OPENPOCKET3D: OPEN-ENDED 3D INSTANCE SEGMENTATION VIA INSTANCE-WISE POCKET VOCABULARY

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-ended 3D scene understanding aims to predict object-level semantics without relying on a predefined query vocabulary, whereas standard open-vocabulary methods require textual queries at inference time. While recent open-ended approaches eliminate this dependency by generating class names directly from visual inputs, they often suffer from noisy evidence and high computational overhead due to extensive multi-view representation construction or exhaustive proposal-wise language querying. To overcome these bottlenecks, we propose *OpenPocket3D*, a training-free, plug-and-play framework that seamlessly converts frozen open-vocabulary 3D backbones into open-ended predictors. By decoupling semantic prediction from 3D representation construction, OpenPocket3D shifts the paradigm from building entirely new features to efficient vocabulary selection. Given precomputed 3D masks, our method selects high-visibility 2D views, localizes VLM-generated candidate classes using internal attention maps, and rigorously filters them by enforcing geometric consistency with projected 3D masks. This distills an instance-wise "pocket vocabulary" that reduces the search space for each instance, enabling final predictions based on semantic similarity and geometric validation. Extensive experiments on ScanNet200, ScanNet++, and Replica demonstrate broad applicability across datasets and state-of-the-art performance on established open-ended benchmarks. The project page is available at: openpocket3d.github.io.

1 INTRODUCTION

Open-vocabulary (OV) 3D instance segmentation extends 3D recognition beyond predefined training categories by constructing language-aligned 3D representations at the point, mask, or proposal level. Representative methods then assign semantic categories by comparing these representations with text queries supplied at inference time. This inference protocol is restrictive for open-ended 3D scene understanding, where the target object vocabulary may be unknown before observing the scene. Open-ended 3D instance segmentation therefore aims to produce object-level semantic predictions directly from visual input without relying on a fixed query vocabulary.

Existing open-ended 3D methods reduce this dependence by obtaining visual representations from 2D views and transferring them to 3D proposals (Nguyen et al., 2025b;a). Although this removes the need for a predefined query vocabulary, signals from views with low instance visibility can still introduce noisy evidence, while proposal-level language querying or dense point-level feature lifting can increase the cost of open-ended representation extraction and prediction.

To address these limitations, we propose **OpenPocket3D**, a novel framework that reformulates open-ended 3D instance segmentation as a plug-and-play class selection process integrated with existing open-vocabulary backbones. Our approach stems from a key observation: once 3D masks and language-aligned features are available, open-ended prediction can be recast as constructing and selecting from a *pocket vocabulary* rather than constructing entirely new 3D representations. A pocket vocabulary is a compact set of semantic candidates tailored to each 3D instance.

As illustrated in Fig. 1 and Table 1, OpenPocket3D differs substantially in structure from previous methods. While prior works rely on computationally expensive 2D segmentors and sequential

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: **Comparison with previous works.** OpenPocket3D is a training-free, plug-and-play framework that operates on class-agnostic 3D instances from frozen OV backbones, replacing dense point lifting with compact instance-wise pocket vocabularies and instance score matching. It avoids **Extra 2D Segmentor** and decouples open-ended semantic reasoning from 3D instance construction. For prior methods, the **Candidates** column reports lower bounds on the sizes of their class spaces based on globally distinct decoded names. For OpenPocket3D, it reports the mean pocket vocabulary size. **Runtime** is reported as the per-scene average, with OpenPocket3D achieving the lowest runtime and the highest AP on ScanNet200 among the compared open-ended methods.

Method	Extra 2D Segmentor	Candidates ↓	2D–3D Integration	Runtime ↓	Performance ↑
OE-3DIS	Detic, SAM	≥ 317	Point-wise Feature Lifting	1259.29s	16.0
Any3DIS	SAM2	≥ 370	Point-wise Feature Lifting	910.54s	19.1
OpenPocket3D	**None**	**14.7**	**Instance-wise Candidate Fusion**	**472.17s**	**21.5**

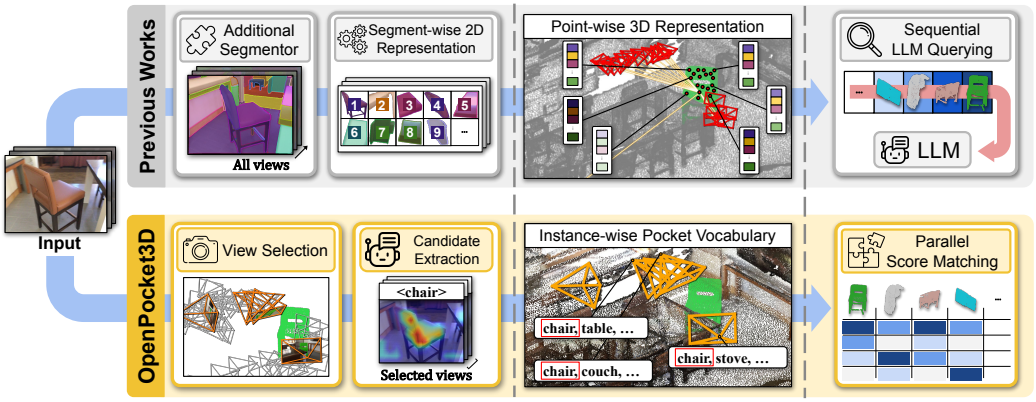


Figure 1: **Comparison of previous works and OpenPocket3D.** Previous works aggregate visual evidence from all associated views with an additional 2D segmentor and query an LLM sequentially for each 3D instance. OpenPocket3D instead selects visibility-ranked views before VLM processing, constructs an instance-wise pocket vocabulary from expanded candidate classes localized by relative attention, and assigns a class to each instance through parallel semantic-geometric score matching.

LLM querying to build open-ended representations, OpenPocket3D streamlines open-ended prediction by identifying visibility-ranked views for each instance. We expand initially decoded candidate hypotheses across selected views and use relative attention to assess whether their spatial support is specific to the projected 3D instance. The validated hypotheses form the instance-wise pocket vocabulary used for subsequent semantic-geometric scoring. This formulation enables efficient, parallel semantic scoring and geometric consistency validation without the need for training additional decoders or executing heavy 2D segmentation pipelines, achieving a superior balance between accuracy and computational efficiency.

The main contributions of **OpenPocket3D** are summarized as follows:

- We reformulate the open-ended semantic prediction stage of 3D instance segmentation as a training-free, plug-and-play class selection process that seamlessly integrates with existing open-vocabulary backbones, replacing complex representation building.
- We propose a compact, instance-specific vocabulary mechanism that expands candidate hypotheses across selected views and validates them through relative attention localization and 3D mask projections. This design improves candidate coverage while suppressing spatially ambiguous evidence before final semantic-geometric scoring.
- We demonstrate the generalizability of OpenPocket3D across multiple backbones and VLMs. It achieves a state-of-the-art 21.5 AP on ScanNet200 and consistently outperforms existing open-ended methods across most metrics on ScanNet200 and ScanNet++.

2 Related Work

3D instance segmentation. With the development of stronger 3D backbones and 3D mask prediction frameworks, 3D instance segmentation has made substantial progress (Qi et al., 2017; He et al., 2020; Vu et al., 2022). Representative methods such as Mask3D (Schult et al., 2023) and ISBNet (Ngo et al., 2023) have achieved strong performance on indoor scene benchmarks through direct 3D mask prediction. However, because these models are trained to predict predefined categories, their ability to generalize to objects beyond the fixed category space remains inherently limited.

Open-vocabulary 3D instance segmentation. In response to this limitation, open-vocabulary 3D scene understanding and instance segmentation methods extend 3D representations into language-aligned semantic spaces to support recognition beyond predefined training categories (Ding et al., 2023; Yang et al., 2024; Huang et al., 2024; Boudjoghra et al., 2025; Zhou et al., 2025). OpenScene (Peng et al., 2023) constructs language-aligned 3D representations by aligning multi-view 2D visual features with 3D points. OpenMask3D (Takmaz et al., 2023) aggregates multi-view CLIP (Radford et al., 2021) image embeddings over class-agnostic 3D instance masks for open-vocabulary instance-wise recognition. Open3DIS (Nguyen et al., 2024) combines class-agnostic 3D instances with mask-guided 2D proposals to improve object prediction, particularly for small or spatially ambiguous objects. However, these methods still rely on text queries supplied at inference time, typically from benchmark category names or a user-provided vocabulary.

