IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Open-Vocabulary Online Semantic Mapping for SLAM

Tomas Berriel Martins[1], Martin R. Oswald[2], and Javier Civera[1]

***Abstract*— This paper presents an Open-Vocabulary Online 3D semantic mapping pipeline, that we denote by its acronym OVO. Given a sequence of posed RGB-D frames, we detect and track 3D segments, which we describe using CLIP vectors. These are computed from the viewpoints where they are observed by a novel CLIP merging method. Notably, our OVO has a significantly lower computational and memory footprint than offline baselines, while also showing better segmentation metrics than offline and online ones. Along with superior segmentation performance, we also show experimental results of our mapping contributions integrated with two different full SLAM backbones (Gaussian-SLAM and ORB-SLAM2), being the first ones using a neural network to merge CLIP descriptors and demonstrating end-to-end open-vocabulary online 3D mapping with loop closure.**

I. Introduction

Semantic mapping targets the estimation of the category to which each element in a scene belongs, along with a consistent geometric representation. Rich semantic representations in 3D are essential for advanced robotic applications. Traditionally, semantic 3D reconstruction has relied on a closed-set approach in both offline [1, 2] and online [3, 4] settings, including integrations into semantic Simultaneous Localization and Mapping (SLAM) systems [5–7]. However, these methods are constrained by a predefined set of categories, which limits their flexibility and applicability in open-ended, real-world environments.

Following the emergence of Contrastive Language-Image Pre-training (CLIP) [8], there has been a surge of interest in open-vocabulary 3D representations [9–11], including efforts in online mapping [12–14]—though not yet in full SLAM systems. While these recent approaches have shown strong performance, their dependence on offline processing or ground-truth camera poses for mapping significantly limits their applicability in robotics, augmented reality, and virtual reality scenarios.

In this paper, we present OVO, an Open-Vocabulary Online mapping algorithm, which we integrate into two distinct visual SLAM pipelines. An example of our online reconstruction results is shown in Fig. 1. Our method processes RGB-D keyframes to generate 3D segments, each associated with a CLIP embedding. These segments are initialized by back-projecting masks predicted by Segment Anything Model (SAM) 2.1 [15], and are tracked over time by projecting them into 2D and matching against new masks. Each 3D segment's CLIP descriptor is selected from the keyframe views with the best visibility. Additionally, we introduce a novel model to extract per-instance CLIP descriptors directly from images, which are then assigned to the corresponding 3D masks. Our CLIP merging employs a neural network that learns per-dimension weighting to fuse CLIP descriptors of the same instance, while effectively generalizing to unseen classes and environments. Our pipeline not only operates online and supports loop-closure optimization, but also outperforms existing baselines in segmentation accuracy.

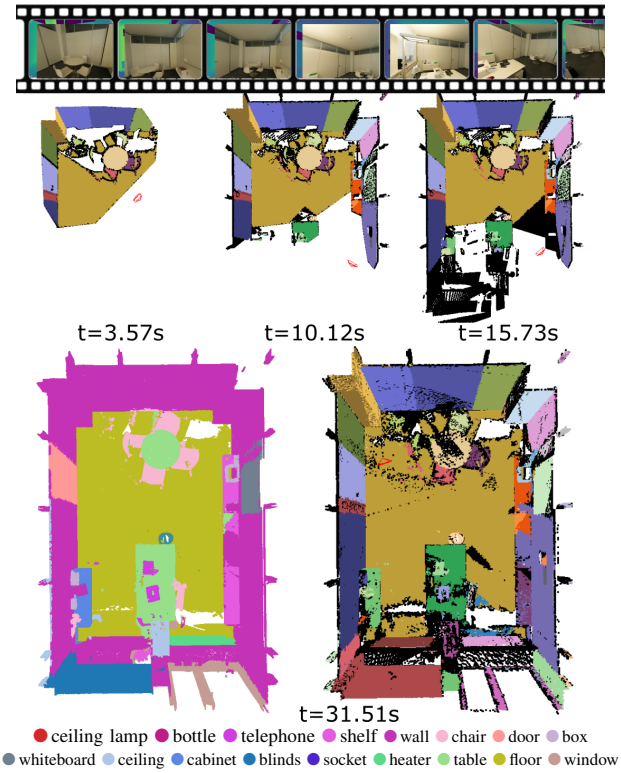

Fig. 1. **OVO mapping.** Given a RGB-D set of keyframes (**top**), our method successively reconstructs a 3D open-vocabulary representation of a scene over time (**middle**). At any moment, both semantic labels (**bottom left**) as well as instance labels (**bottom right**) can be effectively recovered.

II. Related Work

Our OVO estimates consistent 3D open-vocabulary semantics and seamlessly integrates with SLAM pipelines. Unlike previous methods that either use a closed set of categories, offline processing, 2D semantic representations or odometry. Tab. I provides a comparative summary of recent related

This work was supported in part by Spanish Government under grants PID2021-127685NB-I00 and PID2024-155886NB-I00, and in part by Aragón Government under grant T45_23R.

