OpenMap: Instruction Grounding via Open-Vocabulary Visual-Language Mapping

Danyang Li*
lidanyang1919@gmail.com
School of Software, Tsinghua University
Beijing, China

Zenghui Yang*
zenghuiyang36@gmail.com
School of computer science and engineering, Central South University
Hunan, China

Guangpeng Qi
qigp@inspur.com
Inspur Yunzhou Industrial Internet Co., Ltd
Shandong, China

Songtao Pang
pangst@inspur.com
Inspur Yunzhou Industrial Internet Co., Ltd
Shandong, China

Guangyong Shang
shangguangyong@inspur.com
Inspur Yunzhou Industrial Internet Co., Ltd
Shandong, China

Qiang Ma†
tsinghuamq@gmail.com
Tsinghua University
Beijing, China

Zheng Yang
hmilyyz@gmail.com
School of Software, Tsinghua University
Beijing, China

Abstract

Grounding natural language instructions to visual observations is fundamental for embodied agents operating in open-world environments. Recent advances in visual-language mapping have enabled generalizable semantic representations by leveraging vision-language models (VLMs). However, these methods often fall short in aligning free-form language commands with specific scene instances, due to limitations in both instance-level semantic consistency and instruction interpretation. We present **OpenMap**, a zero-shot open-vocabulary visual-language map designed for accurate instruction grounding in navigation tasks. To address semantic inconsistencies across views, we introduce a *Structural-Semantic Consensus* constraint that jointly considers global geometric structure and vision-language similarity to guide robust 3D instance-level aggregation. To improve instruction interpretation, we propose an LLM-assisted *Instruction-to-Instance Grounding* module that enables fine-grained instance selection by incorporating spatial context and expressive target descriptions. We evaluate OpenMap on ScanNet200 and Matterport3D, covering both semantic mapping and instruction-to-target retrieval tasks. Experimental results show that OpenMap outperforms state-of-the-art baselines in zero-shot settings, demonstrating the effectiveness of our method in bridging free-form language and 3D perception for embodied navigation.

*Both authors contributed equally to this research.
†Qiang Ma is the corresponding author.

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754887

CCS Concepts

• **Information systems** → **Multimedia information systems**; • **Computing methodologies** → **Cognitive robotics**.

Keywords

Instruction Grounding, Open-Vocabulary Mapping, Vision-Language Models, 3D Semantic Mapping, Embodied Navigation

ACM Reference Format:
Danyang Li, Zenghui Yang, Guangpeng Qi, Songtao Pang, Guangyong Shang, Qiang Ma, and Zheng Yang. 2025. OpenMap: Instruction Grounding via Open-Vocabulary Visual-Language Mapping. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746027.3754887

1 Introduction

As the field of embodied intelligence continues to evolve, enabling agents to navigate using natural language instructions has emerged as a core challenge [4]. In Vision-and-Language Navigation (VLN), agents are expected to interpret language commands and perform goal-directed planning based on visual observations in complex 3D environments [26, 40]. To support this process, semantic maps are essential, as they enhance perceptual understanding and enable precise, instruction-driven navigation [34, 37]. Recent advances have further incorporated semantic features from VLMs [13, 29] into 3D scene representations, giving rise to open-vocabulary visual-language maps [7, 9, 38] that generalize well across a wide range of navigation tasks. However, effectively grounding natural language instructions to specific 3D instances within these maps—*i.e.*, instruction grounding—remains an open challenge.

Existing open-vocabulary visual-language maps typically follow a two-stage pipeline: (1) **Semantic mapping**: As agents navigate the environment, they collect visual observations and use VLMs

(a) Instance-level semantic mapping. **(b) Grounding generic navigation instructions to target instances.**

Figure 1: OpenMap constructs an open-vocabulary visual-language map. (a) OpenMap performs fine-grained, instance-level semantic mapping on navigation scenes from Matterport3D [43]. (b) Three types of navigation instructions are shown from left to right: object-goal, demand-driven, and language-guided. OpenMap accurately grounds generic instructions to the intended targets, where darker regions in the heatmaps indicate stronger alignment between the instruction and the predicted instance.

to extract open-vocabulary features at the pixel level. These features are back-projected into 3D and aggregated across views to construct the semantic map. (2) **Instruction grounding**: Given a free-form instruction, large language models (LLMs) generate descriptive target expressions, which are then grounded to the map by matching them against visual-language features.

While promising, these approaches still face significant limitations. In many cases, the alignment between natural language instructions and map instances underperforms even basic text-to-image matching capabilities of VLMs. Two core challenges remain:
• **In semantic mapping**, existing methods often rely on spatial or structural constraints to merge observations from different viewpoints. Some cluster point clouds by proximity or predefined grids [2, 9], which may lead to over- or under-segmentation. More advanced techniques leverage structural overlaps [38, 44], but incomplete point clouds and object occlusion can still cause erroneous merges between semantically distinct instances.
• **In instruction grounding**, despite operating on open-vocabulary features, most methods remain tied to predefined object lexicons. Some rely on static category labels [7, 38], while others restrict LLM outputs to a fixed set of terms [9, 10]. These constraints hinder the expressive capacity of LLMs and limit their ability to capture fine-grained, contextualized object references. For example, given the instruction *"Get the chair ready—I want to eat"*, identifying the chair near the dining table, rather than a generic chair, is non-trivial.

In summary, existing visual-language maps—across both semantic mapping and instruction grounding—largely follow closed-vocabulary paradigms, limiting the potential of VLMs and LLMs in open-world navigation. On one hand, they fail to fully exploit structural and semantic cues for robust instance association; on the other, they underutilize the generative flexibility of LLMs in grounding diverse instructions.

