	43.6	18.3	29.6	35.3	47.2	42.1	18.5	29.3	33.5
room2	42.3	51.4	11.6	25.6	30.1	42.6	51.5	11.2	21.7	29.8	43.1	54.1	11.6	25.3	29.9
office0	39.4	30.9	6.4	21.1	26.1	39.3	30.6	5.9	21.1	26.1	38.6	29.6	5.9	21.1	26.1
office1	7.3	21.3	8.9	16.1	27.9	37.1	23.3	9.4	21.1	33.1	37.5	22.3	6.4	11.5	26.1
office2	39.0	37.6	9.0	21.8	26.2	38.5	38.9	9.0	21.8	26.3	38.6	37.6	9.1	22.6	27.0
office3	32.4	44.5	5.2	14.5	21.1	32.3	45.9	4.6	13.2	17.6	31.9	46.1	4.6	17.4	17.2
office4	28.5	46.8	3.7	7.5	20.6	37.7	47.6	3.7	7.5	20.7	28.5	43.8	5.2	6.7	19.6
avg room	46.8	55.8	**13.2**	**25.3**	**33.3**	**47.2**	57.2	12.9	24.1	32.6	46.6	**58.5**	13.1	25.2	32.9
avg office	29.3	36.2	**6.7**	16.2	24.4	**37.0**	**37.2**	6.5	**16.9**	**24.8**	35.0	35.9	6.3	15.9	23.2
avg scenes	35.9	43.6	9.1	**19.6**	**27.7**	**40.8**	**44.7**	8.9	**19.6**	**27.7**	39.4	44.3	8.8	19.4	26.9

Table C.2: **Details on ablation study of crop level (k).** The total number of crops k, considered for multiscale feature fusion, assessed per scene on the Replica [44] dataset.

Scenes	Top Images 1					Top Images 5					Top Images 10				
	mAcc	F-mIoU	AP	AP51	AP26	mAcc	F-mIoU	AP	AP50	AP25	mAcc	F-mIoU	AP	AP50	AP25
room0	49.7	73.1	9.9	24.0	33.7	49.8	76.6	9.2	20.9	32.7	49.3	75.9	8.6	17.8	29.6
room1	50.1	43.8	18.5	29.2	35.3	49.1	43.6	18.3	29.6	35.3	48.4	42.0	18.3	29.6	32.4
room2	43.0	52.5	11.1	21.2	29.1	42.6	51.5	11.2	21.7	29.8	43.2	53.0	11.5	25.0	30.4
office0	39.3	29.8	5.4	20.9	25.9	39.3	30.6	5.9	21.1	26.1	38.7	30.8	5.9	20.8	25.8
office1	36.0	22.1	7.3	16.0	28.5	37.1	23.3	9.4	21.1	33.1	36.1	22.8	7.9	16.4	32.5
office2	39.5	40.3	9.0	21.8	26.9	38.5	38.9	9.0	21.8	26.3	38.5	38.5	9.0	21.8	26.2
office3	31.7	43.5	5.1	13.7	17.8	32.3	45.9	4.6	13.2	17.6	32.1	45.9	4.7	13.8	19.5
office4	27.6	42.3	3.0	7.4	20.3	37.7	47.6	3.7	7.5	20.7	28.3	45.2	3.7	7.4	23.6
avg room	**47.6**	56.5	**13.1**	**24.8**	**32.7**	47.2	**57.2**	12.9	24.1	32.6	46.9	57.0	12.8	24.1	30.8
avg office	34.8	35.6	6.0	15.9	23.9	**37.0**	**37.2**	**6.5**	**16.9**	24.8	34.7	36.7	6.3	16.1	**25.5**
avg scenes	39.6	43.4	8.7	19.3	27.2	**40.8**	**44.7**	**8.9**	**19.6**	**27.7**	39.3	44.3	8.7	19.1	27.5

Table C.3: **Details on ablation study of top images (**m**).** The total images m with top s_{pred} considered for multiview feature fusion, assessed per scene on Replica [44] dataset.