[1]Authors are with Instituto de Investigación en Ingeniería de Aragón (I3A), University of Zaragoza, Spain. {tberriel,jcivera}@unizar.es

[2]Author is with the Computer Vision Group, University of Amsterdam, Netherlands m.r.oswald@uva.nl

Digital Object Identifier (DOI): see top of this page.

works based on these aspects, with further details discussed in the remainder of this section.

Open-Vocabulary Image Semantics. The introduction of Contrastive Language-Image Pretraining (CLIP) [8], which encodes image and text tokens into a shared latent space, revolutionized semantic segmentation. By computing similarity to text inputs, CLIP enables classification into any category expressible in language. Several variations of CLIP have enhanced its performance [16, 17] and improved feature granularity, aiming to generate dense feature vectors [18, 19] rather than per-image representations. While closed-vocabulary methods outperform on predefined sets, open-vocabulary offers optimization-free generalization, highly relevant for diverse applications.

Offline 3D Open-Vocabulary from 3D point clouds. Most open-vocabulary 3D semantic approaches assume a known 3D point cloud. OpenScene [9] leverages OpenSeg [20] to compute CLIP features from images and trains a network to associate 2D pixels with 3D points. For each 3D point it performs average pooling on CLIP vectors from multiple views and supervises an encoder to directly assign CLIP features to 3D point clouds. OpenMask3D [10] selects k views per object, crops its 2D SAM mask to compute a CLIP features, and then features are average-pooled across crops and views. Open3DIS [11] integrates SuperPoint [21] with 2D instance segmentations and a 3D instance segmentator to generate multiple 3D instance proposals, describing each with CLIP features following OpenMask3D [10]. In contrast, OpenYolo-3D [22] uses a 2D open-vocabulary object detector instead of relying on 2D instance masks and CLIP features. It classifies each object based on the most common class across all views. While this approach eliminates the need for CLIP feature extraction, it limits each scene to a predefined set of classes.

Offline 3D Open-Vocabulary from RGB and RGB-D. OpenNeRF [23] optimizes a NeRF to encode the scene representation along with per-pixel CLIP features from OpenSeg. The OpenSeg features are projected into 3D to compute the mean and covariance of 3D points. The NeRF then renders novel views, prioritizing areas with high covariance to compute additional OpenSeg features and refine the model. Hierarchical Open-Vocabulary 3D Scene Graphs (HOV-SG) [24] relies on an offline hierarchical global fusion approach that requires precomputing 3D segments and features for all frames. These 3D segments and features are incrementally fused by merging observations across consecutive frames. The authors argue that relying solely on masked segments, as in Concept-Graphs [13], discards crucial contextual information. To address this, they propose a descriptor that merges in a handcrafted manner three CLIP embeddings per mask: (1) the full image, (2) the masked segment without background, and (3) the masked segment with background. We adopt this strategy, and contribute by proposing a novel approach to learn the CLIP merging operation.

Online Semantics. To date, online semantic methods have focused mostly on closed vocabularies. SemanticFusion [3] was one of the first semantic SLAM pipelines, predicting per-pixel closed-set categories and fusing predictions from different views in 3D space. Fusion++ [25] uses Mask-RCNN [26] to initialize per-object Truncated Signed Distance Functions (TSDFs), building a persistent object-graph representation. In contrast, PanopticFusion [27] 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. [28] and ALSTER [29], jointly reconstruct geometry and semantics in a SLAM framework. Additionally, NIS-SLAM [30] trains a multi-resolution tetrahedron NeRF to encode color, depth and semantics. NEDS-SLAM [31] 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 [32] and SGS-SLAM [7] augment a 3DGS SLAM with semantic ids of predefined set of classes. These approaches either assume known 2D ground-truth closed set of semantic classes (and therefore only tackle a multi-view fusion problem), or only represent 2D semantics, with limited capabilities for 3D segmentation or precise 3D object localization. More recently, OpenFusion [14] and Concept-Graphs [13] integrated open-vocabulary semantic descriptors into online 3D mapping pipelines. Concept-Graphs relies on the naive mask-cropping to compute CLIP descriptors, while OpenFusion uses SEEM [33] and creates a TSDF with 3D segments. None of them, however, addresses the integration into a full SLAM pipeline with loop closure optimization as we do.

TABLE I

OVERVIEW OF 3D SEMANTIC RECONSTRUCTION BASELINES.

Method	Open Vocabulary	3D semantics	Online	Loop Closure
OpenScene [9]	✓	✓	✗	-
OpenMask3D [10]	✓	✓	✗	-
Open3DIS [11]	✓	✓	✗	-
HOV-SG [24]	✓	✓	✗	-
OpenNeRF [23]	✓	✓	✗	-
NEDS-SLAM [31]	✗	✗	✓	✗
NIS-SLAM [30]	✗	✗	✓	✗
SGS-SLAM [7]	✗	✓	✓	✗
Kimera-VIO [4]	✗	✓	✓	✗
Concept-Fusion [12]	✓	✓	✓	✗
Concept-Graphs [13]	✓	✓	✓	✗
Open-Fusion [14]	✓	✓	✓	✗
OVO (ours)	✓	✓	✓	✓

