OVO-SLAM: Open-Vocabulary Online Simultaneous Localization and Mapping

Tomas Berriel Martins
University of Zaragoza

Martin R. Oswald
University of Amsterdam

Javier Civera
University of Zaragoza

Abstract

This paper presents the first Open-Vocabulary Online 3D semantic SLAM pipeline, that we denote as OVO-SLAM. Our primary contribution is in the pipeline itself, particularly in the mapping thread. Given a set of posed RGB-D frames, we detect and track 3D segments, which we describe using CLIP vectors, calculated through a novel aggregation from the viewpoints where these 3D segments are observed. Notably, our OVO-SLAM pipeline is not only faster but also achieves better segmentation metrics compared to offline approaches in the literature. Along with superior segmentation performance, we show experimental results of our contributions integrated with Gaussian-SLAM, being the first ones demonstrating end-to-end open-vocabulary online 3D reconstructions without relying on ground-truth camera poses or scene geometry.

1. Introduction

Simultaneous Localization and Mapping (SLAM) refers to the online estimation of a platform's motion, along with a map of its surrounding environment, from the data streams of its embedded sensors [4]. While early SLAM research targeted robotics, where it is seen as a fundamental step for autonomy [13], its wide industrial adoption came first from augmented and virtual reality [26] and it expands nowadays other use cases [14, 34]. Visual SLAM research, however, has mainly focused on geometric models, processing pipelines and optimizations [7, 36, 42, 57], and much less and mostly recently on the crucial aspect of the scene representation [46, 47, 53], that would further expand its potential for a wider array of tasks.

Over the years, semantic SLAM representations have adopted many forms, e.g., object annotations in 3D point clouds [10, 58], objects as features [3, 38, 48, 54] and semantic segmentations of point cloud maps or implicit 3D representations [28, 30, 47, 64]. All of them, however, have been limited to a pre-defined closed set of categories, which narrows its applicability to real-world situations. Offline semantic 3D reconstructions have also been traditionally closed-set [1, 27, 28], but, after the development of CLIP [44], there has been a surge of work on open-vocabulary 3D representations [2, 23, 37, 40, 41, 52]. In the latter cases, however, the focus on offline processing limits their use in robotics, augmented or virtual reality applications.

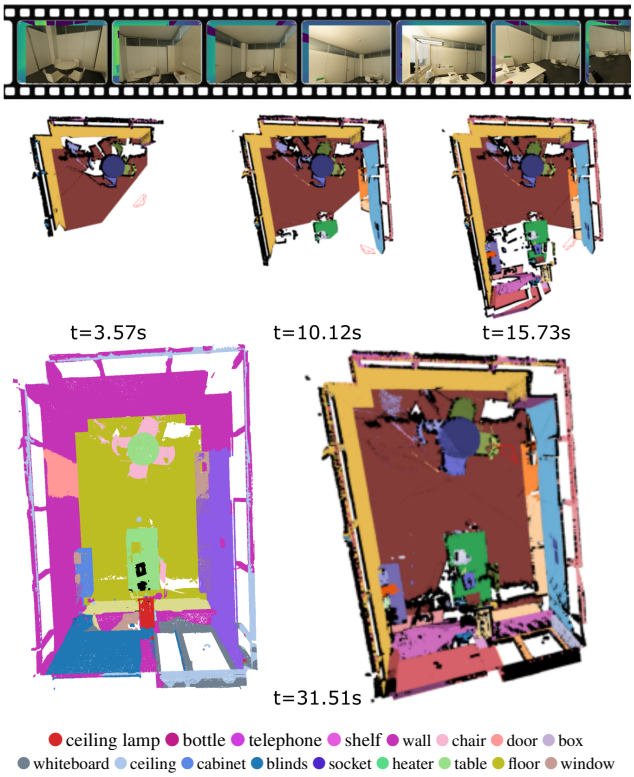

Figure 1. **OVO-SLAM overview.** Given an RGB-D input sequence (**Top**), our method successively reconstructs camera poses, scene geometry and open-vocabulary semantics over time (**Middle:** shows reconstructions at 3 different time instants). Our scene representation is composed of 3D instances with latent visual-language vector annotations inferred from the 2D input. At the end of the sequence the full map is recovered (**Bottom:** the left depicts open-vocabulary representation after mapping to the target vocabulary color-coded by the provided legend; the right depicts 3D segments colored by instances).

In this paper we present the first Open-Vocabulary Online visual SLAM (OVO-SLAM). Our novel pipeline estimates, from a 3D point cloud representation, a set of 3D segments that are assigned a CLIP vector per segment. Specifically, our segments are initialized by back-projecting SAM 2.1 [45] masks, and are tracked by projecting and matching the 3D segments against the 2D masks. The CLIP descriptor of th 3D segment is selected between the descriptors from the keyframes with better visibility. In particular, we also contribute with a novel model to extract per-instance CLIP descriptors from images before assigning them to 3D masks. In addition to being online and faster, our pipeline outperforms the segmentation metrics of offline baselines.

2. Related Work

As discussed earlier, our OVO-SLAM is the first pipeline for open-vocabulary online 3D semantic reconstruction. Unlike prior methods that rely on ground-truth camera poses or scene geometry, our approach integrates pose estimation and geometry reconstruction. Table 1 summarizes recent related works according to these aspects, with further details provided in the remainder of this section.

Open-Vocabulary Image Semantics. Traditionally, semantic segmentation assigned pixels to a fixed set of categories [19, 24]. Seminal work on Contrastive Language Image Pretraining (CLIP) [44], that encodes image and text tokens in a common latent space, revolutionized the field. CLIP features can be classified into any category that language can express by computing their similarity to text inputs. Variations of CLIP improve its performance [9, 17, 20, 62] and the granularity of its features attempting to generate dense feature vectors [18, 51, 63] rather than per-image ones.