Open-ended 3D instance segmentation. To eliminate the dependence on externally provided text queries, open-ended 3D instance segmentation directly produces object-level semantic predictions from visual input. Existing methods (Lin et al., 2024; Ülger et al., 2025) address this setting in the 2D domain by generating open-ended predictions without a predefined vocabulary. In particular, OSM (Yu et al., 2024) conditions class-name generation on class-agnostic instance masks and their masked image regions to directly decode a class name for each proposal. OE-3DIS (Nguyen et al., 2025b) extends this formulation to 3D by lifting visual tokens from masked RGB views, aggregating them across views, and querying a language model for each 3D proposal. Any3DIS (Nguyen et al., 2025a) improves 3D instance construction through 2D mask tracking and superpoint-level refinement. In contrast, our framework directly reuses the 3D masks and semantic features from existing open-vocabulary backbones and constructs a compact instance-wise pocket vocabulary from multiple candidate hypotheses gathered across selected views. Retaining and spatially validating these candidate hypotheses before final scoring enables class selection using frozen 3D features and projected-mask geometry, rather than direct class decoding from a newly constructed open-ended representation.

3 Method

Overview. Given an aligned RGB-D image set, let I_i and D_i denote the RGB image and depth map of the i-th view, respectively. For each instance n, an open-vocabulary backbone provides a 3D mask M_n^{3D} and a language-aligned 3D feature F_n^{3D}. As illustrated in Fig. 2, OpenPocket3D formulates open-ended prediction as selecting the most compatible class from a pocket vocabulary constructed for each instance. First, we rank views by visibility for each 3D instance and use a 2D vision-language model on the selected views to obtain an expanded set of candidate classes and relative attention maps for candidate localization (Sec. 3.1). Next, we construct a pocket vocabulary for each instance and evaluate its candidate classes by jointly considering semantic evidence from the language-aligned 3D feature and geometric consistency across the selected views (Sec. 3.2). We then output the class with the highest ensemble score for each instance (Sec. 3.3).

3.1 Candidate Localization

Visibility-based view selection. To efficiently obtain visual evidence, we first identify visibility-ranked views for each instance. For each instance, we rank views by the instance visibility $v_{n,i}$, measured by the number of projected instance points whose depths are consistent with valid observations in D_i under the fixed depth-consistency criterion. Points outside the image or without valid depth observations are discarded. We select the top k views and denote them by $\{I_{n,i}\}_{i=1}^{k}$, ordered

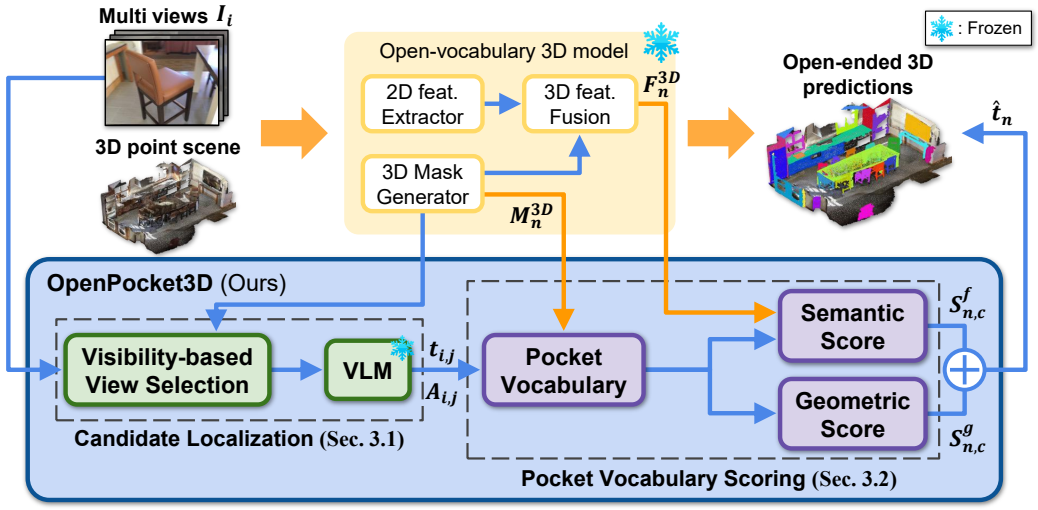


Figure 2: **Overview of OpenPocket3D.** For each 3D instance from an open-vocabulary backbone, we first rank views by visibility and obtain candidate classes and attention maps from the top-ranked views (Sec. 3.1). We then use Relative Attention Maps together with projected 3D masks to construct a pocket vocabulary and compute semantic and geometric scores for each candidate class (Sec. 3.2). For each instance, we output the class with the largest weighted sum of semantic and geometric scores (Sec. 3.3).

by decreasing $v_{n,i}$. This selection focuses VLM processing on high-visibility views and limits the introduction of noisy evidence.

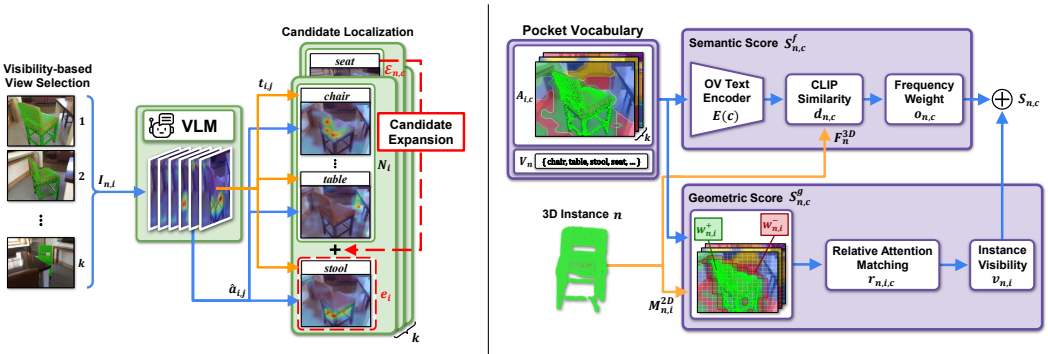
Relative attention localization. Prior open-ended 3D methods rely on additional 2D segmentors for spatial localization, making their localization quality dependent on the resulting 2D segmentation outputs. To avoid this dependency, we exploit the internal attention responses of the same VLM used for candidate generation. For each selected view $I_{n,i}$ of instance n, we use a 2D vision-language model to generate candidate classes $\{t_{i,j}\}_{j=1}^{N_i}$ from the selected RGB view, where N_i is the number of candidate classes produced from that view and j indexes each generated candidate class. However, the initial decoding can omit plausible class names, while the view in which a candidate is generated may provide limited evidence for its localization. To improve candidate coverage, **Candidate Expansion** augments the initial candidates with e_i high-likelihood classes from the same view and extends each candidate's localization to one complementary selected view that reveals the largest number of additional instance points. This allows subsequent localization to consider additional class hypotheses together with complementary visual evidence. From these views, we extract the corresponding spatial attention responses using the layer selected in Sec. C.3 and normalize them over spatial locations to obtain $\hat{a}_{i,j}$. At each spatial location q, $\mu_{i,-j}(q)$ denotes the mean normalized response of the remaining classes in the same view. We then define the Relative Attention Map $A_{i,j}$ as

$$A_{i,j}(q) = \log \frac{\hat{a}_{i,j}(q) + \varepsilon}{\mu_{i,-j}(q) + \varepsilon} \tag{1}$$

By normalizing each class response against the mean response of the remaining classes, Eq. 1 suppresses spatial responses shared across candidates and emphasizes class-specific evidence, enabling candidate localization directly from the VLM responses.

3.2 Pocket Vocabulary Scoring

For reliable class prediction, we construct a pocket vocabulary by filtering noisy semantic candidates according to their geometric consistency with the projected 3D mask. For instance n, let $M_{n,i}^{2D}$ denote the visible 2D projection of the 3D mask M_n^{3D} in the selected view $I_{n,i}$, as illustrated in



(a) **Candidate localization.** (b) **Pocket vocabulary scoring**

Figure 3: **Main modules of OpenPocket3D. (a) Candidate localization.** Views are ranked by instance visibility, and **Candidate Expansion** augments the initial candidates with e_i high-likelihood candidates and extends their localization to one complementary selected view that provides additional visible instance points. Relative Attention Maps are then extracted for each candidate across its expanded view set. **(b) Pocket vocabulary scoring.** Relative Attention Maps are matched with projected 3D masks to construct the pocket vocabulary, followed by semantic and geometric scoring.