Our Work. We introduce **OpenMap**, a zero-shot **Open** vocabulary visual-language **Map** designed for accurate instruction grounding in embodied navigation. OpenMap addresses the above challenges by aligning natural language instructions with 3D instances through a unified visual-language representation. Specifically, we propose a structural-semantic consensus constraint that leverages both geometric and semantic consistency to drive robust instance merging during mapping (Fig. 1a). Furthermore, we introduce an instruction-to-instance grounding module that allows LLMs to generate fine-grained target descriptions and reason over spatial context for precise grounding (Fig. 1b). OpenMap offers key advantages in two core aspects:
• We propose a *structural-semantic consensus mapping* strategy (§3.2) to resolve feature inconsistencies commonly introduced during the aggregation stage of existing mapping methods. Our approach incrementally constructs a 3D instance-level semantic map from 2D masks, using structural and semantic consensus as joint criteria for observation fusion. Specifically, two masks are merged only when supported by both global structural and semantic consistency—that is, they are mutually observable from other viewpoints (*i.e.*, exhibit containment relationships in the point cloud) and closely aligned in the vision-language feature space (*i.e.*, refer to the same object or its constituent parts). Guided by these consensus cues, OpenMap iteratively aggregates 2D instances across views, effectively extending 2D vision-language alignment to 3D instance-level representations.
• We introduce a OpenMap-enhanced *Instruction-to-Instance grounding* module (§3.3). Unlike prior approaches that constrain LLM outputs using a predefined scene instance lexicon when interpreting navigation instructions [9, 22, 38], OpenMap enables more precise grounding of natural language to scene instances. This allows large language models to generate fine-grained instance descriptions—for example, interpreting "*I am thirsty*" as "*a cup filled with water*" rather than simply "*cup*." Furthermore, OpenMap incorporates spatial context to support reasoning over candidate instances; for instance, given the instruction "*Get the chair ready—I want to eat*," it can identify the intended chair by considering nearby objects such as a dining table. The synergy between LLMs and OpenMap enables accurate indexing from high-level navigation instructions to specific map instances.

We evaluate OpenMap on the public benchmark ScanNet200 [30], focusing on instance segmentation precision and semantic accuracy. To further assess its target retrieval capabilities in navigation scenarios, we conduct experiments on the Matterport3D dataset [43] using a variety of instruction types, including object-goal [33], demand-driven [36], and language-guided instructions [14]. Compared to state-of-the-art (SOTA) methods, OpenMap consistently outperforms them in both zero-shot semantic mapping and instruction-to-target grounding.

Our contributions are summarized as follows:

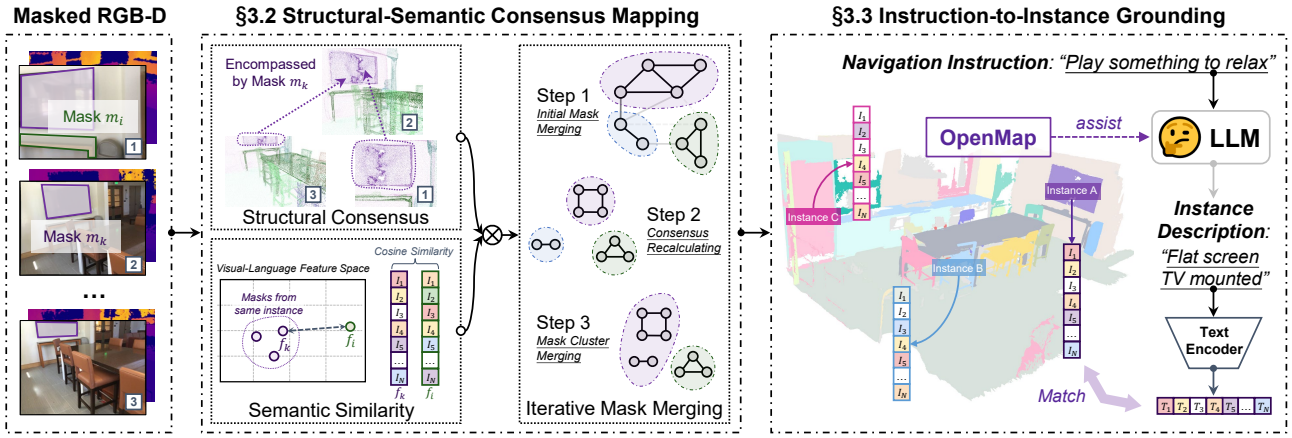

Figure 2: OpenMap Overview. OpenMap takes RGB-D inputs from multiple viewpoints and applies pretrained models to predict 2D masks and extract open-vocabulary features. During semantic mapping (§3.2), it iteratively aggregates 2D masks into 3D instances using a structural-semantic consensus constraint. During instruction grounding (§3.3), an LLM selects the target instance by reasoning over candidate proposals and scene context provided by OpenMap.

• We develop **OpenMap**, an open-vocabulary visual-language mapping framework that bridges LLM-based instruction parsing and 3D instance grounding, enabling precise instruction grounding from free-form navigation commands.
• We propose a novel structural-semantic consensus constraint that jointly leverages global geometric consistency and vision-language semantics to enable fine-grained 3D instance-level mapping.
• We evaluate OpenMap on ScanNet200 and Matterport3D, covering both semantic mapping and target retrieval tasks, and show consistent improvements over existing methods. *Our code is publicly available at https://github.com/openmap-project/OpenMap.*

2 Related Work

Vision-Language Foundation Models. Large-scale VLMs, such as CLIP [29] and BLIP [19], align visual and textual modalities within a shared embedding space [47]. These advances have facilitated open-vocabulary understanding [39] across tasks such as classification and retrieval. Recent efforts extend these capabilities to 2D segmentation, with models like OVSeg [20] and OVSAM [45] incorporating segmentation heads to support instance-level open-vocabulary queries [48]. However, transferring this alignment into 3D space remains challenging due to sparse and incomplete data, especially in navigation settings where environments are incrementally explored [3, 11].

Open-Vocabulary 3D Instance Mapping. Among various forms of open-vocabulary 3D semantic mapping, instance-level mapping is particularly challenging yet essential for accurate instruction grounding. Recent methods for open-vocabulary 3D instance segmentation [15, 25] follow two main paradigms. The first, *3D-to-2D* [12, 27, 34], performs segmentation in 3D and projects results to 2D for feature extraction, but often suffers from poor completeness and semantic consistency due to sparse point clouds. The second, *2D-to-3D* [7, 24, 44], segments 2D frames, uses depth maps for 3D back-projection, and aggregates instance masks via geometric overlap or spatial clustering. While effective, these methods typically overlook semantic similarity in the merging process.