Open-Vocabulary 3D semantics. Most approaches to open-vocabulary 3D semantics assume a known 3D point cloud and focus on the semantics. OpenScene [40] leverages OpenSeg [18] to compute CLIP features from images and trains a network to associate 2D pixels with 3D points. For each 3D point they perform average pooling on CLIP vectors from multiple views and then supervise an encoder to directly assign CLIP features to 3D point clouds. OpenMask3D [52] selects k views per object, crops its 2D SAM mask–similarly to LERF [23]–and computes a CLIP vector per mask. CLIP features are then average-pooled over each crop and view. Open3DIS [37] combines SuperPoint [12] with 2D instance segmentations and a 3D instance segmentator to generate multiple 3D instance proposals, describing each instance with CLIP features following OpenMask3D [52]. In contrast, OpenYolo-3D [2] uses a 2D open-vocabulary object detector rather than on 2D instance masks and CLIP features, classifying each object based on the most common class across all views. Although this approach avoids extracting CLIP features, it restricts each scene to the initial set of predefined classes.

Method	Open Vocabulary	Online	3D semantics	Cam. poses	Scene geometry
LERF [23]	✓	✗	✗	✗	✗
LangSplat [41]	✓	✗	✗	✗	✗
OpenScene [40]	✓	✗	✓	✗	✗
OpenMask3D [52]	✓	✗	✓	✗	✗
Open3DIS [37]	✓	✗	✓	✗	✗
HOV-SG [56]	✓	✗	✓	✗	✓
OpenNeRF [15]	✓	✗	✓	✗	✓
NEDS-SLAM [21]	✗	✓	✗	✓	✓
NIS-SLAM [61]	✗	✓	✗	✓	✓
Kimera-VIO [47]	✗	✓	✓	✓	✓
OVO-SLAM (ours)	✓	✓	✓	✓	✓

Table 1. **Overview of open-vocabulary 3D reconstruction baselines.** OVO-SLAM is the first method that jointly estimates open-vocabulary semantics in an online manner with 3D semantic output, while estimating camera poses and scene geometry within a SLAM setting. In contrast, most other works partially use a fixed vocabulary, offline processing, 2D output, ground truth poses or geometry.

Offline 2D semantics from RGB. With the raise in popularity of Neural Radiance Fields (NeRFs) [33] and 3D Gaussian Splatting (3DGS) [22], semantics have been increasingly integrated into these representations. LERF [23] embeds per-object multi-scale CLIP features within NeRF, enabling 2D image searches using language queries. LangSplat [41] uses the Segment Anything Model (SAM) [25] to generate 3 levels of segmentation maps for each viewpoint, remove the background of each segmentation mask, and individually encode them to generate CLIP vectors. For each scene, CLIP features are encoded into smaller dimensional spaces, and a 3DGS [22] representation is augmented with the reduced features to render novel viewpoints and query semantic labels on the decoded 2D rasterizations. SAGA [8] builds on LangSplat to compute CLIP descriptors and incorporates affinity features optimized with a multi-view mask graph to cluster 3D targets; however, it has only been validated on 2D tasks. LEGaussians [49], and Language-Driven Physics-Based Scene Synthesis and Editing via Feature Splatting [43] instead integrate 3DGS with both CLIP and DINOv2 [39] features. While these approaches aim to learn 3D representations, their reliance on multi-point 3D-2D transformation (through rendering or rasterization) to compute 2D semantic features restricts their semantic representation to 2D, as evidenced by the lack of proper 3D evaluation.

Offline 3D semantics from RGB and RGB+D. On the other hand, OpenNeRF [15], optimizes a NeRF to encode the scene representation along with per-pixel CLIP features from OpenSeg. OpenSeg features are projected into 3D to compute the mean and covariance of 3D points. The