Scenes	ViT-L-14					ViT-H-14					ViT-G-14				
	mAcc	F-mIoU	AP	AP50	AP25	mAcc	F-mIoU	AP	AP50	AP25	mAcc	F-mIoU	AP	AP50	AP25
room0	48.7	90.5	15.8	29.4	38.6	55.2	73.1	14.2	28.4	33.8	54.5	93.1	15.3	28.4	28.4
room1	45.3	43.7	16.9	21.4	34.5	51.2	44.8	17.2	25.3	35.1	51.4	46.8	18.1	26.2	37.4
room2	42.5	53.4	13.8	27.3	31.6	48.2	48.2	12.3	22.0	28.2	41.8	54.6	13.8	27.1	31.3
office0	44.9	32.9	6.9	21.1	34.2	46.7	28.6	6.3	21.1	31.3	46.5	34.2	9.5	26.2	35.5
office1	32.7	25.5	7.0	14.8	30.5	38.8	24.7	10.0	28.5	38.0	40.1	20.3	7.9	15.7	29.5
office2	35.6	42.5	7.2	21.7	26.8	41.5	34.8	7.2	21.2	30.4	43.0	39.6	7.3	21.8	27.5
office3	32.9	47.7	5.9	21.5	21.6	33.3	35.7	9.8	19.8	35.7	37.3	42.3	5.2	14.5	23.1
office4	28.8	49.7	0.7	6.7	24.5	26.0	36.9	2.1	6.6	19.9	26.4	40.6	5.9	8.5	14.2
avg room	45.5	62.5	15.5	26.1	**34.9**	51.5	55.4	14.6	25.2	32.4	**49.2**	**64.8**	**15.8**	**27.3**	32.4
avg office	35.0	**39.7**	5.5	17.1	27.5	37.3	32.1	7.1	**19.4**	**31.1**	**38.7**	35.4	**7.2**	17.3	25.9
avg scenes	38.9	48.2	9.3	20.5	30.3	42.6	40.9	9.9	**21.6**	**31.6**	**42.6**	**46.4**	**10.4**	21.1	28.4

Table C.4: **Details on ablation study of CLIP model variants.** Influence of different CLIP Models on instance recall performance, assessed per scene on Replica [44] dataset.

C.2 Details on Quantitative Evaluation

We conducted a self-evaluation of the baseline method, examining both ConceptGraph [9] and OpenMask3D [46]. Our findings revealed that the overall results are very close to the one those in the original papers. Below, we provide a detailed per scene breakdown of quantitative metrics.

Scenes	ConceptGraph [9]			OpenSU3D (Ours)		
	mAcc	mIoU	F-mIoU	mAcc	mIoU	F-mIoU
room0	37.9	26.5	46.3	55.2	26.9	73.1
room1	37.0	24.3	38.7	51.2	30.6	44.8
room2	28.6	15.1	40.0	48.2	26.9	48.2
office0	30.1	16.0	28.4	46.7	23.5	28.6
office1	29.7	15.5	12.8	38.8	28.8	24.7
office2	37.5	26.5	47.9	41.5	27.5	34.8
office3	31.3	18.8	38.5	33.3	21.9	35.7
office4	56.8	43.5	42.1	26.0	18.5	36.9
avg room	34.5	21.9	41.7	**51.5**	**28.2**	**55.4**
avg office	37.1	**24.1**	**33.9**	**37.3**	**24.1**	32.1
avg scenes	36.1	23.3	36.8	**42.6**	**25.6**	**40.9**

Table C.5: **Details on comparison of open-vocabulary segmentation results on Replica [44].** Per scene quantitative evaluation and comparison of open vocabulary instance segmentation performance with ConceptGraph [9], using experimental settings as proposed in ConceptGraph [9].