Semantic Mapping for Instruction Grounding. Semantic maps are critical in VLN, as they allow agents to reason over spatial and semantic structures for instruction interpretation and execution [9, 40]. Early works project 2D instance masks onto bird's-eye-view layouts [5, 21], or aggregate features into top-down grids to support open-vocabulary querying [10, 37]. VLMap [9] introduces semantic grid maps and uses LLMs to translate instructions into open-vocabulary object names. ConceptGraphs [7] and HOV-SG [38] further enhance instruction interpretation by constructing spatial graphs over scene instances, enabling explicit modeling of inter-object relationships. However, these methods still rely on predefined labels or coarse semantic features, limiting their ability to support fine-grained, open-vocabulary instruction grounding.

3 Methodology

3.1 Method Overview

An overview of OpenMap is shown in Fig. 2. We follow a generic agent setting for embodied localization [18, 41] and navigation [9, 22], where an agent collects a sequence of RGB-D observations during exploration [16, 17], denoted as $\mathcal{I} = \{I_1, I_2, ..., I_T\}$ and $\mathcal{D} = \{D_1, D_2, ..., D_T\}$. For each frame I_t, we apply an off-the-shelf 2D segmentation model to generate masks $\{m_i^t \mid i = 1, ..., n_t\}$ and use a vision-language model to extract corresponding open-vocabulary features $\{f_i^t \mid i = 1, ..., n_t\}$, where n_t is the number of masks in I_t.

During the semantic mapping stage (§3.2), we apply structural and semantic consensus constraints to determine whether any two masks across the image sequence correspond to the same instance, and iteratively merge those that satisfy both into a unified 3D instance. We then adopt a completeness-guided strategy to select representative masks and aggregate their features to form a holistic semantic embedding for each instance.

In the instruction grounding stage (§3.3), the agent receives a natural language instruction and leverages an LLM to parse it into a target instance description. Unlike conventional methods constrained by predefined vocabularies, our approach allows for

free-form, fine-grained descriptions of navigation goals. Guided by OpenMap, the LLM selects the most semantically relevant instance by reasoning over candidate regions and their associated features within the constructed map.

3.2 Structural-Semantic Consensus Mapping

The core philosophy behind constructing OpenMap is to concurrently consider both spatial structure and semantic feature constraints across different observations.

To visualize this concept, consider a scenario where three observers (*e.g.*, O_A, O_B, and O_C) are examining the same object (*e.g.*, a bunch of roses) from distinct angles. How should they describe it to ascertain that they are indeed looking at the same object?

• Semantically, the observations should exhibit similarities—for instance, O_A might note '*a few green leaves*', O_B could describe '*three roses*', and O_C might see a '*bunch of flowers*'.

• Structurally, there should be a consensus, such that the parts of the instance observed by O_A and O_B are also encompassed in O_C's observation, suggesting that these observations originate from the same instance.

We proceed by modeling these structural-semantic consensus constraints to facilitate precise instance merging.

3.2.1 ***Structural-Semantic Consensus Rate Computing.*** For any two masks m_i and m_j, we evaluate their potential for merging by assessing the structural-semantic consensus rate between them.

Structural Consensus Rate. Drawing on well-established structural consensus analysis [35, 42], we strategically harness the redundancy of observations to ensure structural self-consistency of instances Specifically, following [44], we back-project each mask m_k into a 3D point cloud P_k using depth maps. If the point cloud of an image I overlaps with P_k (*i.e.*, overlap exceeds τ_{obs}), then mask m_k is deemed observable in image I, and we define $\mathcal{I}(m_k)$ as the set of all images that can observe mask m_k.

We then identify the set of images that can simultaneously observe both masks m_i and m_j intended for merging, denoted as $O(m_i, m_j) = \mathcal{I}(m_i) \cap \mathcal{I}(m_j)$. Subsequently, we seek frames capable of supporting the merger of m_i and m_j. Specifically, for an image I_t containing a mask m_k that spatially encompasses both m_i and m_j (*i.e.*, both have at least τ_{sub} of their point clouds within m_k), this subset of images is defined as $S(m_i, m_j) = \{I_t \in O(m_i, m_j) | P_i, P_j \subseteq P_k\}$. Consequently, the structural consensus rate for the two masks m_i and m_j is calculated as the ratio of supporters to observers:

$$R_{struc.}(m_i, m_j) = |S(m_i, m_j)|/|O(m_i, m_j)|. \quad (1)$$

Semantic Similarity Rate. Another criterion for determining whether masks can be merged is their semantic similarity. Diverging from traditional models that rely on a closed vocabulary, VLMs like CLIP capture subtle semantic connections between observations, even different components of the same instance. As shown in Fig. 3, features from the same instance are tightly clustered in the latent space, while spatially close but distinct instances exhibit clear feature separation.

Consequently, by integrating open-vocabulary feature similarity metrics into instance merging, we can effectively reduce the misalignment of different instances that are close in spatial structure but semantically distinct. We define the semantic similarity rate

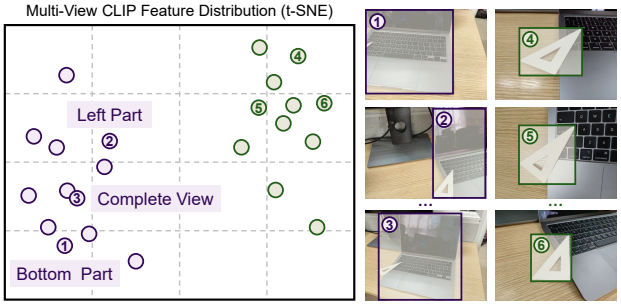

Figure 3: Feature distribution of adjacent objects. Although the triangle ruler is physically attached to the laptop, it remains clearly distinguishable in the vision-language feature space.

between masks m_i and m_j as the cosine similarity between their respective features f_i and f_j:

$$R_{seman.}(m_i, m_j) = \cos(f_i, f_j). \quad (2)$$

Put Together. In mask merging, we balance the above structural consensus and semantic similarity. Specifically, when

$$R_{struc.}(m_i, m_j) * R_{seman.}(m_i, m_j) \geq \tau_{thres}, \quad (3)$$

masks m_i and m_j are considered to form the same instance, where τ_{thres} is a predefined threshold.