III. OVO METHODOLOGY

OVO relies on a parallel-tracking-and-mapping architecture, as first defined by Klein and Murray [34] and adopted by most visual SLAM implementations [35]. Fig. 2 shows an overview of OVO. It takes as input a stream of RGB-D keyframes ($\{k_0, \ldots, k_n\}$ in the figure) and their respective poses and local point clouds. From this 3D representation, Sec. III-A, OVO extracts and tracks a set of 3D segments covering the whole representation (*3D segment mapper* in the figure, detailed in Sec. III-B). We compute a CLIP descriptor per each segment's viewpoint merging 3 different CLIPs (*CLIP merging* in the figure, detailed in Sec. III-E). Then

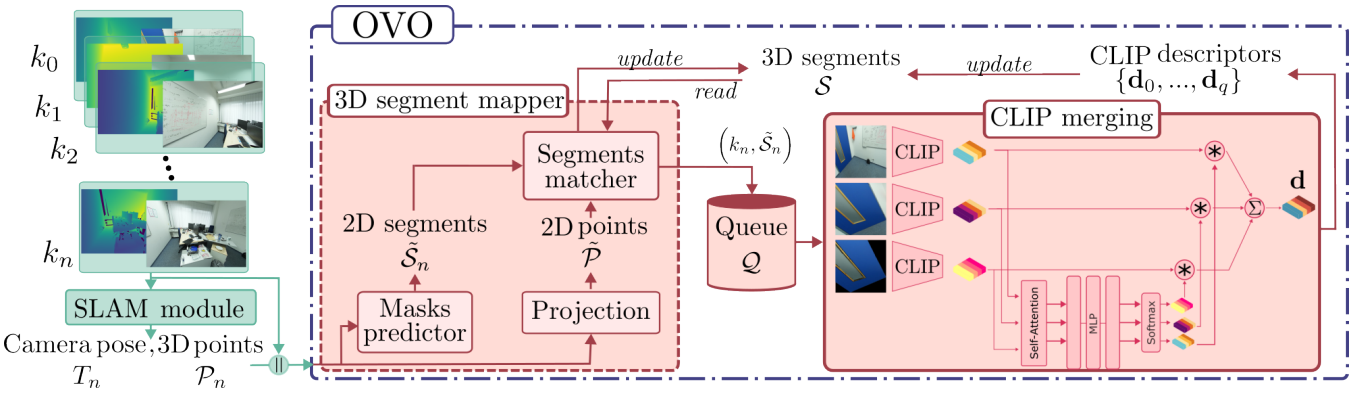

Fig. 2. **Overview.** From a stream of RGB-D keyframes, OVO builds, online, a 3D semantic representation of the scene. It relies on a 3D segment mapper to cluster 3D points into 3D segments; a queue to distribute the CLIP extraction computation, and a novel CLIP merging method to aggregate CLIP descriptors from multiple keyframes into one for each 3D segment.

assign to the 3D segment the most representative descriptor, Sec. III-D. When the SLAM module performs a loop closure or bundle-adjustment optimization, a routine searches for repeated 3D segments, and fuses those that were not correctly tracked, Sec. III-C.

A. Map Definition

Its input is an RGB-D video $\mathcal{V} = \{f_0, \dots, f_\tau\}$, $f_\tau \in \mathbb{N}^{w \times h \times 3}_{\leq 255} \times \mathbb{R}_{>0}$ representing the RGB-D frame of size $w \times h$ captured at time step τ. A SLAM front-end estimates in real-time the pose T_n of every frame f_τ in the world reference frame. The SLAM back-end selects a set of keyframes $\mathcal{K} = \{k_0, \dots k_n\} \subset \mathcal{V}$ from which it iteratively refines their poses $\mathcal{T} = \{T_0, \dots, T_n\}$, $T_n \in SE(3)$ asynchronously, at a rate lower than the video rate of the tracking thread.

Our scene representation or 'map' $\mathcal{M} = \{\mathcal{T}, \mathcal{P}, \mathcal{S}\}$, consists on these keyframe poses $\mathcal{T}$, a point cloud $\mathcal{P} = \{P_0, \dots, P_m\}$ and a set of 3D segments $\mathcal{S} = \{S_0, \dots, S_q\}$, being q the identifier of the last added segment. Every map point $P = \left(\begin{bmatrix} x & y & z \end{bmatrix}^\top, l_p\right)$ is defined by its 3D coordinates $\begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3$ and a discrete label $l_p \in \{-1, 0, 1, \dots, q\}$, $l_p > -1$ indicating the 3D segment the point belongs to, and $l_p = -1$ indicating that it is unassigned. The dense point cloud $\mathcal{P}$ is built concatenating at each keyframe k_n the estimated 3D points $\mathcal{P}_n$ provided by the SLAM front-end. If the SLAM front-end does not estimate a dense point cloud, $\mathcal{P}_n$ is computed as the unprojection of the input depth map to 3D using the estimated camera pose $T_n \in SE(3)$. To avoid $\mathcal{P}$ growing unconstrained, a pixel is not projected to 3D if a previously unoccluded 3D point falls inside its neighborhood when projected back to 2D. For every 3D point, occlusion is assessed by comparing its projected depth to its measured depth in the 2D pixel it is projected. Every 3D segment $S = (\mathbf{d}, \kappa)$ has a unique identifier, its semantics are described by a CLIP feature $\mathbf{d} \in \mathbb{R}^d$, and stores in a heap κ the indices of the best keyframes in which S was seen, ordered by visibility scores.