Fig. 3(b). We collect the unique class strings obtained from the selected views into an instance-wise candidate pool $\mathcal{C}_n$. We use c to denote a class in $\mathcal{C}_n$. For each class c, $\mathcal{E}_{n,c}$ denotes the expanded view set used for localization, with $A_{i,c}$ representing its Relative Attention Map in view i. Directly measuring candidate support within the projected instance can overestimate the candidate's relevance when the attention also extends to nearby regions. To account for this spatial overlap, we define $w^+_{n,i}(q)$ and $w^-_{n,i}(q)$ at each spatial location q as the normalized weights of the projected visible 3D points and the surrounding context, respectively. Their relative weighting suppresses attention shared with the surrounding context and emphasizes evidence concentrated on the target instance.

The pocket vocabulary V_n is constructed using Relative Attention Matching $r_{n,i,c}$:

$$V_n = \left\{ c \in \mathcal{C}_n \mid \max_{i \in \mathcal{E}_{n,c}} r_{n,i,c} > \sigma(0) \right\}, \quad r_{n,i,c} = \sigma\left(\sum_q [w^+_{n,i}(q) - w^-_{n,i}(q)] A_{i,c}(q) \right) \tag{2}$$

Here, σ denotes the sigmoid function, and $r_{n,i,c}$ measures how strongly the Relative Attention Map of class c aligns with the projected instance compared with its surrounding context. When $M^{2D}_{n,i}$ is empty, we set $r_{n,i,c} = 0$ for that view. This filtering suppresses distractor candidates before scoring, providing more reliable candidate evidence for subsequent class prediction.

Semantic score. To leverage the rich language-aligned 3D semantics captured by the frozen open-vocabulary backbone, we reuse its 3D feature for semantic scoring of the retained candidate classes. For each class $c \in V_n$ retained by the Relative Attention Matching criterion in Eq. 2, the semantic similarity $d_{n,c}$ is computed as the cosine similarity between the 3D feature F^{3D}_n provided by the open-vocabulary backbone and the text embedding $E(c)$. Here, $E(c)$ is obtained from the text encoder associated with the corresponding open-vocabulary backbone, and normalized representations are used when computing the cosine similarity. We apply a softmax over the semantic similarities in V_n and denote the resulting score by $\hat{d}_{n,c}$. In addition, $o_{n,c}$ denotes the number of selected views from which class c is obtained during candidate generation and expansion, and is used as a frequency weight to favor classes repeatedly observed across selected views. The feature-based semantic score $S^f_{n,c}$ is defined by combining the normalized semantic similarity $\hat{d}_{n,c}$ with the frequency weight $o_{n,c}$.

$$S^f_{n,c} = \frac{o_{n,c}}{\sum_{c' \in V_n} o_{n,c'}} \hat{d}_{n,c}. \tag{3}$$

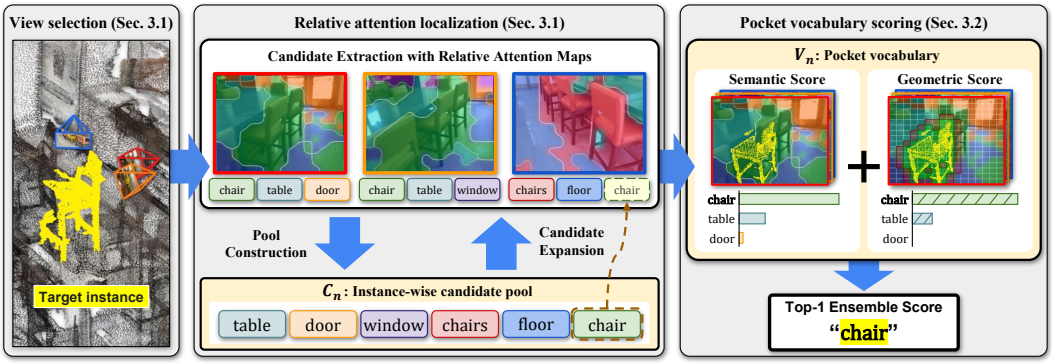


Figure 4: **OpenPocket3D workflow for a target instance.** This example illustrates how OpenPocket3D assigns an open-ended class to a target instance in a scene using 3D instances provided by an open-vocabulary backbone. Views with high instance visibility provide the image evidence. Candidate Expansion adds high-likelihood alternatives beyond the initial decoded candidates and extends each candidate's localization to one complementary selected view. Each candidate in the resulting instance-wise pocket vocabulary is assigned a semantic score derived from language-aligned 3D features and a geometric score derived from mask projection.

Geometric score. Feature–text similarity alone does not capture how strongly a candidate is spatially aligned with the target instance. In contrast to semantic matching based solely on feature-text similarity, we additionally consider Relative Attention Matching $r_{n,i,c}$ across the expanded view set as a complementary spatial cue. Specifically, we aggregate $r_{n,i,c}$ over $\mathcal{E}_{n,c}$, weighting each view by its instance visibility $v_{n,i}$:

$$S^g_{n,c} = \frac{\sum_{i\in\mathcal{E}_{n,c}} v_{n,i}\, r_{n,i,c}}{\sum_{i\in\mathcal{E}_{n,c}} v_{n,i}}. \tag{4}$$

This allows candidates with similar semantic compatibility to be further distinguished by their spatial alignment with the target instance.

3.3 Class Assignment

Given the instance-wise pocket vocabulary, OpenPocket3D performs final prediction over a compact set of semantic candidates tailored to each 3D instance. If $V_n = \emptyset$, the instance is treated as having no reliable open-ended class and is excluded from the prediction set. Otherwise, for each class $c \in V_n$, we combine the feature-based semantic score $S^f_{n,c}$ and the geometric score $S^g_{n,c}$ through a weighted sum, where α controls the balance between the two scores. Figure 4 illustrates this overall workflow for a target instance. The predicted class is the highest-scoring class in the pocket vocabulary:

$$\hat{t}_n = \arg\max_{c\in V_n} S_{n,c}, \quad S_{n,c} = \alpha S^f_{n,c} + (1-\alpha) S^g_{n,c} \tag{5}$$

With this compact prediction space, OpenPocket3D avoids the sequential class decoding used in prior methods and enables efficient class assignment through parallel score matching.

4 Experiments

Datasets. We evaluate OpenPocket3D on three indoor 3D benchmarks: ScanNet200 (Dai et al., 2017; Rozenberszki et al., 2022), ScanNet++ (Yeshwanth et al., 2023), and Replica (Straub et al., 2019). These benchmarks differ in category distribution, data quality, and domain, allowing evaluation under complementary conditions. On ScanNet200, we use the official validation split of 312 scenes and follow the 198-category instance evaluation setting used in prior open-vocabulary 3D

Table 2: **Quantitative comparison on ScanNet200 and ScanNet++.** Open-vocabulary methods that use benchmark category names as queries are included as references. **Query-free** indicates inference without user-provided object-class queries. Following previous work, all open-ended methods are evaluated under the same AP protocol, where predicted classes are mapped to benchmark category names before computing the standard benchmark metrics.

Setting	Method	Query-free	ScanNet200						ScanNet++					
			AP	AP_{50}	AP_{25}	AP_h	AP_c	AP_t	AP	AP_{50}	AP_{25}	AR	RC_{50}	RC_{25}
Open-vocab.	OpenScene*	✗	11.7	15.2	17.8	13.4	11.6	9.9	-	-	-	-	-	-
	OpenMask3D	✗	15.4	19.9	23.1	17.1	14.1	14.9	2.0	2.7	3.4	4.6	8.3	12.4
	Open3DIS	✗	23.7	29.4	32.8	27.8	21.2	21.8	13.1	20.8	24.6	22.1	33.9	39.1
Open-ended	Image-Tagging	✔	10.7	14.3	16.0	11.6	11.0	9.3	9.1	15.5	19.1	16.0	24.8	28.7
	OE-3DIS	✔	16.0	22.0	24.7	20.0	14.3	13.2	18.4	29.4	33.6	23.3	35.2	39.3
	Any3DIS	✔	19.1	-	-	18.2	16.9	**22.8**	20.1	30.4	35.5	-	-	-
	OpenScene* + Ours	✔	16.4	21.6	24.2	23.1	16.2	8.6	-	-	-	-	-	-
	OpenMask3D + Ours	✔	17.8	23.5	26.7	25.4	17.5	9.2	16.5	29.2	41.5	21.4	34.9	45.6
	Open3DIS + Ours	✔	**21.5**	**28.7**	**32.8**	**25.8**	**19.4**	18.7	**21.9**	**37.1**	**45.4**	**27.4**	**43.1**	**50.4**

Table 3: **Quantitative results on Replica.** Open-vocabulary methods are included as references to examine the effectiveness of the proposed extension in a synthetic domain. Query-free follows the definition in Table 2.