3.2.2 ***Iterative Mask Merging.*** After computing pairwise relationships between masks, we iteratively merge them following the general procedure in [44]. This results in a set of 3D point clouds, each representing a distinct instance, with open-vocabulary semantic features aggregated from the corresponding masks.

Specifically, we prioritize merging mask pairs associated with more robust observations (*i.e.*, a larger $|O(m_i, m_j)|$). Therefore, during the iterative process, we set a gradually decreasing threshold for observer counts, N_o. In each merging cycle, two masks, m_i and m_j, are merged into a new mask, $m_{i,j}$, with its corresponding point cloud, $P_{i,j}$, if they not only meet the conditions set by Eq. 3 but also exceed the observer count threshold, $|O(m_i, m_j)| > N_o$.

After each merging cycle, it's necessary to recalculate the structural semantic consensus rate among the newly formed instances. The strategy is as follows:

• Given the structural changes in the new instance, along with altered observational and containment relationships, we update its observers and supporters and recompute $R_{struc.}$.

• Furthermore, due to the aggregation of diverse observations, the semantic features of the combined instance need to be updated. We select the features from the mask that most completely captures the point cloud of the new instance and recalculate $R_{seman.}$.

Each iteration cycle reduces threshold N_o, allowing the instance to incorporate more masks, and this process continues until no further merging is feasible. At the end of the iterations, a list of 3D instances is generated, each linked to multiple 2D masks.

Finally, based on the completeness of the instance's observation, we strategically aggregate features for each instance. Following OpenMask3D [34], we select the top-k masks that best cover the instance and obtain L multi-level crops from the corresponding image areas. Subsequently, features are extracted from these $k * L$

crops using CLIP, with the average pooling results serving as the open-vocabulary feature vector for the instance.

3.3 Instruction-to-Instance Grounding

In everyday life, when colleagues hand over a task, from the arranger's perspective, it is essential to describe the requirements as accurately as possible, rather than being vague. From the executor's perspective, any unclear requirements should be clarified based on contextual information to eliminate ambiguities.

These experiences are equally applicable to embodied navigation tasks, and the parsing of instructions to instances adheres to the following principles:

• When utilizing LLMs for instruction translation, the instructions should be converted into instances described precisely in natural language, instead of choosing from a limited set of dataset labels or an instance dictionary specific to a certain scenario.

• Even with accurate instance descriptions, there may be multiple suitable targets in the scene, and agents should enhance their decision-making by considering contextual information such as the locations of candidates and their surrounding environment.

Next, we will discuss how to utilize OpenMap to assist LLMs in implementing these concepts.

Generic Instruction Parsing. We first utilize an LLM to convert generic navigation instructions into precise instance descriptions that enable target retrieval within OpenMap. Unlike existing methods that translate various types of navigation instructions into targets from a fixed instance dictionary [22], our approach fundamentally differs in that we do not restrict the LLM outputs to predefined dictionary terms, thus fully unleashing its vast knowledge and analytical capabilities.

The textual prompt is composed of the following key components: (1) *Navigation Task Definition*: Similar to existing work in navigation instruction analysis, we provide the background of the navigation task, including environmental information and the format of the instructions. (2) *Instance Description Criteria*: We retain the descriptions of instance characteristics from the original navigation instructions and infer additional features based on environmental information to reduce linguistic ambiguities. (3) *Output Format Constraints*: We streamline the instance descriptions to avoid excessive length, given that most existing VLMs (e.g., the original CLIP) have limited capabilities to process long and complex texts.

For an instance description derived from a navigation instruction, where the VLM encodes the feature vector as f_l, we calculate the similarity with all instance features in OpenMap $\{f_1, f_2, ..., f_N\}$ and rank them. The top N_c are selected as candidate instances.

Instance Selection with OpenMap. Due to the potential presence of multiple instances in a scene that closely match the target description, relying solely on semantic similarity often fails to provide accurate measurements. However, leveraging environmental constraints around candidate instances can significantly enhance the performance of targeted retrieval. A typical case is "*Prepare the chair, I want to eat*". Typically, chairs can found in every room, but the follow-up prompt "*I want to eat*", implies that the chair needed is one near the dining table.

For N_c candidates, we further refine our selection using the instance information provided by OpenMap. Specifically, for each

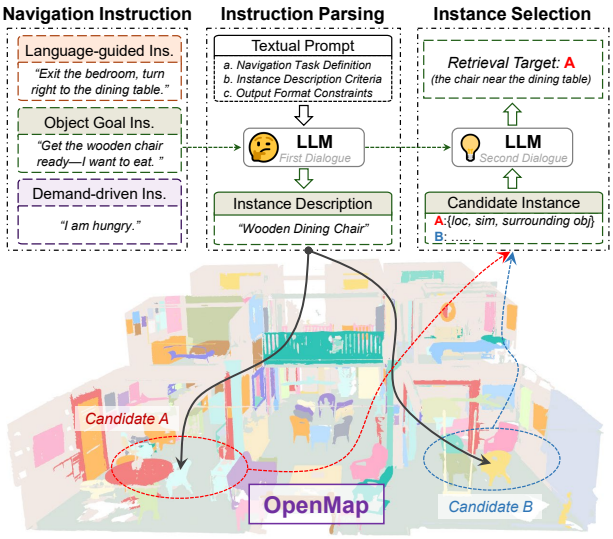

Figure 4: Instruction-to-Instance Grounding pipeline. In the first round, the LLM generates a precise description of the navigation target and retrieves candidate instances from OpenMap. In the second round, it reasons over the candidates and their surrounding context to infer the final target instance.

candidate instance $\mathtt{I}_k$, we use a KD-tree to search within OpenMap for the N_s nearest instances $\mathtt{I}_k^i$ (i = 1, 2, ..., N_s) within a 2-meter radius. Subsequently, we label these $\mathtt{I}_k^i$ (e.g., match with the labels from the LVIS dataset [8]). Note that applying fixed labels here is solely to assist the LLM in instance selection and does not compromise the open-vocabulary querying capabilities of OpenMap.