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($\mathcal{P}, T_n$)
  for (s, l_s) in $\tilde{\mathcal{S}}_n$ **do** ▷ For every 2D segment in k_n
    $mode, v \leftarrow$ get_label_mode_and_votes($\tilde{\mathcal{P}}_n, s, \epsilon$)
    if $v > \epsilon$ **then** ▷ #votes greater than threshold
      if $mode = -1$ **then**
        $S_{q+1} \leftarrow$ new_3D_segment($q+1, n, s$)
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_{q+1}\}$
        $l_s \leftarrow q+1$
      else
        $\mathcal{S} \leftarrow$ update_3D_segment(S_{mode}, n, s)
        $l_s \leftarrow z_l$
  $\tilde{\mathcal{S}}_n \leftarrow$ merge_and_prune_2D_segments($\tilde{\mathcal{S}}_n$)
  $\mathcal{P} \leftarrow$ update_pcd_labels($\mathcal{P}, \tilde{\mathcal{P}}_n, \tilde{\mathcal{S}}_n$)
  return $\mathcal{P}, \mathcal{S}, \tilde{\mathcal{S}}_n$

B. 3D Segment Mapper

For every new keyframe k_n, we run an image segmentation model that returns a set of 2D segments $\tilde{\mathcal{S}}_n = \{(s_0, l_{s0}), (s_1, l_{s1}), \dots\}$, each segment being composed of a mask s and a label l_s, which is initialized as $l_s := -1$. We then select the 3D map points in k_n's frustum, project them to k_n, and remove occluded points by comparing their projected depth to the input depth. In this manner, we obtain the 2D point set $\tilde{\mathcal{P}}_n = \{p_0, p_1, \dots\}$, for which $p = \left(\begin{bmatrix} u & v \end{bmatrix}^\top, l_p\right)$. We compute the label mode of all points p within a segment s, that we will represent slightly abusing notation as $z_l := \arg\max_{l_p} \left(\tilde{\mathcal{P}} \cap s\right)$. If the mode receives less votes v than a predefined threshold ϵ, we discard s. If not, two possibilities can occur:

1) If $z_l = -1$, we set $z_l := q+1$ and initialize a new 3D segment S_{q+1} with an empty CLIP feature $\mathbf{d}$ (filled later as described in Sec. III-D), and a keyframe heap $\kappa := \{(n, r)\}$, initialized with k_n's index and s' visibility score r.
2) Otherwise, 2D segment s is a match for 3D segment S_{z_l} and the keyframe will be inserted into κ, and stored if it is one of the best views or if κ is not full.

For both, the unassigned 3D points and 2D segment's labels, l_p and l_s are updated to the identifier of the matched S_{z_l}.

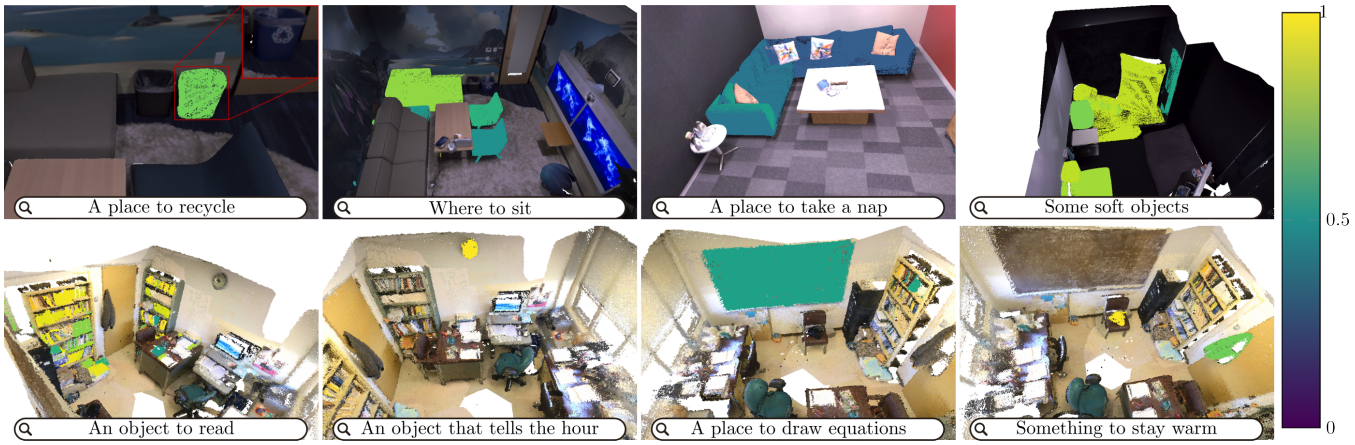

Fig. 3. **Out-of-distribution queries.** From left to right, top to bottom, observe how common-language queries allow to differentiate bins based on a recycling symbol; recongize sofas and chairs as places to sit; that you can take a nap in a sofa, pillows and couches are soft objects, and books are readable, that the clock tells the hour, the blackboard is to draw equations, and the jacket is something to stay warm. Colorbar shows similarity strength.