Setting	Method	Query-free	AP	AP_{50}	AP_{25}
Open-vocab.	OpenScene*	✗	10.9	15.6	17.3
	OpenMask3D	✗	13.1	18.4	24.2
	Open3DIS	✗	18.5	24.5	28.2
Open-ended	**OpenScene* + Ours**	✔	15.2	20.7	29.6
	OpenMask3D + Ours	✔	17.7	24.7	**32.0**
	Open3DIS + Ours	✔	**18.0**	**25.3**	29.7

instance segmentation work. For ScanNet++, we follow the released open-ended evaluation code of prior open-ended 3D instance segmentation work (Nguyen et al., 2025b;a) and evaluate the validation split with the 84 countable instance categories used by that evaluator. On Replica, following Open3DIS (Nguyen et al., 2024), we evaluate 8 scenes with 48 categories. More evaluation details are provided in Appendix A.2.

Models. Our framework builds on two types of pretrained components: 3D open-vocabulary backbones that provide class-agnostic 3D instance masks and language-aligned 3D features, and 2D vision-language models that provide candidate classes and attention-based localization cues. As 3D open-vocabulary backbones, we use OpenScene, OpenMask3D, and Open3DIS, all of which construct language-aligned 3D representations from multi-view 2D features. Additionally, for OpenScene, we follow the OpenMask3D baseline configuration (Takmaz et al., 2023) by adopting OpenSeg (Ghiasi et al., 2022) for 2D fused features and using Mask3D to pool these features into instance-level representations via class-agnostic 3D instance masks. We denote this configuration as OpenScene* throughout the paper. For 2D vision-language models, we use RAM++ (Huang et al., 2025), LLaVA (Liu et al., 2024a), and InternVL (Chen et al., 2024) to obtain candidate classes and attention maps.

Evaluation protocol. Following previous studies (Nguyen et al., 2025b;a), we evaluate our method under the open-ended setting using the same AP protocol. Because predicted classes may not exactly match the benchmark category names, we map each predicted class to the nearest benchmark category name using Sentence-BERT (Reimers & Gurevych, 2019). The score in Eq. 5 is used only to determine the predicted class for each instance before evaluation and is not used as a confidence score for AP computation. As detailed in Appendix A.2, the mapped predictions are assigned unit confidence and evaluated using the standard 3D instance segmentation metrics of each benchmark. The benchmark category names are not used as candidate classes or text queries during scene-level inference, nor are they used to select views or construct pocket vocabularies. They enter only through the post-prediction evaluation mapping and the validation AP used to choose the fixed global hyperparameters, as detailed in Appendix A.3.

Table 4: **Ablation study of OpenPocket3D components.** We conduct the ablation on ScanNet200 using Open3DIS and RAM++, with the base configuration corresponding to the OE-3DIS Image-Tagging setting. The columns summarize the active components in each setting. $d_{n,c}$ is the semantic similarity between the 3D instance feature and the class text embedding. $v_{n,i}$ denotes the instance visibility used for view selection. $A_{i,j}$ is the Relative Attention Map used for candidate localization. $o_{n,c}$ counts the selected views from which class c is obtained and is normalized over the pocket vocabulary when used as the frequency weight in Eq. 3. $S^g_{n,c}$ denotes the geometric score computed from Relative Attention Matching across selected views, with cross-view localization applied whenever $A_{i,j}$ is enabled. ✓ indicates that the corresponding component is used.

Semantic sim. $\mathbf{d}_{n,c}$	View selection $\mathbf{v}_{n,i}$	Localization $\mathbf{A}_{i,j}$	Freq. weight $\mathbf{o}_{n,c}$	Geometric score $\mathbf{S}^g_{n,c}$	AP	$\mathbf{AP}_{50}$	$\mathbf{AP}_{25}$
✓					10.7	14.3	16.0
✓	✓				11.4	15.2	17.5
✓	✓	✓			12.1	15.9	18.4
✓	✓	✓	✓		12.6	16.5	18.9
	✓	✓		✓	12.0	16.5	19.3
✓	✓	✓	✓	✓	**14.2**	**19.3**	**22.2**

4.1 Experimental Results

Quantitative results on ScanNet200 and ScanNet++. We report quantitative results on ScanNet200 and ScanNet++ under the same open-ended evaluation protocol as prior open-ended methods in Table 2. To keep the main comparison compact, we include the strongest OpenPocket3D configuration for each open-vocabulary backbone; all three configurations use InternVL. Additional ScanNet200 results across RAM++, LLaVA, and InternVL for all three backbones are provided in Table 5, showing that the proposed pipeline improves AP across most VLM/backbone combinations. Under this shared protocol, OpenPocket3D with Open3DIS achieves the best open-ended results on most reported ScanNet200 and ScanNet++ metrics, while other backbones remain competitive with existing open-ended baselines. Overall, the results show that instance-wise pocket vocabularies can turn existing open-vocabulary 3D models into open-ended predictors.

Quantitative results on Replica. Table 3 reports results on Replica, where InternVL is selected for all three backbones. OpenPocket3D remains effective across all three open-vocabulary backbones, preserving or improving most reported metrics relative to the corresponding references. This consistent behavior on a synthetic benchmark further supports the plug-and-play applicability of OpenPocket3D beyond the ScanNet-based settings.

4.2 Ablation and Analysis

Component analysis of OpenPocket3D with RAM++. Table 4 evaluates the contribution of OpenPocket3D components using Open3DIS as the 3D backbone and RAM++ as the candidate source. The first row reports the RAM++ tagging setting from the OE-3DIS Image-Tagging approach (Nguyen et al., 2025b), which obtains 10.7 AP using RAM++ tags extracted from full RGB images and Open3DIS semantic matching. Applying the full OpenPocket3D configuration increases AP to 14.2, improving over this setting by 3.5 AP. The intermediate rows show AP gains from view selection using $v_{n,i}$, attention-guided localization with $A_{i,j}$, frequency weighting using $o_{n,c}$, and geometric scoring using $S^g_{n,c}$ under the same 3D backbone and candidate source. Overall, the ablation indicates that OpenPocket3D improves the RAM++ tagging setting by replacing tag aggregation over full RGB images with view selection and localization specific to each instance, and by combining semantic and geometric evidence for class prediction. Appendix D.2 further isolates the two Candidate Expansion operations: high-likelihood candidate addition and complementary-view localization each improve AP, and their combination achieves the best result.

Generalization across VLMs and backbones. Table 5 evaluates OpenPocket3D across VLMs and open-vocabulary backbones on ScanNet200. Relative to semantic matching over the same selected views, the full pipeline improves AP for eight of nine VLM-backbone pairs. Open3DIS shows the largest gains; with InternVL, AP increases by 4.9, from 16.6 to 21.5. These results indicate that relative attention localization, candidate frequency, and mask-projection consistency provide com-

Table 5: **Analysis of OpenPocket3D across VLMs and open-vocabulary backbones on ScanNet200.** We evaluate whether the gains from OpenPocket3D remain consistent across different VLM candidate sources and open-vocabulary 3D backbones. († denotes a matched baseline that uses the same VLM candidate source, open-vocabulary backbone, selected views, and candidate set as OpenPocket3D but excludes localization $A_{i,j}$, frequency weighting $o_{n,c}$, and geometric scoring $S^g_{n,c}$.)