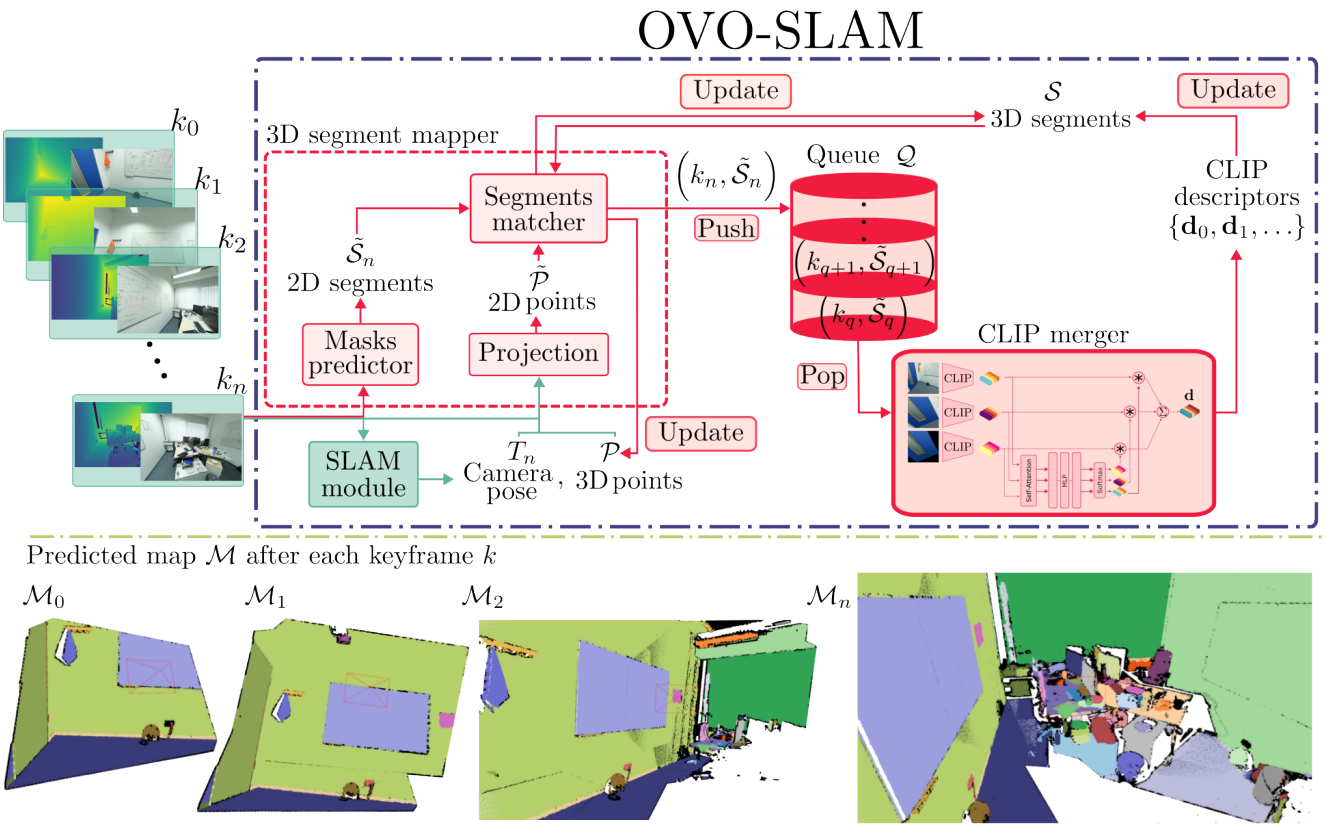

Figure 2. **OVO-SLAM method overview.** From a RGB-D input stream, OVO-SLAM sequentially builds a 3D semantic representation of the scene. It relies on: a SLAM backbone to track the camera pose and build a dense 3D point cloud of the scene; a 3D segment mapper to cluster 3D points into 3D segments; a queue to distribute expensive CLIP extraction, and a novel CLIP merging approach to compute a CLIP descriptor for each 3D segment's observations.

NeRF then renders novel views focusing on areas with high covariance to compute additional OpenSeg features and refine the model. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation (HOV-SG) [56] first computes the full point-cloud of the scene from RGB+D using DBSCAN [16]. It then applies SAM and CLIP to compute local 2D segmentations with corresponding CLIP vectors, which are projected to 3D and fused into the global representation using HDBSCAN [5, 6]. They argue that relying solely on masked segments, as in LangSplat [41], lacks critical contextual information. Rather than computing CLIP descriptors like LeRF or LangSplat, they introduce a novel approach: three CLIP descriptors are computed for each mask based on (1) the full image, (2) the masked segment without background, and (3) the masked segment with background. These descriptors are then combined using a weighted average.

Closed-Vocabulary Online 3D Semantics. To date, online semantic methods have focused exclusively on closed vocabularies. Semantic Fusion [30] was one of the first semantic SLAM pipelines, predicting per-pixel closed-set categories and fusing predictions from different views in 3D space. Fusion++ [31] uses Mask-RCNN [19] to initialize per-object Truncated Signed Distance Functions (TSDFs), building a persistent object-graph representation. In contrast, PanopticFusion [35] combines predicted instances and class labels (including background) to generate pixel-wise panoptic predictions, which are then integrated into a 3D mesh. More recent works, such as those by Menini et al. [32] and ALSTER [55], jointly reconstruct geometry and semantics in a SLAM framework Additionally, NIS-SLAM [61] trains a multi-resolution tetrahedron NeRF to encode color, depth and semantics. NEDS-SLAM [21] is a 3DGS-based SLAM system with embedded semantic features to learn an additional semantic representation of a closed set of classes. Similarly, Hi-SLAM [29] augments a 3DGS SLAM with semantic features. Like earlier offline methods based on NeRF and 3DGS, these approaches represent 2D semantics, with limited capabilities for 3D segmentation or precise 3D object localization.

3. OVO-SLAM

Figure 2 shows an overview of OVO-SLAM. Given an input RGB-D video, a visual SLAM pipeline selects a set of keyframes ($\{k_0, \dots, k_n\}$ in the figure) and estimates their poses and a 3D point cloud representing the scene. From the 3D representation, our OVO-mapping module extracts and tracks 3D segments and assigns per-3D-segment CLIP vectors aggregated from those extracted in the keyframes.

3.1. Map Definition

OVO-mapping builds on top of a geometric visual SLAM pipeline. We assume a parallel-tracking-and-mapping architecture, as first defined by Klein and Murray [26] and adopted by most visual SLAM works [7]. Our input is an RGB-D video $\mathcal{V} = \{f_0, \dots, f_\tau\}$, $f_\tau \in \mathbb{N}^{w \times h \times 3}_{\leq 255} \times \mathbb{R}_{>0}$ standing for the RGB-D frame of size $w \times h$ captured at time step τ. The SLAM front-end estimates, for f_τ, in real time, its pose in the world reference frame. The back-end selects a set of keyframes $\mathcal{K} = \{k_0, \dots k_n\} \subset \mathcal{V}$ from which it estimates in an online manner the scene representation or 'map' $\mathcal{M} = \{\mathcal{T}, \mathcal{P}, \mathcal{S}\}$, composed by the keyframe poses $\mathcal{T} = \{T_0, \dots, T_n\}$, $T_n \in SE(3)$, a point cloud $\mathcal{P} = \{P_0, \dots, P_m\}$ and a set of 3D segments $\mathcal{S} = \{S_0, \dots, S_q\}$[1].

Every map point $P = \left(\begin{bmatrix} x & y & z \end{bmatrix}^\top, l \right)$ is defined by its 3D coordinates $\begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3$ and a discrete label $l \in \{-1, 0, 1, \dots, i\}$, that indicates if the point belongs to any of the $(i+1)$ 3D segments of the map or if it is unassigned to any of them (*i.e.* $l = -1$). Every 3D segment $S = (l, \mathbf{d}, \kappa)$ is identified by its label l, its semantics are described by a CLIP feature $\mathbf{d}$, and stores a heap κ saving the indices of the best keyframes in which S is seen ordered by their visibility scores.