After matching all 2D masks, those that share the same l_s are merged. Finally, once all masks are gathered in $\tilde{\mathcal{S}}_n$, the tuple $\left(k_n, \tilde{\mathcal{S}}_n\right)$ is pushed to the queue $\mathcal{Q}$. Keyframes and masks remain in $\mathcal{Q}$ until processing resources become available to compute the CLIP descriptors for the highest-scoring 2D segments.

C. Loop Closure

When the SLAM module closes a loop or completes a Global Bundle Adjustment, OVO updates both its map and the set of 3D instances. We denote both after the update as $\mathcal{M}'$ and $\mathcal{S}'$. For each updated keyframe $T'_n \in \mathcal{T}'$, its associated local point cloud is also updated by propagating the pose correction as $\mathcal{P}_n := T'_n \, T_n^{-1} \, \mathcal{P}_n$. This transforms the points from the world frame to the original keyframe's and back using the updated pose T'_n. Keyframes that are removed during SLAM optimization are discarded along with their associated 3D points. After updating the 3D points, the temporary queue $\mathcal{Q}$ is cleared. Next, the set of 3D instances $\mathcal{S}'$ is pruned by removing instances whose associated points were entirely deleted during optimization. Following, instance fusion is performed by comparing remaining pairs of 3D instances. Two instances are merged if they satisfy the following criteria: **(1)** The distance between their point cloud centroids is < 150cm, **(2)** the cosine similarity between their CLIPs is > 0.8, and **(3)** more than 50% of their points lie within 10cm of a point in the other instance. For a pair of segments S_i and S_j to be merged, their point indices are unified as $\kappa_i := \kappa_i \cup \kappa_j$, and all map points previously labeled as j are reassigned to i, i.e., $\forall P_k \in \mathcal{P} | l_k = j, \implies l_k := i$.

D. CLIP Descriptors

When a tuple $\left(k_q, \tilde{\mathcal{S}}_q\right)$ is popped from $\mathcal{Q}$, only the matched 2D segments for which k_q is still in the κ of their 3D instance S are selected. A CLIP descriptor $\mathbf{d}$ is computed for each of them as explained in Sec. III-E. Then, the final descriptor for a 3D segment S is selected between the 2D segments in its keyframes' heap κ, as the CLIP descriptor with the smallest aggregated distance to the rest. To query the 3D semantic representation, text queries are encoded to CLIP space. Then, we compute the cosine similarity between the CLIP descriptor of the query and the descriptor $\mathbf{d}$ of each 3D segment in $\mathcal{S}$.

E. CLIP Merging

Similarly to HOV-SG [24], for each 2D segment we compute three CLIP descriptors: 1) $\mathbf{d}_0$ for the full keyframe, 2) $\mathbf{d}_1$ for the segment masking the rest of the image out, and 3) $\mathbf{d}_2$ for the minimum bounding box that contains the segment. In contrast, in our case, the CLIP descriptor $\mathbf{d} = \sum_{i=0}^{2} \mathbf{w}_i \odot \mathbf{d}_i$ of a 2D segment is the result of merging the three descriptors $\mathbf{d}_{i=\{0,1,2\}}$ using a per-dimension weighted average with weights $\mathbf{w}_i \in \mathbb{R}^d$ ($\odot$ is the Hadamard product). Our weights $\mathbf{w}_{i=\{0,1,2\}}$ are predicted by a neural model, as shown in Fig. 2. Note that HOV-SG's merging is done with hand-crafted scalar weights (*i.e.*, $\mathbf{d} = \sum_{i=0}^{2} w_i \mathbf{d}_i$, $w_i \in \mathbb{R}$).

As seen in Fig. 2, the input to our CLIP merging is three CLIPs $\mathbf{d}_{i=\{0,1,2\}}$. These are first passed by a transformer encoder, and the output is flattened and fed to a MLP, predicting the weights, and a softmax, forcing $\sum_{i=0}^{2} \mathbf{w}_i = \mathbf{1}^d$. Our CLIP merging is pre-trained following SigLIP [16]. For a mini-batch $\mathcal{B} = \{(s_0, c_0), (s_1, c_1), \dots\}$ composed by pairs of 2D segments s_j and semantic classes c_j, we minimize the sigmoid cosine similarity loss

$$L = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \log\left(\frac{1}{1+\exp(z_{ij}(-t\mathbf{d}_i \cdot \mathbf{y}_j + b))} \right) \quad (1)$$

between the merged CLIP descriptor $\mathbf{d}_i$, and the CLIP embedding $\mathbf{y}_j$ of the semantic class c_j associated to the 2D segment s_j in the same batch $\mathcal{B}$. z_{ij} is the label for a given image and class input, which equals 1 if they are paired and -1 otherwise. b and t are learnable bias and temperature parameters, used to compensate the imbalance coming from negative pairs dominating the loss.