VLM	OV backbone	Baseline†			OpenPocket3D		
		AP	AP_{50}	AP_{25}	AP	AP_{50}	AP_{25}
RAM++	OpenScene*	9.6	13.0	14.9	10.6(+1.0)	14.5(+1.5)	16.4(+1.5)
	OpenMask3D	10.7	14.1	15.6	12.8(+2.1)	17.0(+2.9)	19.2(+3.6)
	Open3DIS	11.4	15.2	17.5	14.2(+2.8)	19.3(+4.1)	22.2(+4.7)
LLaVA	OpenScene*	10.9	14.6	16.4	11.0(+0.1)	14.6(+0.0)	16.6(+0.2)
	OpenMask3D	12.0	15.9	18.2	11.7(-0.3)	15.7(-0.2)	18.2(+0.0)
	Open3DIS	13.9	18.3	20.5	14.8(+0.9)	19.8(+1.5)	23.1(+2.6)
InternVL	OpenScene*	13.7	17.9	19.9	16.4(+2.7)	21.6(+3.7)	24.2(+4.3)
	OpenMask3D	15.2	19.8	22.8	17.8(+2.6)	23.5(+3.7)	26.7(+3.9)
	Open3DIS	16.6	22.2	24.6	21.5(+4.9)	28.7(+6.5)	32.8(+8.2)

Table 6: **Comparison of open-ended methods across text encoders.** Following prior work, we evaluate open-ended predictions under different semantic mapping spaces using the same evaluation protocol.

Text Encoder	Method	AP	AP_{50}	AP_{25}
BERT	OE-3DIS	13.5	19.0	21.2
	OpenPocket3D (Ours)	**19.1**	**25.2**	**28.5**
CLIP	OE-3DIS	15.9	22.0	24.5
	OpenPocket3D (Ours)	**21.4**	**28.8**	**32.6**
Sentence-BERT	OE-3DIS	16.0	22.0	24.7
	Any3DIS	19.1	–	–
	OpenPocket3D (Ours)	**21.5**	**28.7**	**32.8**

plementary cues beyond candidate-space reduction, supporting plug-and-play use across candidate sources and 3D backbones without retraining.

Evaluation across text encoders. Following prior work (Nguyen et al., 2025b), Table 6 evaluates OpenPocket3D across BERT, CLIP, and Sentence-BERT under the same open-ended protocol. OpenPocket3D achieves the best reported AP with all three encoders, showing that its advantage does not depend on a particular evaluation text encoder.

5 Conclusion

We presented OpenPocket3D, a modular framework that extends existing open-vocabulary 3D instance segmentation models to the open-ended setting. By expanding candidate hypotheses across selected views and validating their spatial support with relative attention, OpenPocket3D constructs compact instance-wise pocket vocabularies from precomputed 3D representations. This allows existing open-vocabulary models to serve as effective backbones for open-ended 3D instance segmentation. These results suggest a simple decomposition: open-ended semantic prediction in 3D instance segmentation can be separated from 3D representation construction and handled as compact instance-wise class selection over existing open-vocabulary backbones.

Limitations. (1) Our method selects views based on 3D masks produced by a frozen open-vocabulary backbone and therefore depends on the quality of these masks. This dependency can limit performance when the backbone misses small objects, produces fragmented masks, or under-segments adjacent instances. (2) Candidate classes are currently generated from 2D observations, so the candidate pool depends on view coverage and the chosen VLM. Deriving candidate classes directly from 3D data is a promising direction for reducing this reliance.

AI USE STATEMENT

We used generative AI tools to assist with literature retrieval and discovery, manuscript drafting and polishing, and LaTeX formatting, including tables and equations. We also used them as coding assistants for implementation, debugging, and running experiments with configurations specified by the authors. We additionally used these tools to obtain feedback on the research methodology, evaluation protocol, and interpretation of experimental results. All AI-assisted text, code, and retrieved information were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

We provide the implementation details, evaluation protocol, and hyperparameters used in our experiments to facilitate reproduction of the reported results. The overall OpenPocket3D pipeline is described in Sec. 3, with implementation and prompting details in Sec. A.1, the evaluation protocol in Sec. A.2, hyperparameters in Sec. A.3 and Table 7, and runtime and memory analysis in Sec. B.1 and Table 8. Additional ablations and analyses are provided in Appendices B–E. We will publicly release the source code for OpenPocket3D after the review process.

REFERENCES

Mohamed El Amine Boudjoghra, Angela Dai, Jean Lahoud, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan, and Fahad Shahbaz Khan. Open-YOLO 3D: Towards fast and accurate open-vocabulary 3d instance segmentation. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

David F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. PLA: Language-driven open-vocabulary 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 540–557, 2022.

Tong He, Dong Gong, Zhi Tian, and Chunhua Shen. Learning and memorizing representative prototypes for 3D point cloud semantic and instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 564–580, 2020.

Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. Open-Set Image Tagging with Multi-Grained Text Supervision. In *Proceedings of the 33rd ACM International Conference on Multimedia (ACM MM)*, pp. 4117–4126, 2025. doi: 10.1145/3746027.3755316.

Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. OpenIns3D: Snap and lookup for 3d open-vocabulary instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Omri Kaduri, Shai Bagon, and Tali Dekel. What's in the image? a deep-dive into the vision of vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14549–14558, 2025.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, 2023.

Zhiwei Lin, Yongtao Wang, and Zhi Tang. Training-free open-ended object detection and segmentation via attention as prompts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 38–55, 2024b.

Tuan Duc Ngo, Binh-Son Hua, and Khoi Nguyen. ISBNet: A 3D point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13550–13559, 2023.

Phuc Nguyen, Minh Luu, Anh Tran, Cuong Pham, and Khoi Nguyen. Any3DIS: Class-agnostic 3D instance segmentation by 2D mask tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3636–3645, 2025a.

Phuc Nguyen, Minh Luu, Anh Tran, Cuong Pham, and Khoi Nguyen. Open-ended 3D point cloud instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2601–2611, 2025b.

Phuc D. A. Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3DIS: Open-vocabulary 3D instance segmentation with 2D mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Songyou Peng, Kyle Genova, Chiyu Max Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded SAM: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3D semantic segmentation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask transformer for 3D semantic instance segmentation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8216–8223, 2023.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat,

Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-vocabulary 3D instance segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Osman Ülger, Maksymilian Kulicki, Yuki Asano, and Martin R. Oswald. Auto-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Thang Vu, Kookhoi Kim, Tung M. Luu, Thanh Nguyen, and Chang D. Yoo. SoftGroup for 3D instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2708–2717, 2022.

Jihan Yang, Runyu Ding, Weipeng Deng, Zhe Wang, and Xiaojuan Qi. RegionPLC: Regional point-language contrastive learning for open-world 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3D indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Qihang Yu, Xiaohui Shen, and Liang-Chieh Chen. Towards open-ended visual recognition with large language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Mingquan Zhou, Chen He, Ruiping Wang, and Xilin Chen. Ov3d-cg: Open-vocabulary 3d instance segmentation with contextual guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5305–5314, 2025.

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

A IMPLEMENTATION DETAILS

A.1 PIPELINE DETAILS

Open-vocabulary backbones. OpenPocket3D is applied without additional training and keeps both the 3D open-vocabulary backbone and the VLM frozen. We start with class-agnostic 3D instances and language-aligned 3D features provided by three backbones. For Open3DIS (Nguyen et al., 2024), we use reproduced Open3DIS outputs constructed from Detic (Zhou et al., 2022) 2D masks and class-agnostic ISBNet (Ngo et al., 2023) 3D instances, following the original design of the backbone. For OpenMask3D (Takmaz et al., 2023) and OpenScene (Peng et al., 2023), we use the same Mask3D proposal masks (Schult et al., 2023) so that the two backbones share identical instance masks. We use reproduced outputs for Open3DIS and OpenMask3D, whereas OpenScene uses the released features from the authors. Open3DIS and OpenMask3D use CLIP (Radford et al., 2021)-based 3D features from a large vision transformer model, while OpenScene uses the released OpenSeg features (Ghiasi et al., 2022).

Candidate localization. For each instance, we select views with high instance visibility $v_{n,i}$ and obtain candidate classes using InternVL2.5-8B (Chen et al., 2024), LLaVA-1.5-7B (Liu et al., 2024a), or RAM++ (Huang et al., 2025). For LLaVA-1.5 and InternVL2.5, we use a fixed object-listing prompt to obtain parseable candidate classes without using benchmark category names or dataset-specific vocabulary:

> *"List all visible objects in this image as simple nouns only. Output format: object1, object2, object3, ... Do not include descriptions, adjectives, or explanations. Only output the comma-separated list."*

For autoregressive VLMs, the initial decoding from each selected view produces N_i candidate classes. To improve candidate coverage without an additional image-conditioned VLM query, Candidate Expansion considers alternative class hypotheses from the same image-conditioned decoding process, removes classes already present in the initial decoded list, and retains e_i additional unique classes according to their autoregressive likelihood. For a class phrase $t_{i,j}$, we define its autoregressive likelihood as

$$p(t_{i,j} \mid I_{n,i}) = \prod_{m=1}^{|t_{i,j}|} p(w_m \mid I_{n,i}, w_{<m}), \tag{6}$$

where w_m denotes the m-th token of $t_{i,j}$.