3.2. 3D Segment Mapper

For every new keyframe k_n, we run an image segmentation model that returns a set of 2D segment masks $\tilde{\mathcal{S}}_n = \{s_0, s_1, \dots\}$. We then select the 3D points in k_n's frustum, remove occluded 3D points and finally project the remaining points to k_n obtaining the 2D point set $\tilde{\mathcal{P}}_n$, for which each point $p \in \tilde{\mathcal{P}}_n = \left(\begin{bmatrix} x & y \end{bmatrix}^\top, l \right)$. After that, we compute the mode m of $\tilde{\mathcal{P}}_n$ projected points' labels for every mask $s \in \tilde{\mathcal{S}}_n$. If m receives less votes v than a threshold ϵ, we discard the mask s. If not, two possibilities can happen:

1. If mode m equals to -1, we will create a new 3D segment S_{i+1} label $l = i + 1$, an empty descriptor $\mathbf{d}$ that will be assigned later as described in Section 3.3, and a keyframe heap, $\kappa = \{(n, r)\}$, initialized with k_n's index and s' visibility score r.
2. If mode m is > -1, we will assign the label m to the 2D mask s and every projected 3D point with label -1 will also be assigned the mask label, $l = m$. The keyframe index will be inserted into the heap κ, and stored if it is one of the best views or if κ is still not full.

After this procedure, the 2D masks that where matched to the same 3D segment S are merged.

[1] Note that we use $(\cdot)$ for tuples, $[\cdot]$ for vectors, and $\{\cdot\}$ for sets.

Algorithm 1 3D Segment Mapper

function 3D_SEGMENT_MAPPER($\mathcal{P}$, $\mathcal{S}$, k_n, T_n)
  $\tilde{\mathcal{S}}_n \leftarrow$ segment_keyframe(k_n)
  $\tilde{\mathcal{P}}_n \leftarrow$ project_point_cloud($\mathcal{P}, T_n$)
  $M = \{\}$
  for s in $\tilde{\mathcal{S}}_n$ **do** ▷ For every 2D segment in k_n
    $m, v \leftarrow$ get_label_mode_and_votes($\tilde{\mathcal{P}}_n, s$)
    $M = M \cup \{m\}$
    if $v > \epsilon$ **then** ▷ #votes greater than threshold
      if $m = -1$ **then**
        $S_{i+1} \leftarrow$ new_3D_segment($i + 1, n, s$)
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_{i+1}\}$
      else
        $\mathcal{S} \leftarrow$ update_3D_segment(S_m, n, s)
      end if
    end if
  end for
  $\tilde{\mathcal{S}}_n \leftarrow$ merge_2D_segments($\tilde{\mathcal{S}}_n, M$)
  $\mathcal{P} \leftarrow$ update_point_cloud_labels($\mathcal{P}$)
  return $\mathcal{P}, \mathcal{S}, \tilde{\mathcal{S}}_n$
end function

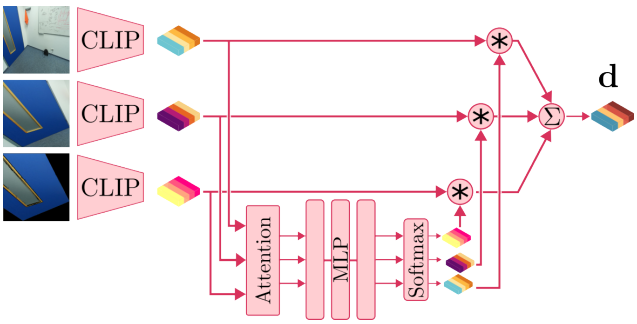

Figure 3. **CLIP merger.** A weight is predicted for each dimension of each of the three CLIP vectors computed for a segmentation mask. Then the final descriptor $\mathbf{d}$ is computed as a weighted average of the three input vectors.

3.3. Queue and CLIP Descriptors

After extracting the 2D masks $\tilde{\mathcal{S}}_n$, the tuple $\left(k_n, \tilde{\mathcal{S}}_n\right)$ is pushed into the queue $\mathcal{Q} = \left\{\left(k_q, \tilde{\mathcal{S}}_q\right), \left(k_{q+1}, \tilde{\mathcal{S}}_{q+1}\right), \dots\right\}$. Keyframes are stored in $\mathcal{Q}$ until processing is available to compute the CLIP descriptors for the top scored 2D segments. When a tuple $\left(k_q, \tilde{\mathcal{S}}_q\right)$ is popped from $\mathcal{Q}$, the matched 2D segments for which k_q is in the κ of their 3D

Method	All		Head		Common		Tail	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
OpenScene [40] (Distilled)‡	14.8	23.0	30.2	41.1	12.8	21.3	1.4	6.7
OpenScene [40] (Ensemble)‡	15.9	24.6	31.7	44.8	14.5	22.6	1.5	6.3
OpenNeRF [15]†	20.4	31.7	35.4	46.2	20.1	31.3	5.8	17.6
HOV-SG [56]†	22.5	34.2	35.9	44.2	23.6	42.3	8.0	16.1
Open3DIS [37] (SigLip)‡	25.6	38.7	**49.7**	**64.4**	22.1	42.4	4.9	9.4
OVO-mapping (ViT-H/14 + SAM)†	22.8	35.5	35.2	45.0	22.1	**44.6**	11.0	16.9
OVO-mapping (ours)†	27.0	**39.1**	45.0	59.9	**25.1**	38.5	11.0	18.8
OVO-SLAM (ours)	**27.1**	38.6	44.1	58.0	25.0	39.0	**12.1**	**18.9**

Table 2. **Evaluation on Replica with the 51 most common labels**. OVO-SLAM gives competitive results and on average outperforms all baselines. †Uses GT camera poses. ‡Uses GT camera poses and 3D geometry.

instance S are selected. Similarly to HOV-SG [56], for each 2D segment, we crop 2 images: one masking the rest of the image out, and another one with the minimum bounding box that contains the full mask (see an example in Fig. 3). We then compute CLIP vectors for the full keyframe and for the two images that we cropped, and the final CLIP vector for the 2D segment is the result of fusing the 3 of them. Differently from HOV-SG [56], that pre-tunes a set of weights on a specified dataset, we train a model to predict the weights for each dimension of the vectors. Fig. 3 illustrates our novel CLIP merger architecture. Specifically, it is composed of self-attention blocks between the three CLIP descriptors. Their output is flattened and feed to an MLP that predicts per-dimension weights for each of the three CLIPs in the input. Finally, the CLIP descriptor $\mathbf{d}$ for a 3D segment S is selected between the 2D segments in the keyframes heap κ, as the one with the smallest aggregated distance to the rest.