Next, we initiate a second round of dialogue with the LLM, providing details about the candidate instance and its surrounding objects, and determining the final retrieval target through a multiple-choice format. The template for providing instance information in the prompt is abstracted as follows:

"Candidate Instance $\mathtt{I}_k$: {location: (x_k, y_k, z_k); semantic similarity: r_k; surrounding objects: [$\mathtt{I}_k^i$, location: (x_k^i, y_k^i, z_k^i), label: l_k^i], ...}"

4 Experiments

In this section, we evaluate OpenMap against current SOTA methods in terms of semantic mapping and target retrieval.

4.1 Experimental Setup

Dataset. We utilize the ScanNet200 validation dataset [30] to evaluate OpenMap's semantic mapping capabilities. This dataset features 312 indoor scans across 200 categories, organized into three subsets based on the frequency of instance occurrences, allowing for an effective assessment across a long-tail distribution. Additionally, we examine OpenMap's navigation target retrieval effectiveness with the Matterport3D Semantics dataset [43], following established navigation map research [9][38]. Our evaluation spans 20 scenes—11 from the R2R-CE val-unseen split [14] and 9 from the VLMap evaluation dataset [9]. We construct a comprehensive set of test cases using subsets of navigation instructions from R2R-CE, VLMap, and ALFRED [32], covering three instruction types: object-goal, demand-driven, and language-guided.

Model	Features	Semantic						Class-agnostic		
		AP	AP_{50}	AP_{25}	head(AP)	common(AP)	tail(AP)	AP	AP_{50}	AP_{25}
sup. mask + sup. semantic										
Mask3D [31]	–	26.9	36.2	41.4	39.8	21.7	17.9	39.7	53.6	62.5
sup. mask + z.s. semantic										
OpenScene [27] + Masks	OpenSeg [6]	11.7	15.2	17.8	13.4	11.6	9.9	39.7	53.6	62.5
OpenMask3D [34]	CLIP [29]	15.4	19.9	23.1	17.1	14.1	14.9	39.7	53.6	62.5
z.s. mask + z.s. semantic										
OVIR-3D [24]	CLIP [29]	9.3	18.7	25.0	10.1	9.4	8.1	14.4	27.5	38.8
MaskClustering [44]	CLIP [29]	12.0	23.3	30.1	11.9	10.5	13.8	19.0	36.6	50.8
OpenMap (Ours)	CLIP [29]	**14.3**	**26.0**	**33.3**	**14.5**	**13.8**	**14.7**	**19.8**	**38.0**	**51.8**

Table 1: 3D Instance Segmentation Results on ScanNet200 [30]. Mask3D [31] requires supervised (*sup.*) training on ScanNet200 for mask and semantic extraction. OpenScene [27] + Masks and OpenMask3D [34] depend on masks provided by Mask3D. In a fully zero-shot (*z.s.*) setting, our method, OpenMap, surpasses both OVIR-3D [24] and MaskClustering [44] across all metrics.

Method	SR[%]	SR_4[%]	SR_8[%]	SR_{16}[%]
NLMap [2]	25.1	28.4	31.5	37.1
VLMap [9]	27.2	29.7	32.1	37.5
ConceptGraphs [7]	40.9	43.4	50.6	54.9
OpenMap (Ours)	**49.6**	**58.3**	**68.2**	**73.7**

Table 2: Navigation target retrieval results on Matterport3D [43]. Compared with existing open-vocabulary mapping methods designed for navigation tasks, OpenMap achieves the best performance across all target retrieval success rate metrics.

Baselines. We evaluated OpenMap against SOTA 3D semantic mapping and VLN target retrieval methods. For semantic mapping, **Mask3D** [31] is a representative work trained under supervision on ScanNet200. **OpenScene** [27] is an open-vocabulary 3D scene understanding model that generates per-point feature vectors, for which we average the per-point features within each instance mask, following the approach in [34]. **OpenMask3D** [34] utilizes supervised mask proposals from Mask3D and employs CLIP for open-vocabulary semantic aggregation. **OVIR-3D** [24] and **MaskCLustering** [44] are zero-shot open-vocabulary mapping methods that aggregate instances progressively from 2D to 3D, closely related to our approach. For target retrieval, **NLMap** [2] and **VLMap** [9] utilize LLMs to retrieve targets of known categories on constructed queryable maps. **ConceptGraphs** [7] and **HOV-SG** [38] further enhance the object retrieval and reasoning capabilities by constructing graphs between instances.

Metrics. To validate our mapping accuracy, we report Average Precision (AP) at 25% and 50% Intersection over Union (IoU) thresholds, along with the mean AP from 50% to 95% at 5% intervals. We also evaluate performance in a class-agnostic setting that focuses solely on mask quality, ignoring semantic labels. For target retrieval in OpenMap, we use the Success Rate (SR), defined as successful if the target is retrieved within 1 meter of the ground truth center. Additionally, we measure the top-k Success Rate (SR_k), indicating success within up to k retrieval attempts in the scene.

Implementation Details. To obtain complete object masks rather than overly fragmented results (*i.e.*, all pixels of an object belonging to one mask), we employ CropFormer [23, 28] for 2D segmentation. An intuitive approach for encoding visual-language features for each mask involves cropping bounding boxes and extracting features using CLIP. However, this process generates an excessive number of image patches and, limited by CLIP's processing speed, proves inefficient. We utilize OVSAM [46] to extract features for candidate regions within an image in one go, which is achieved by using the bounding boxes of these regions as prompts for OVSAM. Note that OVSAM features are only used for computing the semantic similarity rate. For a fair comparison of semantic matching capabilities with existing methods, we use features extracted by CLIP [29] ViT-H for the final feature aggregation. We apply post-processing methods from MaskCLustering to filter under-segmented masks and separate disconnected point clusters into distinct instances. Regarding parameter settings, in §3.2.1, the observational threshold for masks $\tau_{obs} = 0.3$, the containment threshold for masks $\tau_{sub} = 0.8$, and the threshold of structural-semantic consensus rate $\tau_{thres} = 0.6$; in §3.2.2, the initial threshold for the number of observers N_o is set at the top 5% of all mask pairs, reducing by 5% in each iteration until the process concludes; in §3.3, the number of candidate instances $N_c = 8$ and the number of neighboring instances $N_s = 5$.