Scenes	OpenMask3D [46]			OpenSU3D (Ours)		
	AP	AP50	AP25	AP	AP50	AP25
room0	15.7	20.2	21.3	9.2	20.9	32.8
room1	18.7	24.6	34.4	18.3	29.6	35.3
room2	12.9	18.1	20.9	11.2	21.7	29.8
office0	12.7	16.5	19.2	5.9	21.1	26.1
office1	18.2	23.5	26.5	9.4	21.1	33.1
office2	14.9	26.0	30.5	9.0	21.8	26.3
office3	8.7	13.2	13.2	4.6	13.2	17.6
office4	13.0	13.2	18.4	3.7	7.5	20.7
avg room	**15.8**	21.0	25.5	12.9	**24.1**	**32.6**
avg office	**13.5**	**18.5**	21.6	6.5	16.9	**24.8**
avg scenes	**14.4**	19.4	23.1	8.9	**19.4**	**27.7**

Table C.6: **Details on comparison of open-vocabulary segmentation resultson Replica [44].** Per scene quantitative evaluation and comparison of open vocabulary instance segmentation performance with OpenMask3D [46], using experimental settings as proposed in OpenMask3D [46].

Scenes	OpenSU3D (Ours)		
	AP	AP50	AP25
Scene0000_00	2.8	10.7	23.9
Scene0034_00	11.2	31.7	37.2
Scene0525_01	2.1	6.2	18.6
Scene0164_03	16.4	36.5	35.9
Scene0549_00	5.7	18.4	24.5
avg scenes	**7.6**	**20.7**	**28.0**

Table C.7: **Details on comparison of open-vocabulary segmentation results on ScanNet200 [41].** Per scene quantitative evaluation of open vocabulary instance segmentation performance, using experimental settings as proposed in OpenMask3D [46].

Scenes	Scheme 1			Scheme 2			Scheme 3			Scheme 4		
	AP	AP50	AP25	AP	AP50	AP25	AP	AP50	AP25	AP	AP50	AP25
room0	14.2	28.4	33.8	12.3	20.2	30.7	12.6	22.7	33.2	13.4	20.3	33.5
room1	17.2	25.3	35.1	14.9	21.9	39.9	16.3	22.6	36.3	17.5	22.0	38.6
room2	12.3	22.0	28.2	12.3	22.6	28.6	11.1	20.6	26.4	12.7	26.3	27.8
office0	6.3	21.1	31.3	7.4	21.2	34.7	6.8	21.1	31.4	7.4	21.2	34.4
office1	10.0	28.5	38.0	7.7	20.1	37.2	5.6	10.1	34.8	9.7	25.0	37.7
office2	7.2	21.2	30.4	6.8	17.0	20.9	6.8	16.9	22.4	9.3	23.1	27.5
office3	9.8	19.8	35.7	4.7	13.8	23.4	4.3	13.6	21.9	5.6	18.3	22.3
office4	2.1	6.6	19.9	2.8	6.7	20.6	2.9	6.7	20.7	2.9	7.2	21.1
avg room	**14.6**	25.2	32.4	13.2	21.6	33.1	13.3	22.0	32.0	14.5	**22.8**	**33.3**
avg office	**7.1**	**19.4**	**31.1**	5.9	15.7	27.3	5.3	13.7	26.2	7.0	19.0	28.6
avg scenes	**9.9**	**21.6**	**31.6**	8.6	17.9	29.5	8.3	16.8	28.4	9.8	20.4	30.4

Table C.8: **Details on quantitative evaluation of feature fusion schemes on open vocabulary instance retrieval performance on Replica [44].** Per scene quantitative evaluation of open vocabulary instance segmentation performance for different feature fusion schemes.