Querying. To query the 3D semantic representation, text queries are encoded to CLIP space using the template "This is a photo of a $\{category\}$". Then, we compute the cosine similarity between the CLIP descriptor of the query and the descriptor $\mathbf{d}$ of each 3D segment in $\mathcal{S}$. Fig. 4 illustrates this by showing the most similar 3D segments for a set of different queries.

Figure 4. **Querying CLIP descriptors.** 3D points, in green, belonging to several queries on maps from Replica (top) and ScanNetv2 (bottom). Left–right and top–down, queries are: *sofa*, *chair*, *blackboard*, and *books*.

4. Experiments

Datasets. We ablate our model and train our CLIP merger on ScanNet++ [59], and evaluate our final pipelines OVO-mapping and OVO-SLAM on ScanNetv2 [11] and Replica [50]. ScanNet++ contains 1752×1168 RGB-D images of real indoor scenes with ground-truth 3D meshes and instance and semantic annotations. We use the top 100 semantic labels from the more than 1.6K annotated semantic classes. Its training set has 230 scenes and its validation set has 50 scenes. Each scene has a training camera trajectory and an independent validation one. ScanNetv2 also images real indoor scenes at RGB resolution of 1296×968 and depth resolution of 640×480. It also has ground-truth 3D meshes with ground-truth instance and semantic annotations. ScanNetv2 has two different sets of annotations, with 20 classes (ScanNet20) and 200 classes (ScanNet200). Image blur and noisy depths make this dataset more challenging than ScanNet++. Replica is a synthetic dataset generated from high-fidelity real-world data. Scenes consist of ground-truth 3D meshes with semantic annotations. For all scenes, RGB-D sequences have been rendered at 1200×680.

Metrics. The quantitative performance is evaluated labeling the vertices of ground-truth meshes, and computing 3D mean Intersection Over Union (mIoU) and Accuracy (mAcc) vs ground-truth labels. On Replica, following OpenNerf [15] we report also the metrics splitting the labels in tertiles based on their frequency (*head*, *common*, *tail*). Instead, in ScanNetv2 we report also the metrics weighted by the frequency of the labels in the ground-truth (f-mIoU and f-mAcc). Finally, we also report the average time required to compute the scene's representations measuring clock wall time on a GPU RTX-3090, and for our method

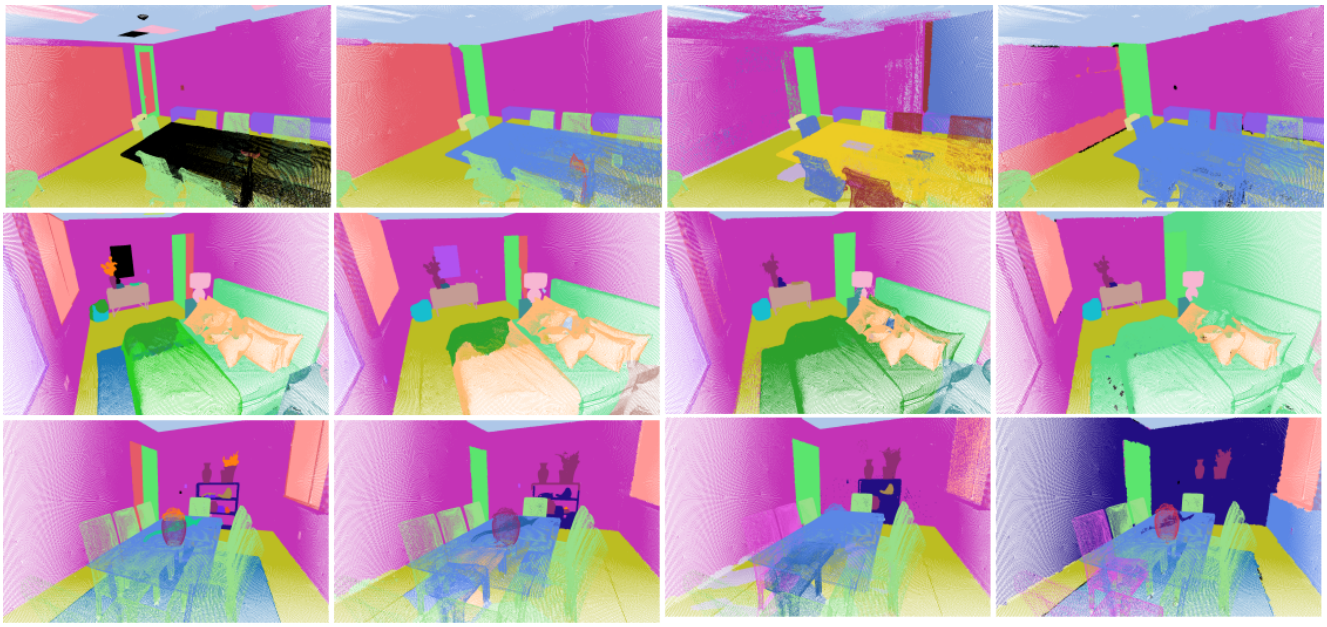

Figure 5. **Qualitative 3D semantic segmentation results on Replica.** OVO-SLAM yields on average more consistent results in comparison to two state-of-the-art offline methods, *e.g.* the blanket on the bed is properly labeled.

we report in seconds the average time spent on each step of the algorithm for each processed frame.

Baselines. Current open-vocabulary SLAM pipelines [21, 61] are based on 2D semantic representations, do not build a semantic 3D representation and hence we cannot benchmark them using 3D metrics. As the fairest baselines for our OVO-SLAM, we chose open-vocabulary offline 3D semantic reconstruction methods, specifically OpenNeRF [15], HOV-SG [56] and Open3DIS [37]. Due to slight differences in metrics computation, we reproduced HOV-SG, and Open3DIS, in both Replica and ScanNetv2. For Open-3DIS we used SigLIP ViT-SO400M rather than its base CLIP ViT-L/14 for a fairer comparison. We report OpenNeRF official metrics on Replica, and were not able to make it converge in ScanNetv2. We show qualitative comparisons against HOV-SG and Open3DIS.