4.2 Mapping Performance

Quantitative Results. Following standard practice in both supervised and zero-shot semantic mapping, we primarily report results on ScanNet200 as it serves as the most widely adopted benchmark, enabling fair comparison with prior work. As shown in Table 1, we categorize the comparison methods into three groups.

Compared to the fully zero-shot OVIR-3D and MaskClustering, OpenMap achieves the highest accuracy on ScanNet200 in both semantic and class-agnostic metrics. Specifically, OpenMap shows a 19.2% improvement in average semantic AP over MaskClustering, which only considers structural features during aggregation. Furthermore, compared to OVIR-3D, which processes local geometric and semantic features frame by frame, OpenMap, which

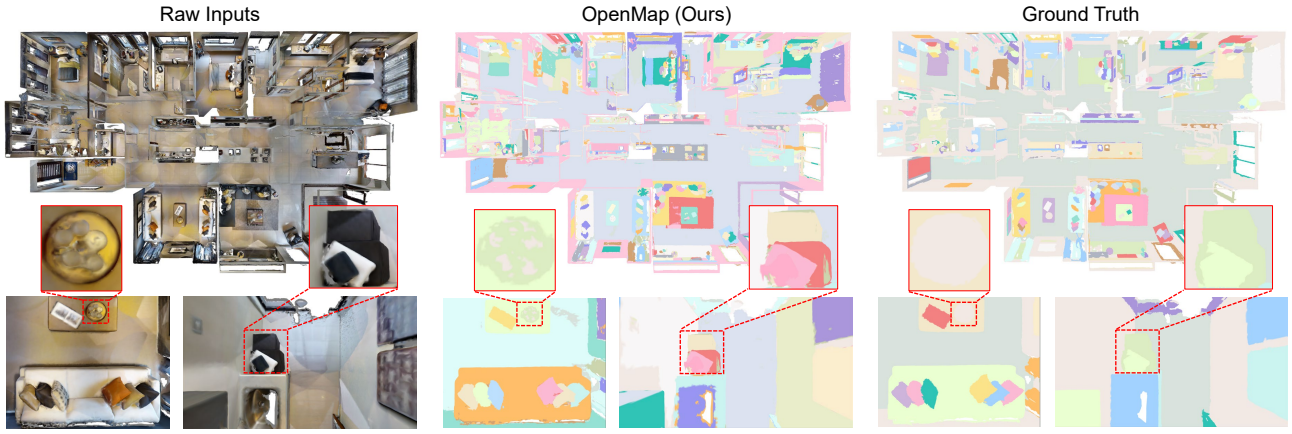

Figure 5: Semantic mapping results of OpenMap on Matterport3D.

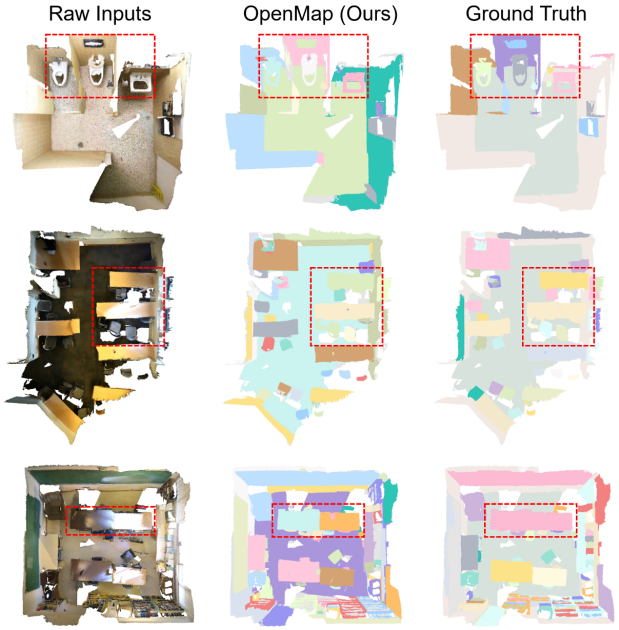

Figure 6: Semantic mapping results on ScanNet200.

incorporates global feature semantic consensus, shows even more pronounced improvements, with increases of 53.8% in semantic AP and 37.5% in class-agnostic AP. Notably, compared to these zero-shot methods, OpenMap maintains consistently high performance across head, common, and tail categories, with an AP gap not exceeding 1.1%. This capability to handle instances of varying frequencies highlights OpenMap's robust open-vocabulary abilities.

In contrast with OpenScene and OpenMask3D, OpenMap, lacking any prior knowledge from ScanNet200, still shows a significant gap in the class-agnostic metrics. Nevertheless, OpenMap significantly surpasses OpenScene in all semantic metrics due to its lack of strategic open-vocabulary feature aggregation. Moreover, OpenMap is close to OpenMask3D in semantic AP, even exceeding it in AP50 and AP25 by +6.1% and +10.2%, respectively. Additionally, OpenMask3D is a semantic mapping method based on 3D-to-2D projection, using a complete 3D point cloud of the scene as a prior. In navigation tasks where the scene is progressively explored, our method, OpenMap, fits more seamlessly, able to integrate into existing navigation tools as a fundamental semantic map to support downstream tasks.

Qualitative Results. As shown in Fig. 5 and Fig. 6, we present qualitative instance segmentation results of OpenMap on Matterport3D and ScanNet200 scenes. OpenMap demonstrates strong performance in two key scenarios: (1) accurately segmenting small objects attached to larger surfaces (*e.g.*, scattered items on a table or glasses on a tray); and (2) preserving the completeness of large objects despite limited viewpoint coverage (*e.g.*, preventing large sofas, tables, and beds from being mistakenly fragmented). Notably, in the first subview of the Matterport3D scene, OpenMap successfully segments individual items on the tray, such as bowls and glasses. However, the ground truth merges these into a single instance, leading to a false negative during evaluation despite the correctness of the prediction.