Candidate Expansion also extends candidate localization beyond the views in which each class is originally obtained. For each class c, we include the views in which c is obtained and one complementary selected view $i^*_{n,c}$ that contributes the largest number of visible instance points not already observed in those views. We therefore define the views used for localization as

$$\mathcal{E}_{n,c} = \left\{ i \mid c \in \{t_{i,j}\}_{j=1}^{N_i+e_i} \right\} \cup \{i^*_{n,c}\}. \tag{7}$$

For each $i \in \mathcal{E}_{n,c}$, candidate c is included in the packed candidate list of view i. This also applies to the complementary view, even when c was not generated there. We then extract its spatial attention response using the same frozen VLM and the attention layer specified in Sec. A.3. The responses are normalized over spatial locations and used to construct the Relative Attention Maps $A_{i,c}$ in Eq. 1. The mean reference response $\mu_{i,-j}(q)$ in Eq. 1 is computed from the remaining candidate responses localized in the same view. High-likelihood candidate addition is not applied to RAM++, while the same cross-view localization procedure is used for its candidate classes.

Pocket vocabulary scoring. For Eq. 2, $w^+_{n,i}$ is obtained by ℓ_1-normalizing the per-cell histogram of projected visible instance points on the VLM attention grid, while $w^-_{n,i}$ is the ℓ_1-normalized one-cell-wide 8-connected ring around the occupied cells. The resulting Relative Attention Matching values $r_{n,i,c}$ determine which candidates are retained in the instance-wise pocket vocabulary. Hyperparameter and attention layer settings are reported in Sec. A.3.

A.2 Evaluation details

Benchmark metrics. For all datasets (Rozenberszki et al., 2022; Yeshwanth et al., 2023; Straub et al., 2019), VLMs, and open-vocabulary backbones, we use the same open-ended evaluation protocol after the predicted class of each instance is determined. The reported metrics follow the standard 3D instance segmentation evaluator for each benchmark. On ScanNet200, we report AP, AP_{50}, AP_{25}, and AP for head, common, and tail categories, denoted by AP_h, AP_c, and AP_t, where the groups follow the category-frequency statistics defined in the benchmark. On ScanNet++, we use the same 84-category instance evaluation set and evaluator as prior open-ended 3D instance segmentation work (Nguyen et al., 2025b;a); although ScanNet++ provides a 100-class semantic category set, the instance segmentation evaluation used here is defined over 84 countable instance categories. We report AP, AP_{50}, AP_{25}, average recall (AR), RC_{50}, and RC_{25} on ScanNet++, and AP, AP_{50}, and AP_{25} on Replica. AP_{50} and AP_{25} are computed at IoU thresholds of 0.50 and 0.25, respectively, and RC_{50} and RC_{25} are the corresponding recall metrics at the same thresholds.

Evaluation protocol. The evaluator takes one predicted class for each 3D instance and maps it to the nearest benchmark category name only during evaluation. To perform this mapping, we encode both the predicted class and the benchmark category name using a pretrained Sentence-BERT model (Reimers & Gurevych, 2019) with approximately 0.1B parameters. We format each string by prepending the phrase "This is a" and compute cosine similarity between normalized sentence embeddings. Each predicted class is then mapped to the nearest benchmark category name. Predictions are removed if their maximum similarity to any benchmark category name is below 0.5. For the text-encoder comparison in Table 6, we additionally evaluate BERT (Devlin et al., 2019) and CLIP (Radford et al., 2021) alongside Sentence-BERT while keeping the remaining evaluation protocol unchanged. The retained instance-category pairs are evaluated with their corresponding 3D instance masks and assigned unit confidence during metric computation, following the OE-3DIS open-ended evaluation protocol (Nguyen et al., 2025b). To handle duplicate predictions, the evaluator follows the same scene-level linear sum assignment step (Crouse, 2016) used in the prior protocol before computing the benchmark metrics. This assignment step is used only for evaluation and does not affect view selection, class generation, pocket vocabulary construction, or class prediction in Eq. 5. Thus, OpenPocket3D changes only how classes are obtained and predicted before evaluation, while the evaluation protocol is kept fixed for fair comparison.

A.3 Hyperparameters

To maintain consistent experimental conditions, all runs were conducted with the same random seed. For view selection, the number of selected views is fixed at $k = 5$, and a projected instance point is counted as visible when its projected depth differs from the valid observed depth by less than 0.2 m. For Relative Attention Map computation, ε is set to 10^{-8} for numerical stability. For attention extraction, InternVL2.5 and LLaVA-1.5 both use layer 15, whereas RAM++ uses its last attention layer. The rationale for these layer choices is discussed in Sec. C.3. Within Candidate Expansion, autoregressive VLMs retain at most five additional unique classes per selected view according to their phrase likelihood, while each candidate class uses at most one complementary selected view for cross-view localization. Pocket vocabulary construction follows the fixed criterion in Eq. 2, retaining a candidate when $r_{n,i,c} > \sigma(0) = 0.5$ in at least one localization view, without converting the Relative Attention Map into a binary support region. For each reported dataset, backbone, and VLM setting, the ensemble weight α in Eq. 5 is selected once within the range from 0.3 to 0.7 using validation AP and is kept fixed during inference. The resulting fixed values of α are reported in Table 7.

B Efficiency and Reduction

B.1 Runtime and Memory Analysis

All runtimes are measured using a single RTX A6000 per run, and the peak reserved GPU memory of each evaluated VLM component is at most 18.9 GB. We report instrumented runtimes for the stages executed by each method during open-ended prediction. Table 8 details the runtime comparison summarized in Table 1. OpenPocket3D achieves the lowest total runtime at 472.17 s/scene, corre-

Table 7: **Ensemble weights used for the reported OpenPocket3D results.** α denotes the weight used to combine the semantic and geometric scores.

Dataset	VLM	Ensemble Weight α		
		OpenScene*	OpenMask3D	Open3DIS
ScanNet200	RAM++	0.3	0.4	0.4
	LLaVA	0.5	0.6	0.5
	InternVL	0.4	0.5	0.4
ScanNet++	RAM++	–	0.4	0.4
	LLaVA	–	0.7	0.5
	InternVL	–	0.6	0.4
Replica	RAM++	0.3	0.5	0.6
	LLaVA	0.5	0.6	0.7
	InternVL	0.4	0.3	0.5

Table 8: **Runtime comparison of open-ended methods.** Inference time is reported as the average time per scene in seconds, with separate measurements for each pipeline stage.

Stage	Module	Time
1. Evidence Prep.	2D mask segmentation	38.70
	2D–3D mask lifting & proposal generation	14.09
	Mask–proposal alignment	266.01
2. VLM Inference	OSM token extraction	125.68
3. 3D Aggregation	Pointwise token lifting & aggregation	721.10
4. Label Prediction	LLM label decoding	93.72
Overall	**Total runtime**	**1259.29**

(a) OE-3DIS

Stage	Module	Time
1. Evidence Prep.	Mask tracking/lifting & refinement	579.00
	Tracked mask–proposal alignment	8.10
2. VLM Inference	OSM token extraction	160.21
3. 3D Aggregation	Pointwise token lifting & aggregation	51.56
4. Label Prediction	LLM label decoding	111.67
Overall	**Total runtime**	**910.54**

(b) Any3DIS

Stage	Module	Time
1. Evidence Preparation	Visibility-based view selection	3.39
2. VLM Inference	VLM candidate generation	463.86
	Cross-view localization	3.88
3. Pocket Vocabulary	Pocket vocabulary construction	< 0.01
4. Class Prediction	Semantic–geometric score matching	1.04
Overall	**Total runtime**	**472.17**

(c) OpenPocket3D

sponding to 2.67× and 1.93× speedups over OE-3DIS and Any3DIS, respectively. This efficiency primarily comes from reusing precomputed 3D masks and language-aligned features, avoiding the additional open-ended representation construction required by prior methods.

B.2 View and candidate reduction

Table 9 reports the reduction achieved by the instance-wise view and candidate filtering used in OpenPocket3D. Compared with the unreduced reference setting, OpenPocket3D substantially reduces both the number of VLM image inputs and the size of the instance-wise pocket vocabulary used for class prediction.