Implementation details. For a fair comparison against baselines that use ground-truth camera poses, we first evaluate our method using ground-truth poses, which we denote as OVO-mapping. After that, we also evaluate our contributions integrated with Gaussian-SLAM [60], which we denote OVO-SLAM. Both implementations use SAM2.1 and SigLip ViT-SO400M for 2D instance segmentation and CLIP extraction respectively. We use the size in pixels of segmented 2D mask as metric of the views quality and show results storing the 10 best keyframes for each 3D segment. Our CLIP merger has a 5 layers self-attention block and a 4 layers MLP. All training and experiments were run in a GPU RTX-3090. Further details on the implementation can be found in Sec. 4.2.

We reproduce previous approaches' [15, 40, 52, 56] evaluation setup. We select as keyframe 1 every 10 frames. For ScanNetv2 we evalaute on a 5 scenes subset (*scene0011_00*, *scene0050_00*, *scene0231_00*, *scene0378_00*, and *scene0518_00*), while for Replica on a 8 scenes subset (*office-0...4*, *room-0...2*). We query the models with the set of classes of each dataset. We assign to each 3D segment the class with higher similarity, and then match our estimated point-cloud to the vertices of ground-truth meshes using KD-tree search with 5 neighbors.

Method	Avg time per scene
HOV-SG [56]	11h 12m
OpenNeRF [15]	19m 3s
OVO-mapping (ours)	**8m 17s**

Table 3. **Average runtime on Replica scenes.** We are not including Open3DIS, as it requires pre-processing with SuperPoint and ISBNet that is difficult to measure.

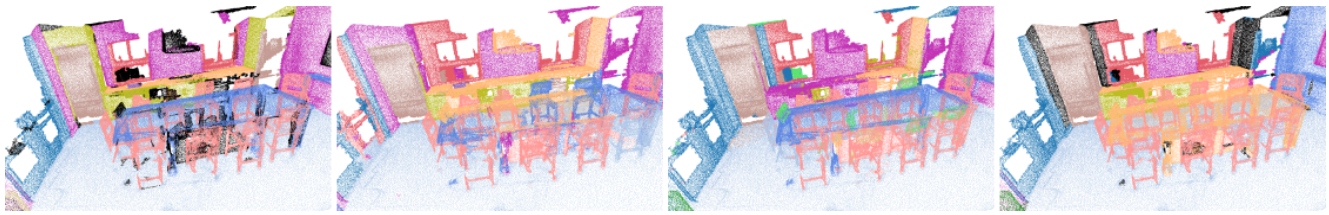

Figure 6. **Qualitative results on ScanNetv2 [11].** We visualize the 3D semantic segmented point cloud of two SotA offline baselines with the ground-truth on ScanNetv2 *scene0011_00*. Our method achieves competitive results despite the more challenging online setting.

Method	ScanNet20				ScanNet200			
	mIoU	mAcc	f-mIoU	f-mAcc	mIoU	mAcc	f-mIoU	f-mAcc
HOV-SG [56]†	34.4	51.1	47.3	61.8	11.2	18.7	27.7	37.6
Open3DIS [37] (SigLip)‡	37.3	**52.8**	57.0	67.9	**17.8**	23.7	27.9	34.1
OVO-mapping (ours)†	**38.1**	50.5	**57.6**	**70.5**	17.2	**25.3**	**45.4**	**56.4**
OVO-SLAM (ours)	29.3	41.1	43.0	59.5	11.8	18.8	30.1	42.6

Table 4. **Quantitative results on ScanNetv2 [11].** OVO-mapping outperforms offline baselines, in particular for frequency-weighted metrics and for the ScanNet200 set. †Uses GT camera poses. ‡Uses GT camera poses and 3D geometry.

4.1. 3D semantic segmentation

Replica. Tab. 2 shows our results, along with those of relevant baselines. Note how OVO-SLAM outperform all offline baselines on both metrics aggregated ('All' column), even if we are not using ground-truth camera poses nor geometry. Thanks to the generalization of CLIP merger , both OVO-SLAM and OVO-mapping have a significantly better performance on Tail categories. As a consequence, OVO-mapping is able to slightly outperform HOV-SG using the same backbones (SAM and ViT-H/14). In addition, observe in Tab. 3 how OVO-mapping is $2\times$ faster than OpenNerf and $80\times$ faster than HOV-SG, which relies on the expensive HDB-Scan. Furthermore, as seen in Fig. 5, OVO-SLAM is able to properly segment and classify challenging 3D instances like chairs and tables, that are commonly confused by other baselines that naively add too much context information into the CLIP descriptors. In contrast with HOV-SG and Open3DIS, OVO-SLAM is also able to properly segment the subtle case of a comforter over a blanket, although it later fail to classify it.

ScanNetv2. Our results on ScanNetv2, summarized in Tab. 4, show how OVO-mapping ourperforms HOV-SG, and even Open3DIS in the set ScanNet20. On the bigger set ScanNet200, OVO-mapping has a similar performance to Open3DIS in mIoU, although it is significantly better in terms of f-mIoU and f-mAcc. Despite our approach missing performance in less common classes, its performance on the most common classes comfortably makes up for that. On the other side, the difference between OVO-SLAM and OVO-mapping is bigger in ScanNetv2 than in Replica (compare Tab. 2 and Tab. 4). This is due to the image blur and noisy depths present in ScanNetv2, that propagates to the estimated camera poses and scene maps.

In Fig. 6 Qualitative analysis, see bottom Fig. 6, show how the three models (OVO-mapping, HOV-SG, and Open3DIS) struggle misclassifying and missegmenting several objects. OVO-mapping labels part of the table as counter and over segments a window, we can see how HOV-SG is predicting parts of the chairs as *picture* and classifying most of the counter and furniture as *wall*, while Open3DIS completely missclasified the table as *counter* and a cabinet as *refrigerator*.