IV. Experiments

First, we report OVO evaluation on 3D online semantic mapping on two established datasets, one synthetic (Replica),

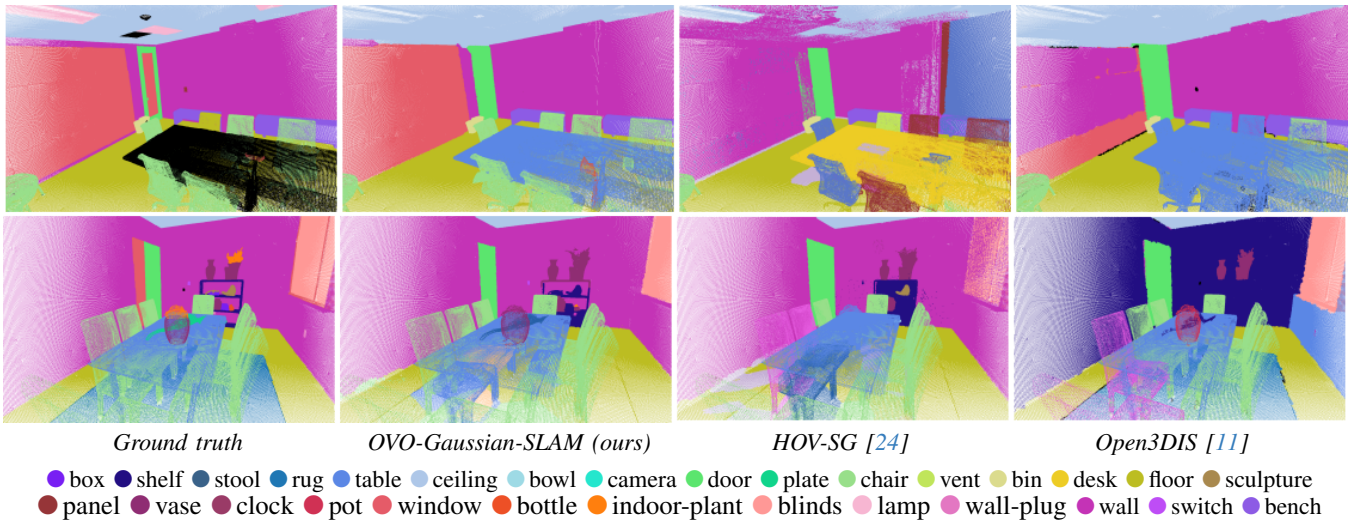

Fig. 4. **3D semantic segmentation on Replica.** OVO yields more accurate results in comparison to the two best offline baselines.

and one real (ScanNetv2). Then, we present our CLIP merging evaluation on semantic classification of images with ground-truth segmentation masks both on a dataset with multiple masks per image (ScanNet++) and with a single mask per image (ImageNet-S), and against alternative methods integrated into OVO for 3D semantic mapping (Replica).

Implementation. For OVO, we implemented three different configurations to show its flexibility: (1) **OVO-mapping**, that uses ground-truth camera poses, (2) **OVO-Gaussian-SLAM**, where we integrate our contributions within Gaussian-SLAM [36], a SLAM method targeting novel-view synthesis and dense point cloud reconstruction, although not real-time, and (3) **OVO-ORB-SLAM2** for which we integrate with ORB-SLAM2 [37], a real-time feature-based SLAM system with loop-closure. While OVO-Gaussian-SLAM uses the center of 3D Gaussians as the dense point cloud $\mathcal{P}$, for **OVO-ORB-SLAM2** we build a dense point cloud by registering the local point clouds from the RGB-D images. All three configurations use SAM2.1-l for 2D segmentation and SigLip ViT-SO400 for CLIP descriptors. **Our CLIP merging** has 5 self-attention layers with 8 heads, a 1152 latent dimension, with drop-out of 0.1, and 4 layers MLP with 3×1152 input/output neurons and $\times4$ inverse bottleneck with Leaky ReLU activations. It was trained with 4 Nvidia V100 GPUs, using Pytorch, with AdamW optimizer, learning rate 1×10^{-6}, gradient clipping at 1, and batch size of 512 per GPU, for 15 epochs using the top 100 semantic labels from ScanNet++ 250 training set. To compensate for class imbalance, in the loss we weight each element of the batch by the inverse of their class frequency in the training set.

Baselines. We evaluate CLIP merging against baselines that compute local CLIP descriptors [12, 13, 24] (using all of them SigLIP-SO400M) and Alpha-CLIP [19], a state-of-the-art model developed to condition CLIP using masks. Additionally, we include two variations of our CLIP merging trained in the same setup, in order to validate its design: directly predicting the fused descriptor, and predicting only one weight per descriptor. As detailed in Sec. II, existing semantic SLAM pipelines do not construct a 3D representation that can be evaluated using 3D metrics for open-set classes. Thus, we compare OVO against similar 3D open-vocabulary online mapping systems, Concept-Graphs [13], and OpenFusion [14]; and the state-of-the-art 3D open-vocabulary offline baselines OpenScene [9], OpenNeRF [23], Open3DIS [11] and HOV-SG [24]. Finally, we evaluate computational cost against Concept-Graphs, OpenFusion, HOV-SG and OpenNeRF, but exclude Open3DIS and OpenScene, as they rely on pre-processed 3D geometry and features.