C Hyperparameter and Localization Analysis

C.1 Sensitivity to the ensemble weight

As shown in Table 10, strong performance is maintained over an intermediate range of α around the selected values, rather than being confined to a single optimum. The Open3DIS-based variant also remains competitive with prior open-ended methods as α varies, supporting the conclusion that the reported gains are not driven by narrow hyperparameter tuning.

Table 9: **View and candidate reduction. Candidates** denotes the average number of candidate classes retained in the instance-wise pocket vocabulary, and **Frames** denotes the number of unique RGB frames used as VLM image inputs after taking the union of the top-k views ranked by instance visibility across all instances, where $k = 5$ in all reported experiments. Percentages indicate the fraction of frames retained relative to the full frame set. The N/A rows report unreduced reference settings using the full instance-evaluation category set and all evaluation frames.

Dataset	Backbone	Candidates	Frames
ScanNet200	N/A	198	54,090
	OpenScene*	15.99	33,266(61.5%)
	OpenMask3D	15.99	33,266(61.5%)
	Open3DIS	14.70	43,303(80.1%)
ScanNet++	N/A	84	21,756
	OpenMask3D	18.38	14,374(66.1%)
	Open3DIS	19.07	16,713(76.8%)
Replica	N/A	48	1,600
	OpenScene*	18.12	1,298(81.1%)
	OpenMask3D	18.12	1,298(81.1%)
	Open3DIS	11.48	1,075(67.2%)

Table 10: **Sensitivity to the semantic–geometric mixing weight α.** We report AP on ScanNet200 using InternVL across three open-vocabulary backbones. **Bold** indicates the fixed α used for the reported results.

VLM	Method	Semantic–Geometric Mixing Weight α (AP)								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
InternVL	OpenScene* + Ours	14.0	15.3	15.8	**16.4**	15.7	14.9	14.7	14.6	14.5
	OpenMask3D + Ours	14.7	15.9	17.2	17.8	**17.8**	16.9	16.8	16.6	16.5
	Open3DIS + Ours	19.1	20.2	21.3	**21.5**	20.4	19.2	18.8	18.5	18.4

C.2 Sensitivity to the number of selected views

Table 11 analyzes the effect of varying k for view selection. A small number of selected views limits the visual evidence available for each instance and reduces the reliability of open-ended prediction. Increasing the number of selected views provides more observations, but views with lower instance visibility can also introduce redundant candidate classes or noisy evidence. The results show that $k = 5$ yields the highest AP, while larger values of k do not improve AP and continue to increase the number of processed frames. We therefore use $k = 5$ as the default setting in the reported experiments, as it balances reliable visual evidence with a compact set of VLM image inputs.

Table 11: **Sensitivity to the number of selected views.** We vary k on ScanNet200 using Open3DIS and InternVL. **Frames** denotes the number of unique RGB frames processed after taking the union of selected views across instances. Percentages indicate the fraction of frames retained relative to the full frame set. **Bold** indicates the default $k = 5$ used in the reported experiments.

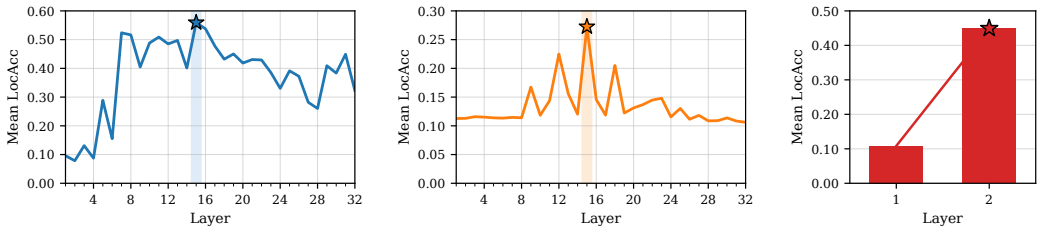
k	1	3	**5**	10	20
AP	18.7	20.7	**21.5**	21.0	19.9
Frames	21,041 (38.9%)	36,719 (67.9%)	**43,303 (80.1%)**	47,203 (87.3%)	51,303 (94.8%)

C.3 Attention layer selection for localization

For this analysis, we use the full ScanNet200 validation frame set, rather than only the views selected by the main pipeline. InternVL2.5 and LLaVA-1.5 each have 32 attention layers in their language models, whereas RAM++ has two attention layers in its tagging head. For each generated candidate class in an RGB frame, we obtain Grounded-SAM (Ren et al., 2024) pseudo masks and use them as reference regions. If multiple pseudo masks are associated with the same candidate class in a frame, we merge them into a single reference mask by taking their union. We then extract, for every layer, the attention map associated with the same candidate class and compute localization accuracy following prior attention grounding analysis (Kaduri et al., 2025). A prediction is counted as correct when the peak location of the attention map lies inside the reference mask. Each layer's score is

the fraction of correct predictions over all generated candidate classes with valid pseudo masks. This layer selection uses only pseudo masks for attention localization and does not use benchmark category annotations or 3D evaluation metrics.

As shown in Fig. 5, localization accuracy varies across layers. InternVL2.5 and LLaVA-1.5 show the clearest localization behavior at layer 15, while RAM++ shows the most reliable localization at its last attention layer. Based on this analysis, we use one fixed attention layer for each model when extracting attention maps in the main experiments.



(a) InternVL2.5 (b) LLaVA-1.5 (c) RAM++

Figure 5: **Localization accuracy across attention layers.** Localization quality varies across attention layers. The selected layer for each model is marked with ⋆, and the shaded regions in **(a)** and **(b)** indicate the selected layer locations. Each subplot uses an independently scaled vertical axis to emphasize the trend across layers within each model.

D Method Ablations

D.1 Across VLM Sources

Table 12 extends the component analysis beyond the RAM++ setting in Table 4 by evaluating InternVL and LLaVA on ScanNet200 with Open3DIS as the backbone. While Table 4 aligns the ablation with the OE-3DIS Image-Tagging setting through RAM++ tags, Table 12 tests the same OpenPocket3D design with additional VLM candidate sources.

Across both VLM sources, Relative Attention Maps with cross-view localization and frequency weighting provide complementary evidence to semantic matching over the same selected views, and the final ensemble achieves the strongest results. Together with the RAM++ ablation in Table 4, these results show that OpenPocket3D remains effective beyond a single tagging model, with semantic similarity, localized candidate support, and projected-mask geometry providing complementary evidence for instance-wise class selection.

Table 12: **Ablation across VLM sources.** We evaluate OpenPocket3D components on ScanNet200 using Open3DIS as the 3D backbone and either InternVL or LLaVA as the candidate source. All settings use the same views selected by instance visibility. The columns indicate whether semantic similarity $d_{n,c}$, localization $A_{i,j}$, frequency weighting based on $o_{n,c}$, and geometric score $S^g_{n,c}$ are used. ✓ denotes an enabled component.

$\mathbf{d}_{n,c}$	$\mathbf{A}_{i,j}$	$\mathbf{o}_{n,c}$	$\mathbf{S}^g_{n,c}$	AP	$\mathbf{AP}_{50}$	$\mathbf{AP}_{25}$
✓				16.6	22.2	24.6
✓	✓			17.4	23.5	26.4
✓	✓	✓		18.1	24.3	27.0
	✓		✓	18.1	24.4	28.3
✓	✓	✓	✓	**21.5**	**28.7**	**32.8**

(a) InternVL

$\mathbf{d}_{n,c}$	$\mathbf{A}_{i,j}$	$\mathbf{o}_{n,c}$	$\mathbf{S}^g_{n,c}$	AP	$\mathbf{AP}_{50}$	$\mathbf{AP}_{25}$
✓				13.9	18.3	20.5
✓	✓			13.5	17.7	20.5
✓	✓	✓		13.6	17.8	20.6
	✓		✓	11.6	15.6	18.4
✓	✓	✓	✓	**14.8**	**19.8**	**23.1**

(b) LLaVA

D.2 Ablation of Candidate Expansion

Unlike prior open-ended 3D pipelines that directly use generated class names or aggregated visual representations for class prediction, OpenPocket3D further exploits the same frozen VLM before

Table 13: **Ablation of Candidate Expansion.** Results are reported on ScanNet200 using Open3DIS and InternVL, with all other OpenPocket3D components fixed.