C.3 Details on Qualitative Evaluation

Scenes	TotalQs	Scheme 1				Scheme 2				Scheme 3				Scheme 4			
		Inst.	Aff.	Prop.	Rel.	Inst.	Aff.	Prop.	Rel.	Inst.	Aff.	Prop.	Rel.	Inst.	Aff.	Prop.	Rel.
room0	66	1.0	0.3	0.8	0.5	1.0	0.3	1.0	0.0	1.0	1.0	1.0	1.0	0.8	1.0	1.0	0.7
room1	73	0.9	0.7	0.7	0.5	0.8	0.7	0.8	0.5	1.0	0.7	0.8	0.5	1.0	0.7	0.8	0.5
room2	88	0.7	0.5	1.0	0.5	0.9	0.8	1.0	0.8	0.8	0.8	1.0	0.5	0.9	1.0	1.0	1.0
office0	84	0.9	0.7	0.6	0.0	0.9	0.9	0.8	0.0	1.0	1.0	0.8	0.0	0.9	0.9	0.8	0.0
office1	80	0.7	0.8	0.6	0.5	0.7	0.5	0.6	0.6	0.9	1.0	0.8	0.8	0.7	1.0	1.0	0.8
office2	100	0.7	1.0	0.8	0.3	0.6	0.8	0.8	1.0	0.8	1.0	1.0	0.3	0.7	0.8	0.8	1.0
office3	88	0.9	0.8	1.0	0.3	0.9	0.6	1.0	1.0	1.0	1.0	1.0	0.6	0.9	1.0	1.0	1.0
office4	56	0.7	1.0	0.5	0.0	0.7	1.0	1.0	0.0	0.9	1.0	0.5	1.0	0.9	1.0	1.0	0.0
avg room	-	0.8	0.5	0.8	0.5	**0.9**	0.6	**0.9**	0.4	**0.9**	0.8	**0.9**	**0.7**	**0.9**	**0.9**	**0.9**	**0.7**
avg office	-	0.8	0.9	0.7	0.2	0.8	0.7	0.8	0.5	**0.9**	**1.0**	0.8	0.5	0.8	0.9	**0.9**	**0.6**
avg scenes	-	0.8	0.7	0.7	0.3	0.8	0.7	**0.9**	0.5	**0.9**	**0.9**	**0.9**	**0.6**	0.8	**0.9**	**0.9**	**0.6**

Table C.9: **Details on qualitative evaluation of feature fusion schemes on open vocabulary instance retrieval performance on Replica [44].** Per scene assessment of the accuracy of feature fusion schemes for object retrieval for "Inst." (instance), "Aff." (affordance), "Prop." (Property). and "Rel." (relative) text queries, as assessed per scene by a human evaluator.

Scenes	TotalQs	Scheme 1				Scheme 2				Scheme 3				Scheme 4			
		Inst.	Aff.	Prop.	Rel.	Inst.	Aff.	Prop.	Rel.	Inst.	Aff.	Prop.	Rel.	Inst.	Aff.	Prop.	Rel.
Scene0000_00	128	0.9	0.8	0.8	0.5	1.0	1.0	0.8	0.5	1.0	0.8	0.8	0.9	1.0	0.9	0.8	0.8
Scene0034_00	72	0.7	0.7	0.7	0.3	0.9	0.3	0.7	1.0	0.8	0.7	0.7	0.5	0.9	0.7	0.7	1.0
Scene0525_01	74	0.8	1.0	0.3	0.7	0.7	1.0	0.7	1.0	0.9	1.0	0.3	0.7	1.0	1.0	0.7	0.7
Scene0164_03	60	0.7	0.8	0.7	0.3	0.6	0.5	1.0	0.3	0.8	0.8	1.0	0.3	0.9	0.5	0.7	0.8
Scene0549_00	56	1.0	1.0	1.0	0.5	1.0	0.7	1.0	0.5	0.8	0.7	1.0	0.5	0.9	0.7	1.0	0.5
avg scenes	-	0.8	**0.8**	0.7	0.4	**0.9**	0.7	**0.8**	0.6	**0.9**	**0.8**	0.7	0.6	**0.9**	0.7	0.7	**0.7**

Table C.10: **Details on qualitative evaluation of feature fusion schemes on open vocabulary instance retrieval performance on ScanNet [4].** Accuracy of feature fusion schemes for instance retrieval with "Inst." (instance), "Aff." (affordance), "Prop." (Property). and "Rel." (relative) text queries, as assessed per scene by a human evaluator.