4.2. Ablation Studies

Fixed weights. First, we ablate a set of fixed weights to fuse CLIPs. Following HOV-SG, we use one weight to merge the two local vectors. In contrast, we search for a second weight to combine the local and global features instead of using a Softmax. This has the benefit of removing the dependency between 2D masks segmented in the same image. We perform a grid search of weights for CLIPs' fusion on a set of 5 scenes from ScanNet++ using SAM-2, SigLIP-384, and OVO-mapping.

Nº of best views. After that, we evaluate the impact of using only the best views where each 3D segment has been seen to compute its CLIP descriptor. We evaluate from using only the best image to using all the images where the object has been seen. The results show that neither using only the best nor using all the views are robust enough to noise. The best performance is achieved using subset of

CLIP	SAM	# best views	Seg. [s]	M&T [s]	PP [s]	CLIP [s]	s/KF	mIoU	mAcc	f-mIoU	f-mAcc
ViT-H/14	1-H	10	1.516	0.269	0.085	0.175	2.112	13.3	22.4	20.2	31.7
	2.1-L		0.338	0.252	0.066	**0.135**	0.865	14.1	24.9	27.3	37.7
SigLIP	2.1-t	10	**0.245**	**0.247**	**0.057**	0.204	**0.820**	11.8	25.7	34.2	46.6
	2.1-L		0.339	0.253	0.065	0.233	0.957	14.2	27.0	34.3	45.6
		all	0.337	0.261	0.110	0.367	1.167	**15.8**	**29.6**	**36.3**	**48.6**

Table 5. **Average runtimes and 3D semantic performance on ScanNet++.** We measure the segmentation (Seg); segments matching and tracking (M&T); segments pre processing (PP); CLIPs computation (CLIP); and total seconds per key frame (s/KF).

views, although, the perfect value of will probably be scene and object dependent. Quantitative results can be found in the Supplementary Material. We decide to set use 10 views as a balance to avoid useless computation of CLIP vectors and being resistant to noisy images. This ablation is also performed on a set of 5 scenes from ScanNet++ using SAM-2, SigLIP-384, and OVO-mapping.

SAM and CLIP. Despite expecting SAM 2.1-L and SigLIP to be the best combination, we quantify the impact in our architecture of these models against less powerful alternatives. For 2D segmentation we evaluate SAM [25] with ViT-H encoder (1-H), and SAM 2.1 [45] with Hiera large (2.1-L) and Hiera tiny (2.1-t) image encoders. For CLIP extraction, we evaluate DFN ViT-H/14-378 [17] and SigLIP-SO400 [62] both with input images of 384 pixels. The results are in Tab. 5. This evaluation is performed on a different set of 10 scenes from ScanNet++ to avoid overfitting to the previous 5 scenes.

Method	2D mIoU	2D mAcc	Latency
Only seg image	2.8	5.5	0.000 ms
HOV-SG's	3.4	5.7	0.003 ms
Fixed-weights	3.4	5.8	0.002 ms
CLIP merger (Ours)	**4.9**	**7.4**	0.118 ms

Table 6. **Evaluation on ScanNet++ [59]** using 50 validation scenes with a total of 1.6k semantic classes.

CLIP merger. To address the limitations of using a fixed set of weights to fuse CLIP descriptors across diverse objects and scenes, we train a neural network to predict these weights based on the image content.

To train the model, we use SAM 2.1 to compute segmentation masks on images and match these with their ground-truth 2D semantic labels. For each set of three CLIP descriptors, the CLIP merger predicts a weight for each dimension of each descriptor, merging them into a single CLIP. We optimize the model by minimizing the cosine similarity between the computed CLIP descriptor and the CLIP descriptor of the semantic 2D label corresponding to the input CLIP vectors. The model is trained for 15 epochs on the 230 scenes of ScanNet++, predicting only from the set of the 100 most common labels in the dataset.

We compare its performance and generalization against the naive approach of using only the segmented image [41]; HOV-SG's approach; and our variation of HOV-SG's using three fixed weights. The evaluation is performed on the 50 scenes validation set of ScanNet++ on the complete set of 1.6k labels. The results, in Tab. 6, show that our merging strategy outperforms the other alternatives.

Limitations. Despite its good performance on 3D semantic segmentation, OVO-SLAM performs a naive detection and tracking of 3D instances, that could certainly be improved. We also lack a system to fuse 3D segments for performing loop closure, as typically done by SLAM pipelines in long exploratory sequences. Furthermore, its current precessing time of $\tilde{1}$ frame per second limits its application to platforms with multiple GPUs available. Finally, note that our approach to train CLIP merger will bias it toward CLIP descriptors of objects classes, losing some of the generalization properties of CLIP vectors. Ideally, to maintain these properties, the training should be done on a bigger dataset, including both object classes and other semantic properties.

5. Conclusions

In this paper we present OVO-SLAM, the first open-vocabulary online SLAM with a semantic 3D representation, based on 3D segments described by CLIP features. We propose a novel pipeline to segment 3D points from 2D masks, and track them across different keyframes. Additionally, we developed a new approach to assign CLIP descriptors to our 3D segments. For each 2D segment in each keyframe, we compute a single CLIP descriptor by taking a weighted sum of CLIPs from the natural image, the masked segment and a bounding box around it. The weights are predicted from a deep model, which we show to be more effective than alternative approaches. We outperform offline baselines in both computation and segmentation metrics on the Replica and ScanNet datasets. We believe that our work, that bridges SLAM and open-vocabulary representations, opens both fields to a broader range of potential applications.