4.3 Target Retrieval Performance

Quantitative Results. OpenMap is designed to achieve accurate grounding of navigation instructions to scene instances, a task that jointly evaluates the quality of semantic mapping and the effectiveness of instruction-to-instance grounding. We compare OpenMap against two representative visual-language mapping baselines, NLMap [2] and VLMap [9], as well as the recent SOTA method ConceptGraphs [7], in terms of target retrieval success rate. For each trial, all methods first perform full-scene mapping, followed by instruction parsing via a LLM, and then query the target location based on the generated map. To ensure a fair comparison, all methods employ GPT-4 [1] as the LLM.

As shown in Table 2, OpenMap significantly outperforms the baselines across all success rate metrics. In particular, under the SR metric—which reflects the most practical requirement in navigation (i.e., succeeding on the first attempt)—OpenMap surpasses NLMap, VLMap, and ConceptGraphs by +24.5%, +22.4%, and +8.7%, respectively. Among the baselines, VLMap suffers from a coarse feature aggregation strategy (i.e., average pooling over 2D grids), which fundamentally limits the quality of the underlying semantic map and thus its retrieval capability. ConceptGraphs, on the other

Structural.	Semantic.	AP	AP_{50}	AP_{25}
✓	✗	12.2	23.4	30.2
✗	✓	10.1	19.3	26.7
✓	✓	**14.3**	**26.0**	**33.3**

Table 3: Ablation study on Mapping methods.

Ins. Parsing	Ins. Selec.	SR[%]	SR_8[%]	SR_{16}[%]
✗	✗	38.1	61.0	67.6
✓	✗	47.2	68.2	73.7
✗	✓	44.7	61.0	67.6
✓	✓	**49.6**	**68.2**	**73.7**

Table 4: Ablation study on Grounding methods.

hand, relies on pre-defined labels for instances, which restricts its generalization to diverse natural language descriptions.

Qualitative Results. Fig. 1 shows instance-colored segmentation results and similarity heatmaps generated by OpenMap on a Matterport3D scene (ID: 8194nk5LbLH). As shown in Fig. 1a, the semantic map yields accurate instance-level segmentation for both common objects (*e.g.*, sofas) and uncommon structures (*e.g.*, columns). White regions in the overview indicate missing scan data. In Fig. 1b, OpenMap accurately localizes target instances for object-goal, demand-driven, and language-guided navigation tasks. Notably, the heatmaps reveal that even non-top candidates retain task-relevant attributes. For example, in the object-goal case (*e.g.*, "a pair of lounge chairs"), secondary matches preserve key spatial and functional cues such as seating layout and back-to-back configuration. In demand-driven scenarios (*e.g.*, "Where should I eat?"), nearby tables also reflect contextual relevance.

4.4 Ablation Studies

Ablation Study on Mapping. In Table 3, we evaluate the impact of two key components in OpenMap's mapping pipeline (§3.2): structural consensus (Structural.) and semantic similarity (Semantic.). When using only structural consensus, the AP drops from 14.3 to 12.2, yet remains higher than all zero-shot baselines. Moreover, since spatial structural relations are essential for associating instances in 3D reconstruction, structural constraints cannot be entirely removed. To evaluate the performance of using only semantic similarity, we adopt the local structural similarity metric from OVIR-3D as a baseline. The result shows a significant AP drop of 4%. When both components are used jointly, OpenMap achieves the best performance, as expected. These results confirm that both structural and semantic cues are critical to the effectiveness of OpenMap's mapping strategy.

Ablation Study on Instruction Grounding. Table 4 presents an ablation study on two key strategies in the instruction-to-instance grounding (§3.3): unconstrained instruction parsing without restricting LLM outputs to a predefined vocabulary (Ins. Parsing), and OpenMap-assisted instance selection (Ins. Selec.). When Ins. Parsing is removed, we follow the parsing approach used in VLMap as a baseline. Experimental results show that removing both strategies leads to a drop of over 10% in SR compared to the full OpenMap pipeline. Nevertheless, due to the accurate semantic map, our method still outperforms VLMap and NLMap (see Table 2), and achieves better performance than ConceptGraphs on SR_8 and SR_{16}. Individually, removing Ins. Parsing results in a 4.9% decrease in SR, while excluding Ins. Selec. causes a 2.4% drop. Notably, since Ins. Selec. performs filtering within the top-8 most relevant candidates, its removal does not affect SR_8 and SR_{16}. The results demonstrate that OpenMap effectively unlocks the instruction interpretation capabilities of LLMs.

	AP	AP_{50}	AP_{25}
τ_{thres} (0.5 – 0.7)	14.0 ± 0.34	36.9 ± 1.14	49.4 ± 2.41

Table 5: Impact of consensus threshold.

	SR[%]	SR_8[%]	SR_{16}[%]
N_c (4-12)	47.7 ± 1.9	57.6 ± 0.7	68.2 ± 0.0

Table 6: Impact of candidate number.

Ablation Study on Hyperparameters. We conducted additional evaluations to assess the robustness of our algorithm with respect to key hyperparameters. As shown in Table 5, we first examined the effect of the consensus threshold used in the mapping stage. Within the range of 0.5–0.7, the AP variation remains within 0.34. The lowest performance occurs at a threshold of 0.7, yielding an AP of 13.7, while the highest AP of 14.3 is achieved at 0.6, which we adopt in practice. We further evaluated the impact of the number of candidate instances used in instruction-to-instance grounding. As shown in Table 6, SR remains stable within a 1.9% fluctuation when the candidate count ranges from 4 to 12, with the best performance (49.6% SR) observed at 8 candidates. These ablation results demonstrate the consistent and robust performance of OpenMap across a range of hyperparameter settings.

5 Conclusion

We present OpenMap, a zero-shot open-vocabulary visual-language mapping framework for accurate instruction grounding in embodied navigation. To address challenges in instance inconsistency and limited instruction expressiveness, we propose a structural-semantic consensus constraint for robust 3D instance aggregation and an instruction-to-instance grounding module for fine-grained grounding of free-form commands. Extensive experiments on ScanNet200 and Matterport3D demonstrate that OpenMap consistently improves both semantic mapping and instruction to target instance retrieval under zero-shot settings. By enabling precise alignment between natural language instructions and 3D scene instances, OpenMap makes a concrete step toward more reliable and generalizable instruction execution in real-world navigation tasks.