High-likelihood Candidates	Cross-view Localization	AP	AP_{50}	AP_{25}
		20.6	27.9	32.2
✓		21.0	28.6	32.6
	✓	20.8	27.9	32.1
✓	✓	**21.5**	**28.7**	**32.8**

pocket vocabulary scoring. **Candidate Expansion** augments the initial candidate set with high-likelihood alternatives and extends candidate localization to a complementary selected view that provides additional visible instance points. Table 13 separately evaluates these two operations within Candidate Expansion, with their combination achieving the best overall performance.

E Generalization across Datasets, VLMs, and Backbones

Table 14 extends the before-after analysis in Table 5 to ScanNet++ and Replica, which provide evaluation conditions complementary to those of ScanNet200. On ScanNet++, OpenPocket3D improves or preserves most reported metrics across the tested VLM and backbone combinations, although the changes are smaller and not uniformly positive. On Replica, the full pipeline consistently improves AP, AP_{50}, and AP_{25} for most tested combinations. Overall, these results indicate that the same class-selection pipeline shows similar trends beyond ScanNet200, while the magnitude of improvement depends on the dataset, backbone, and VLM source.

Table 14: **Analysis of OpenPocket3D across VLMs and open-vocabulary backbones on ScanNet++ and Replica. Before** uses the same selected views and selects candidate classes from those views by semantic matching with the 3D instance feature, whereas **After** applies the full OpenPocket3D pipeline. Values in parentheses indicate the change from **Before**.

Dataset	VLM	Method	Before			After		
			AP	AP_{50}	AP_{25}	AP	AP_{50}	AP_{25}
ScanNet++	RAM++	OpenMask3D	12.7	23.1	32.2	13.4(+0.7)	24.5(+1.4)	33.7(+1.5)
		Open3DIS	17.5	29.8	36.0	18.8(+1.3)	31.6(+1.8)	37.4(+1.4)
	LLaVA	OpenMask3D	13.2	24.6	35.5	12.9(-0.3)	24.2(-0.4)	34.7(-0.8)
		Open3DIS	18.1	30.8	38.4	19.1(+1.0)	32.3(+1.5)	39.7(+1.3)
	InternVL	OpenMask3D	16.0	28.2	40.2	16.5(+0.5)	29.2(+1.0)	41.5(+1.3)
		Open3DIS	20.5	34.4	42.7	21.9(+1.4)	37.1(+2.7)	45.4(+2.7)
Replica	RAM++	OpenScene*	7.1	10.6	14.5	9.1(+2.0)	12.0(+1.4)	14.5(+0.0)
		OpenMask3D	7.0	9.5	12.5	10.7(+3.7)	14.2(+4.7)	17.4(+4.9)
		Open3DIS	7.5	11.3	13.9	9.1(+1.6)	13.5(+2.2)	17.2(+3.3)
	LLaVA	OpenScene*	12.8	18.8	20.7	12.5(-0.3)	18.4(-0.4)	20.7(+0.0)
		OpenMask3D	11.0	15.6	20.8	12.0(+1.0)	16.8(+1.2)	21.4(+0.6)
		Open3DIS	9.7	14.0	20.1	13.5(+3.8)	18.8(+4.8)	23.1(+3.0)
	InternVL	OpenScene*	14.0	19.6	23.5	15.2(+1.2)	20.7(+1.1)	29.6(+6.1)
		OpenMask3D	14.2	21.1	28.3	17.7(+3.5)	24.7(+3.6)	32.0(+3.7)
		Open3DIS	11.4	16.4	21.7	18.0(+6.6)	25.3(+8.9)	29.7(+8.0)

F Qualitative results

Figure 6 presents qualitative examples on ScanNet200, ScanNet++, and Replica showing that OpenPocket3D predicts free-form class names without benchmark-category queries. These examples illustrate how open-ended class prediction can produce benchmark-independent, more specific, or alternative object names without relying on benchmark category names at inference time.

G Third-party assets and terms

We use third-party datasets, codebases, pretrained checkpoints, and evaluation components only for academic benchmark evaluation or implementation reference. We cite the original publications

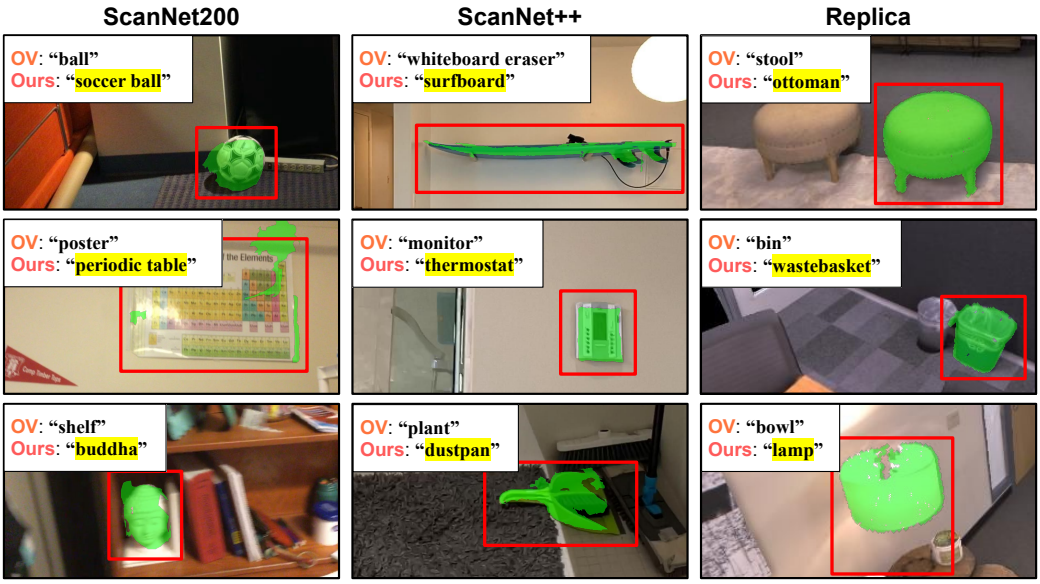


Figure 6: **Qualitative comparisons.** OV denotes the benchmark-category prediction obtained by querying the underlying open-vocabulary model with benchmark category names, and Ours denotes OpenPocket3D's free-form prediction without benchmark-category queries.

throughout Sec. 4 and Appendix A. We do not redistribute third-party datasets, checkpoints, or code. Table 15 summarizes the main third-party assets used in this work and the corresponding licenses or official terms checked against their official sources at submission time.

Table 15: **Third-party assets and terms.** We use or reference these assets only for academic benchmark evaluation, implementation reference, or comparison, and do not redistribute third-party datasets, checkpoints, or code. The listed licenses and terms are based on the official sources checked for this submission.

Asset	Reference	License / terms
Datasets		
Replica	(Straub et al., 2019)	Replica Dataset Research Terms
ScanNet200	(Dai et al., 2017; Rozenberszki et al., 2022)	ScanNet Terms of Use
ScanNet++	(Yeshwanth et al., 2023)	ScanNet++ Terms of Use
Codebases, backbones, and proposal sources		
Detic	(Zhou et al., 2022)	Apache-2.0
ISBNet	(Ngo et al., 2023)	BSD-3-Clause
Mask3D	(Schult et al., 2023)	MIT License
OE-3DIS	(Nguyen et al., 2025b)	Apache-2.0
Open3DIS	(Nguyen et al., 2024)	BSD-3-Clause
OpenMask3D	(Takmaz et al., 2023)	MIT License
OpenScene	(Peng et al., 2023)	Apache-2.0
OpenSeg	(Ghiasi et al., 2022)	Apache-2.0
OV3D-CG	(Zhou et al., 2025)	CC0-1.0
SAM	(Kirillov et al., 2023)	Apache-2.0
SAM2	(Ravi et al., 2025)	Apache-2.0
Checkpoints and auxiliary components		
BERT	(Devlin et al., 2019)	Apache-2.0
CLIP	(Radford et al., 2021)	MIT License
Grounded-SAM	(Ren et al., 2024)	Apache-2.0
Grounding DINO	(Liu et al., 2024b)	Apache-2.0
InternVL2.5	(Chen et al., 2024)	MIT License (Apache-2.0)
LLaVA-1.5	(Liu et al., 2024a)	Llama 2 Community License
RAM++	(Huang et al., 2025)	Apache-2.0
Sentence-BERT	(Reimers & Gurevych, 2019)	Apache-2.0