References

[1] Sid Yingze Bao and Silvio Savarese. Semantic structure from motion. In *CVPR 2011*, pages 2025–2032. IEEE, 2011. (see page: 1)

[2] 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, 2024. (see pages: 1, 2)

[3] Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic slam. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1722–1729. IEEE, 2017. (see page: 1)

[4] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016. (see page: 1)

[5] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013. (see page: 3)

[6] Ricardo JGB Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1): 1–51, 2015. (see page: 3)

[7] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. (see pages: 1, 4)

[8] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. *arXiv preprint arXiv:2312.00860*, 2023. (see page: 2)

[9] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. (see page: 2)

[10] Javier Civera, Dorian Gálvez-López, Luis Riazuelo, Juan D Tardós, and Jose Maria Martinez Montiel. Towards semantic slam using a monocular camera. In *2011 IEEE/RSJ international conference on intelligent robots and systems*, pages 1277–1284. IEEE, 2011. (see page: 1)

[11] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. (see pages: 5, 7)

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. (see page: 2)

[13] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. (see page: 1)

[14] Kamak Ebadi, Lukas Bernreiter, Harel Biggie, Gavin Catt, Yun Chang, Arghya Chatterjee, Christopher E Denniston, Simon-Pierre Deschênes, Kyle Harlow, Shehryar Khattak, et al. Present and future of slam in extreme environments: The darpa subt challenge. *IEEE Transactions on Robotics*, 2023. (see page: 1)

[15] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, Marc Pollefeys, and Federico Tombari. OpenNeRF: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views. In *International Conference on Learning Representations*, 2024. (see pages: 2, 5, and 6)

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, pages 226–231, 1996. (see page: 3)

[17] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. In *ICLR*, 2024. (see pages: 2, 8)

[18] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. (see page: 2)

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. (see pages: 2, 3)

[20] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. (see page: 2)

[21] Yiming Ji, Yang Liu, Guanghu Xie, Boyu Ma, Zongwu Xie, and Hong Liu. Neds-slam: A neural explicit dense semantic slam framework using 3d gaussian splatting. *IEEE Robotics and Automation Letters*, 2024. (see pages: 2, 3, and 6)

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. (see page: 2)

[23] Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. (see pages: 1, 2)

[24] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019. (see page: 2)

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and

Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. (see pages: 2, 8)

[26] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. (see pages: 1, 4)

[27] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 703–718. Springer, 2014. (see page: 1)

[28] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. (see page: 1)

[29] Boying Li, Zhixi Cai, Yuan-Fang Li, Ian Reid, and Hamid Rezatofighi. Hi-slam: Scaling-up semantics in slam with a hierarchically categorical gaussian splatting. *arXiv preprint arXiv:2409.12518*, 2024. (see page: 3)

[30] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017. (see pages: 1, 3)

[31] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *2018 international conference on 3D vision (3DV)*, pages 32–41. IEEE, 2018. (see page: 3)

[32] Davide Menini, Suryansh Kumar, Martin R Oswald, Erik Sandström, Cristian Sminchisescu, and Luc Van Gool. A real-time online learning framework for joint 3d reconstruction and semantic segmentation of indoor scenes. *IEEE Robotics and Automation Letters*, 7(2):1332–1339, 2021. (see page: 3)

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. (see page: 2)

[34] Javier Morlana, Juan D Tardós, and José MM Montiel. Topological slam in colonoscopies leveraging deep features and topological priors. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 733–743. Springer, 2024. (see page: 1)

[35] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4205–4212. IEEE, 2019. (see page: 3)

[36] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. (see page: 1)

[37] Phuc 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*, pages 4018–4028, 2024. (see pages: 1, 2, 5, 6, and 7)

[38] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, 2018. (see page: 1)

[39] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. (see page: 2)

[40] 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. (see pages: 1, 2, 5, and 6)

[41] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. (see pages: 1, 2, 3, and 8)

[42] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics*, 34(4):1004–1020, 2018. (see page: 1)

[43] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Language-driven physics-based scene synthesis and editing via feature splatting. In *European Conference on Computer Vision (ECCV)*, 2024. (see page: 2)

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. (see pages: 1, 2)

[45] 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. *arXiv preprint arXiv:2408.00714*, 2024. (see pages: 2, 8)

[46] David M Rosen, Kevin J Doherty, Antonio Terán Espinoza, and John J Leonard. Advances in inference and representation for simultaneous localization and mapping. *Annual*

Review of Control, Robotics, and Autonomous Systems, 4(1): 215–242, 2021. (see page: 1)

[47] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696. IEEE, 2020. (see pages: 1, 2)

[48] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013. (see page: 1)

[49] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024. (see page: 2)

[50] 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. (see page: 5)

[51] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13019–13029, 2024. (see page: 2)

[52] 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. (see pages: 1, 2, and 6)

[53] Fabio Tosi, Youmin Zhang, Ziren Gong, Erik Sandström, Stefano Mattoccia, Martin R Oswald, and Matteo Poggi. How nerfs and 3d gaussian splatting are reshaping slam: a survey. *arXiv preprint arXiv:2402.13255*, 2024. (see page: 1)

[54] Jingwen Wang, Martin Rünz, and Lourdes Agapito. Dsp-slam: Object oriented slam with deep shape priors. In *2021 International Conference on 3D Vision (3DV)*, pages 1362–1371. IEEE, 2021. (see page: 1)

[55] Silvan Weder, Francis Engelmann, Johannes L Schönberger, Akihito Seki, Marc Pollefeys, and Martin R Oswald. Alster: A local spatio-temporal expert for online 3d semantic reconstruction. *arXiv preprint arXiv:2311.18068*, 2023. (see page: 3)

[56] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *Robotics: Science and Systems (RSS)*, 2024. (see pages: 2, 3, 5, 6, and 7)

[57] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: science and systems*, page 3. Rome, Italy, 2015. (see page: 1)

[58] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019. (see page: 1)

[59] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. (see pages: 5, 8)

[60] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. (see page: 6)

[61] Hongjia Zhai, Gan Huang, Qirui Hu, Guanglin Li, Hujun Bao, and Guofeng Zhang. Nis-slam: Neural implicit semantic rgb-d slam for 3d consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*, 2024. (see pages: 2, 3, and 6)

[62] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. (see pages: 2, 8)

[63] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022. (see page: 2)

[64] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. Sni-slam: Semantic neural implicit slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21167–21177, 2024. (see page: 1)