Scenes	MAX Labels (n)					LLM Labels (n')				
	Total Lbl.	Corr. Lbl.	Flt. Seg.	Label Acc.	Merge Acc.	Total Lbl.	Corr. Lbl.	Flt. Seg.	Label Acc.	Merge Acc.
room0	39	32	3	0.82	0.92	38	36	3	0.95	0.92
room1	31	27	3	0.87	0.90	31	29	3	0.94	0.90
room2	25	22	3	0.88	0.88	24	22	2	0.92	0.92
office0	30	25	5	0.83	0.83	29	24	5	0.83	0.83
office1	21	16	2	0.76	0.90	21	15	2	0.71	0.90
office2	27	21	3	0.78	0.89	26	23	3	0.88	0.88
office3	33	27	6	0.82	0.82	33	28	6	0.85	0.82
office4	31	27	6	0.87	0.81	30	27	5	0.90	0.83
avg room	-	-	-	0.86	0.90	-	-	-	**0.93**	**0.91**
avg office	-	-	-	0.81	**0.85**	-	-	-	**0.83**	**0.85**
avg scenes	-	-	-	0.83	0.87	-	-	-	**0.87**	**0.88**

Table C.11: **Details on qualitative evaluation of segmentation and annotation accuracy on Replica [44].** For MAX Label (n) and LLM Label (n'), the annotation and merge accuracy of segmentation masks, as assessed per scene by a human evaluator.

Scenes	MAX Labels (n)					LLM Labels (n')				
	Total Lbl.	Corr. Lbl.	Flt. Seg.	Lbl. Acc.	Merge Acc.	Total Lbl.	Corr. Lbl.	Flt. Seg.	Lbl. Acc.	Merge Acc.
Scene0000_00	151	109	31	0.72	0.79	148	111	27	0.75	0.82
Scene0525_01	62	50	10	0.81	0.84	62	58	9	0.94	0.85
Scene0549_00	31	22	5	0.71	0.84	30	23	3	0.77	0.90
Scene0164_03	50	40	6	0.80	0.88	50	45	4	0.90	0.92
Scene0034_00	66	46	8	0.70	0.88	64	54	9	0.84	0.86
avg scenes	-	-	-	0.75	0.85	-	-	-	**0.84**	**0.87**

Table C.12: **Details on qualitative evaluation of segmentation and annotation accuracy on ScanNet [4].** For MAX Label (n) and LLM Label (n'), the annotation and merge accuracy of segmentation masks, as assessed per scene by a human evaluator.

Bibliography

[1] J.-B. Alayrac et al. “Flamingo: a Visual Language Model for Few-Shot Learning”. In: *Neural Information Processing Systems (NeurIPS)*. 2022.

[2] Tom B. Brown et al. “Language Models are Few-Shot Learners”. In: *ArXiv* abs/2005.14165 (2020). URL: https://api.semanticscholar.org/CorpusID:218971783.

[3] Mark Chen et al. “Evaluating Large Language Models Trained on Code”. In: *ArXiv* abs/2107.03374 (2021).

[4] A. Dai et al. “ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017. DOI: 10.1109/CVPR.2017.261.

[5] R. Ding et al. “PLA: Language-driven open vocabulary 3d scene understanding”. In: *Proceedings of Computer Vision and Pattern Recognition*. 2023.

[6] F. Engelmann et al. *Open-set 3d scene segmentation with rendered novel views*. 2023.

[7] *FNNTW: Fastest Nearest Neighbor (in the) West*. URL: https://pypi.org/project/pyfnntw/ (visited on 03/28/2021).

[8] M. Grinvald et al. “Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery”. In: *IEEE Robotics and Automation Letters* (2019).

[9] Qiao Gu et al. *ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning*. 2023. DOI: 10.48550/ARXIV.2309.16650. URL: https://arxiv.org/abs/2309.16650.

[10] H. Ha and S. Song. “Semantic Abstraction: Open-World 3D Scene Understanding from 2D Vision-Language Models”. In: *Conference on Robot Learning (CORL)*. 2022.