Datasets. ScanNet++ [38] has 250 training and 50 validation indoor RGB+D scenes sequences. We use 2D rasterized masks for a total of 1.6M and 400k 2D instance samples respectively. Semantic classes are mapped into either the set of 100 most commons (used for training) or the full set of over 1.6k classes (used for evaluation). ImageNet-S [39] has a validation set of $\sim$12k images with 919 semantic labels. ScanNetv2 [40] has a full validation set of 312 RGB+D sequences of real scenes (FVS). We also evaluate on the 5-scene subset used by HOV-SG (HVS). We use the original annotation set with 20 classes (ScanNet20) and the expanded set with 200 classes (ScanNet200 [41]). On Replica [42], we use the standard 8-scene subset (*office-0...4*, *room-0...2*) and its 51 annotated classes.

Metrics. Semantic classification is evaluated using *mean Intersection Over Union* (mIoU) and *mean Accuracy* (mAcc) on ScanNet++, while on ImageNet-S we report the standard Top-1 and Top-5 mAcc. While we assess CLIP merging in 2D to isolate other factors, the full OVO is evaluated in 3D by labeling the vertices of ground-truth meshes and comparing them against ground-truth 3D labels. For Replica, following OpenNeRF [23], we report mIoU and mAcc, categorizing labels into tertiles based on their frequency (*head*, *common*, and *tail*). In ScanNetv2, we further present metrics weighted by the label frequency in the ground truth (f-mIoU and f-mAcc). Additionally, we analyze our computational

TABLE II

3D SEMANTIC SEGMENTATION EVALUATION ON REPLICA 51 CLASSES, SPLITTING BY FREQUENCY TERTILES: HEAD, COMMON AND TAIL.

Method	Online	Geo-metry	Camera pose / ATE RMSE [cm]	All		Head		Common		Tail	
				mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
OpenScene [9] (Distilled)	✗	GT	GT	14.8	23.0	30.2	41.1	12.8	21.3	1.4	6.7
OpenScene [9] (Ensemble)	✗	GT	GT	15.9	24.6	31.7	44.8	14.5	22.6	1.5	6.3
OpenNeRF [23]	✗	Est.	GT	20.4	31.7	35.4	46.2	20.1	31.3	5.8	17.6
HOV-SG [24]	✗	Est.	GT	22.5	34.2	35.9	44.2	23.6	42.3	8.0	16.1
Open3DIS [11] (SigLip)	✗	GT	GT	25.6	38.7	**49.7**	**64.4**	22.1	**42.4**	4.9	9.4
Concept-Graphs [13]	✓	Est.	GT	16.7	33.7	27.3	39.1	15.1	35.4	4.4	**26.8**
Open-Fusion [14]	✓	Est.	GT	20.5	34.8	37.9	51.7	14.0	30.3	9.8	22.2
OVO-mapping (ours)	✓	Est.	GT	27.0	**39.1**	45.0	59.9	**25.1**	38.5	11.0	18.8
OVO-Gaussian-SLAM (ours)	✓	Est.	**0.6**	**27.1**	38.6	44.1	58.0	25.0	39.0	**12.1**	18.9
OVO-ORB-SLAM2 (ours)	✓	Est.	1.9	25.6	39.0	43.0	59.1	21.6	38.3	**12.1**	19.6

TABLE III

3D SEMANTIC SEGMENTATION ON SCANNETV2 WITH FREQUENCY WEIGHTED METRICS ON 5 (HVS) AND ALL 312 VAL. SCENES (FVS).

	Method	Online	Geo-metry	Camera pose / ATE RMSE [cm]	ScanNet20				ScanNet200			
					mIoU	mAcc	f-mIoU	f-mAcc	mIoU	mAcc	f-mIoU	f-mAcc
HVS	Open3DIS [11] (SigLip)	✗	GT	GT	37.3	52.8	57.0	67.9	**17.8**	23.7	27.9	34.1
HVS	OpenScene(Ensemble) [9]	✗	GT	GT	**44.6**	**61.9**	**57.6**	**71.0**	9.4	12.6	27.8	32.0
HVS	HOV-SG [24]	✗	Est.	GT	34.4	51.1	47.3	61.8	11.2	18.7	27.7	37.6
HVS	Concept-Graphs [13]	✓	Est.	GT	17.1	29.1	26.0	33.1	6.0	11.7	21.4	27.7
HVS	Open-Fusion [14]	✓	Est.	GT	30.1	39.9	54.1	68.1	8.6	12.8	38.4	47.9
HVS	**OVO-mapping (ours)**	✓	Est.	GT	38.1	50.5	**57.6**	70.5	17.2	**25.3**	**45.4**	**56.4**
HVS	**OVO-Gaussian-SLAM (ours)**	✓	Est.	23.7	29.3	41.1	43.0	59.5	11.8	18.8	30.1	42.6
HVS	**OVO-ORB-SLAM2 (ours)**	✓	Est.	**21.5**	31.3	45.2	45.8	61.2	13.6	22.2	38.2	51.0
HVS	**OVO-ORB-SLAM2 w/o loop clos.**	✓	Est.	30.2	23.6	34.5	41.4	56.9	10.3	17.3	33.2	46.0
FVS	Open3DIS [11] (SigLip)	✗	GT	GT	24.7	40.9	32.5	45.3	9.4	17.0	22.9	32.2
FVS	OpenScene(Ensemble) [9]	✗	GT	GT	**47.0**	**70.3**	**57.7**	**69.8**	11.6	22.8	24.5	29.2
FVS	**OVO-mapping (ours)**	✓	Est.	GT	37.3	58.9	55.13	69.4	**17.4**	**35.9**	**44.3**	**57.8**

footprint. We measure wall-clock time required to optimize Replica scenes, as well as mean and max GPU vRAM and max system RAM usage (in GB). Each table highlights **first**, second, and third best results.