Acknowledgments

We sincerely thank the MobiSense group and the Tsinghua University - Inspur Yunzhou Joint Research Center for New Industrialization and Trustworthy Networks. This work is supported in part by the National Key Research Plan under grant No. 2021YFB2900100, the NSFC under grant No. 62372265, No. 62302254, and No. 62402276.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. 2023. Open-vocabulary queryable scene representations for real world planning. In *ICRA*.

[3] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. 2023. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *CVPR*.

[4] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. 2022. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence* (2022).

[5] Sheng Fan, Rui Liu, Wenguan Wang, and Yi Yang. 2024. Navigation instruction generation with bev perception and large language models. In *ECCV*.

[6] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. 2022. Scaling open-vocabulary image segmentation with image-level labels. In *European conference on computer vision*.

[7] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. 2024. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*.

[8] Agrim Gupta, Piotr Dollar, and Ross Girshick. 2019. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*.

[9] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. 2023. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*.

[10] Jiacui Huang, Hongtao Zhang, Mingbo Zhao, and Zhou Wu. 2024. Ivlmap: Instance-aware visual language grounding for consumer robot navigation. *arXiv preprint arXiv:2403.19336* (2024).

[11] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. 2023. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

[12] Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. 2024. Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. In *European Conference on Computer Vision*.

[13] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*.

[14] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*.

[15] Xin Lai, Yuhui Yuan, Ruihang Chu, Yukang Chen, Han Hu, and Jiaya Jia. 2023. Mask-attention-free transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

[16] Danyang Li, Jingao Xu, Zheng Yang, Qiang Ma, Li Zhang, and Pengpeng Chen. 2023. Leovr: Motion-inspired visual-lidar fusion for environment depth estimation. *IEEE Transactions on Mobile Computing* (2023).

[17] Danyang Li, Jingao Xu, Zheng Yang, Qian Zhang, Qiang Ma, Li Zhang, and Pengpeng Chen. 2022. Motion inspires notion: Self-supervised visual-LiDAR fusion for environment depth estimation. In *Proceedings of the 20th annual international conference on mobile systems, applications and services*.

[18] Danyang Li, Yishujie Zhao, Jingao Xu, Shengkai Zhang, Longfei Shangguan, and Zheng Yang. 2024. EdgeSLAM2: Rethinking edge-assisted visual SLAM with on-chip intelligence. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*.

[19] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*.

[20] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. 2023. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*.

[21] Rui Liu, Xiaohan Wang, Wenguan Wang, and Yi Yang. 2023. Bird's-eye-view scene graph for vision-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

[22] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. 2024. InstructNav: Zero-shot System for Generic Instruction Navigation in Unexplored Environment. In *8th Annual Conference on Robot Learning*.

[23] Qi Lu, Jason Kuen, Shen Tiancheng, Gu Jiuxiang, Guo Weidong, Jia Jiaya, Lin Zhe, and Yang Ming-Hsuan. 2023. High-Quality Entity Segmentation. In *ICCV*.

[24] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. 2023. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *Conference on Robot Learning*.

[25] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. 2024. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[26] Sang-Min Park and Young-Gab Kim. 2023. Visual language navigation: A survey and open challenges. *Artificial Intelligence Review* (2023).

[27] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. 2023. Openscene: 3d scene understanding with open vocabularies. In *CVPR*.

[28] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Wenbo Li, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. 2023. High Quality Entity Segmentation. In *ICCV*.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

[30] David Rozenberszki, Or Litany, and Angela Dai. 2022. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*.

[31] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. 2023. Mask3d: Mask transformer for 3d semantic instance segmentation. In *ICRA*.

[32] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *IEEE/CVF CVPR*.

[33] Jingwen Sun, Jing Wu, Ze Ji, and Yu-Kun Lai. 2024. A survey of object goal navigation. *IEEE Transactions on Automation Science and Engineering* (2024).

[34] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. 2023. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631* (2023).

[35] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. 2000. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*.

[36] Hongcheng Wang, Andy Guan Hong Chen, Xiaoqi Li, Mingdong Wu, and Hao Dong. 2023. Find what you want: Learning demand-conditioned object attribute space for demand-driven navigation. *Advances in Neural Information Processing Systems* (2023).

[37] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. 2023. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International conference on computer vision*.

[38] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. 2024. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

[39] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, et al. 2024. Towards open vocabulary learning: A survey. *IEEE TPAMI* (2024).

[40] Wansen Wu, Tao Chang, Xinmeng Li, Quanjun Yin, and Yue Hu. 2024. Vision-language navigation: a survey and taxonomy. *Neural Computing and Applications* (2024).

[41] Jingao Xu, Hao Cao, Danyang Li, Kehong Huang, Chen Qian, Longfei Shangguan, and Zheng Yang. 2020. Edge assisted mobile semantic visual SLAM. In *IEEE INFOCOM 2020-IEEE Conference on computer communications*.

[42] Jingao Xu, Hao Cao, Zheng Yang, Longfei Shangguan, Jialin Zhang, Xiaowu He, and Yunhao Liu. 2022. {SwarmMap}: Scaling up real-time collaborative visual {SLAM} at the edge. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*.

[43] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. 2023. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[44] Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. 2024. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[45] Haobo Yuan, Xiangtai Li, Chong Zhou, Yining Li, Kai Chen, and Chen Change Loy. 2024. Open-vocabulary SAM: Segment and recognize twenty-thousand classes interactively. In *European Conference on Computer Vision*.

[46] Haobo Yuan, Xiangtai Li, Chong Zhou, Yining Li, Kai Chen, and Chen Change Loy. 2024. Open-Vocabulary SAM: Segment and Recognize Twenty-thousand Classes Interactively. In *ECCV*.

[47] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[48] Chaoyang Zhu and Long Chen. 2024. A survey on open-vocabulary detection and segmentation: Past, present, and future. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).