[11] Y. Hong et al. “3d concept learning and reasoning from multi-view images”. In: *Proceedings of Computer Vision and Pattern Recognition*. 2023.

[12] Y. Hong et al. “3d-llm: Injecting the 3d world into large language models”. In: *Neural Information Processing Systems*. 2023.

[13] Rui HUANG et al. *Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels*. 2023. DOI: 10.48550/ARXIV.2312.17232. URL: https://arxiv.org/abs/2312.17232.

[14] Zhening HUANG et al. "OpenIns3D: Snap and Lookup for 3D Open-vocabulary Instance Segmentation". In: *arXiv preprint* (2023).

[15] N. HUGHES, Y. CHANG, and L. CARLONE. "Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization". In: (2022).

[16] Gabriel ILHARCO et al. *OpenCLIP*. Version 0.1. If you use this software, please cite it as below. July 2021. DOI: 10.5281/zenodo.5143773. URL: https://doi.org/10.5281/zenodo.5143773.

[17] K. M. JATAVALLABHULA et al. "ConceptFusion: Open-set multimodal 3d mapping". In: *Robotics: Science and Systems*. 2023.

[18] Krishna Murthy JATAVALLABHULA, Ganesh IYER, and Liam PAULL. "∇SLAM: Dense SLAM meets Automatic Differentiation". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2020. DOI: 10.1109/icra40945.2020.9197519. URL: https://doi.org/10.1109/icra40945.2020.9197519.

[19] J. KERR et al. "LERF: Language embedded radiance fields". In: *International Conference on Computer Vision (ICCV)*. 2023.

[20] A. KIRILLOV et al. "Segment anything". In: *Proceedings of International Conference on Computer Vision*. 2023.

[21] B. LI et al. "Language-driven Semantic Segmentation". In: *International Conference on Learning Representations (ICLR)*. 2022.

[22] F. LI et al. *Semantic-SAM: Segment and Recognize Anything at Any Granularity*. http://arxiv.org/abs/2307.04767. 2023.

[23] J. LI et al. "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. PMLR, 2023.

[24] J. LI et al. "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. PMLR, 2022.

[25] F. LIANG et al. "Open-Vocabulary Semantic Segmentation with Mask-adapted CLIP". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[26] H. LIU et al. *Improved Baselines with Visual Instruction Tuning*. http://arxiv.org/abs/2310.03744. Oct. 2023.

[27] Haotian LIU et al. "Visual Instruction Tuning". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=w0H2xGHlkw.

[28] S. LIU et al. *Grounding dino: Marrying dino with grounded pre-training for open-set object detection*. arXiv preprint arXiv:2303.05499. 2023.

[29] T. LUDDECKE and A. ECKER. "Image Segmentation Using Text and Image Prompts". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022. DOI: 10.1109/CVPR52688.2022.00695.

[30] Z. LUO et al. *A Frustratingly Simple Approach for End-to-End Image Captioning*. https://arxiv.org/abs/2201.12723. 2022. DOI: 10.48550/ARXIV.2201.12723.

[31] J. MCCORMAC et al. "SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks". In: *ICRA*. 2017.

[32] OPENAI. *Gpt-4 technical report*. arXiv preprint arXiv:2303.08774. 2023.

[33] S. PENG et al. "Openscene: 3d scene understanding with open vocabularies". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 815–824.

[34] S. PENG et al. "Openscene: 3d scene understanding with open vocabularies". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 815–824.

[35] Minghan QIN et al. "LangSplat: 3D Language Gaussian Splatting". In: *arXiv preprint arXiv:2312.16084* (2023).

[36] A. RADFORD et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. PMLR. 2021, pp. 8748–8763.

[37] Krishan RANA et al. "SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Task Planning". In: *7th Annual Conference on Robot Learning*. 2023. URL: https://openreview.net/forum?id=wMpOMO0Ss7a.

[38] Machel REID et al. *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*. 2024. DOI: 10.48550/ARXIV.2403.05530. URL: https://arxiv.org/abs/2403.05530.

[39] T. Ren et al. *Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks*. http://arxiv.org/abs/2401.14159. Jan. 2024.

[40] A. Rosinol et al. "Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping". In: *ICRA*. 2020.

[41] D. Rozenberszki, O. Litany, and A. Dai. "Language-Grounded Indoor 3D Semantic Segmentation in the Wild". In: *European Conference on Computer Vision (ECCV)*. 2022.

[42] J. Schult et al. "Mask3D: Mask Transformer for 3D Semantic Instance Segmentation". In: *International Conference on Robotics and Automation (ICRA)*. 2023.

[43] N. M. M. Shafiullah et al. "Clip-fields: Weakly supervised semantic fields for robotic memory". In: *Robotics: Science and Systems*. Ed. by K. E. Bekris et al. 2023.

[44] J. Straub et al. *The Replica Dataset: A Digital Replica of Indoor Spaces*. https://arxiv.org/abs/1906.05797. 2019. doi: 10.48550/ARXIV.1906.05797.

[45] Dídac Surís, Sachit Menon, and Carl Vondrick. "ViperGPT: Visual Inference via Python Execution for Reasoning". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)* (2023).

[46] Ayca Takmaz et al. "OpenMask3D: Open-Vocabulary 3D Instance Segmentation". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 68367–68390. url: https://proceedings.neurips.cc/paper_files/paper/2023/file/d77b5482e38339a8068791d939126be2-Paper-Conference.pdf.

[47] Hugo Touvron et al. "Llama 2: Open Foundation and Fine-Tuned Chat Models". In: *ArXiv* abs/2307.09288 (2023). url: https://api.semanticscholar.org/CorpusID:259950998.

[48] Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: *ArXiv* abs/2302.13971 (2023). url: https://api.semanticscholar.org/CorpusID:257219404.

[49] T. Wang et al. *Caption Anything: Interactive Image Description with Diverse Multimodal Controls*. http://arxiv.org/abs/2305.02677. 2023.

[50] Ying Wang, Yanlai Yang, and Mengye Ren. *LifelongMemory: Leveraging LLMs for Answering Queries in Long-form Egocentric Videos*. 2024. arXiv: 2312.05269 [cs.CV].

[51] Shun-Cheng Wu et al. “Incremental 3D Semantic Scene Graph Prediction from RGB Sequences”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[52] Shun-Cheng Wu et al. “SceneGraphFusion: Incremental 3D Scene Graph Prediction From RGB-D Sequences”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7515–7525.

[53] Jianing Yang et al. *LLM-Grounder: Open-Vocabulary 3D Visual Grounding with Large Language Model as an Agent*. 2023. DOI: 10.48550/ARXIV.2309.12311. URL: https://arxiv.org/abs/2309.12311.

[54] Jianwei Yang et al. *Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V*. 2023. DOI: 10.48550/ARXIV.2310.11441. URL: https://arxiv.org/abs/2310.11441.

[55] Y. Yang et al. *SAM3D: Segment Anything in 3D Scenes*. https://arxiv.org/abs/2306.03908v1. 2023.

[56] Chaohui Yu et al. *Points-to-3D: Bridging the Gap between Sparse Points and Shape-Controllable Text-to-3D Generation*. 2023. DOI: 10.48550/ARXIV.2307.13908. URL: https://arxiv.org/abs/2307.13908.

[57] L. Yuan et al. *Florence: A new foundation model for computer vision*. arXiv preprint arXiv:2111.11432. 2021.

[58] H. Zhang et al. *LLaVA-Grounding: Grounded Visual Chat with Large Multimodal Models*. https://arxiv.org/abs/2312.02949. 2023.

[59] Y. Zhang et al. *Recognize Anything: A Strong Image Tagging Model*. http://arxiv.org/abs/2306.03514. Feb. 2024.

[60] X. Zou et al. *Segment Everything Everywhere All at Once*. http://arxiv.org/abs/2304.06718. 2023.

Declaration

I hereby affirm that I have independently written the thesis submitted by me and have not used any sources or aids other than those indicated.

Location, Date, Signature