A. 3D Semantic Segmentation

Replica. Tab. II presents segmentation results for all our OVO configurations alongside relevant baselines. OVO outperforms all baselines in the aggregated mIoU and mAcc ('All' column). OVO-Gaussian-SLAM and OVO-ORB-SLAM2 surpass both offline and online mapping algorithms. This is particularly noteworthy since both implementations estimate camera poses and scene geometry, whereas all baselines (indicated in the table) rely either on ground-truth geometry, camera pose, or both. Thanks to the strong generalization of our CLIP merging, all OVO implementations have a significantly better mIoU on *tail* categories, which demonstrates less false-positives. As shown in Fig. 4, OVO effectively segments and classifies 3D instances, such as chairs and tables, that other baselines often misclassify due to the excessive context information incorporated into CLIP descriptors. OVO even outperforms the ground truth in some instances. For example, in "office4" (top left of Fig. 4), the ground-truth label for the table is missing, and one chair is misclassified as the floor. This underscores the advantage of open-set pipelines, particularly in situations where previous SLAM algorithms, which rely on known 2D semantics [7, 30], would fail.

ScanNetv2. Results, summarized in Tab. III, show how OVO-mapping matches HOV-SG, and even Open3DIS in the set ScanNet20. On the harder 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. OpenScene does achieve the best performance on ScanNet20. Nevertheless, its significant drop when using the extended set of classes highlights a weaker generalization capabilities than OVO and other baselines.

SLAM comparison. The difference between OVO's two SLAM versions and OVO-mapping is bigger in ScanNetv2 than in Replica (compare Tab. II and Tab. III), due to image blur and noisy depths in ScanNetv2. Gaussian-SLAM benefits from a more complex strategy for densification and pruning of the 3D point cloud, outperforming our simpler depth unprojection in Replica. However, while its camera tracking works flawlessly there, it does struggle in ScanNetv2 noisier images, where loop-closure plays a key role. Comparing OVO-ORB-SLAM2 w/o and w/ loop-closure, Tab. III, shows the importance of this feature. Further, Fig. 5 illustrates the loop closure correction over inconsistent reconstructions with repeated semantic instances, caused by odometric drift.

Computational footprint. Despite OpenFusion being 2× faster than OVO, thanks to using SEEM instead of

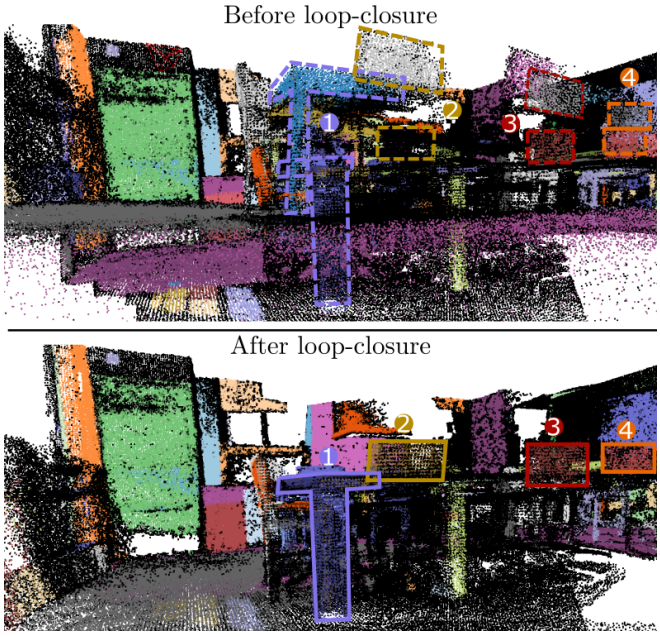

Fig. 5. **Visualization of OVO-ORB-SLAM2 loop closure on "scene0011_00" (ScanNet)**. We highlight four instances split due to tracking drift and effectively merged after loop-closure by our semantic fusion.

TABLE IV

RUNTIME STATISTICS ON REPLICA WITH 2K FRAMES PER SCENE.

Method	vRAM Avg / Max	RAM Max	Time Avg
HOV-SG [24]	6 / 12 GB	139 GB	∼11h
OpenNeRF [23]	4 / 22 GB	44 GB	∼20m
Open-Fusion [14]	**3 / 4 GB**	**6 GB**	**∼3m**
CG [13]	7 / 11 GB	16 GB	∼16m
OVO-mapping (ours)	4 / 8 GB	12 GB	∼6m

SAM+SigLIP, OVO achieves a better balance between speed and performance. It is still 2.5× faster than Concept-Graphs, 3× faster than OpenNerf and 80× faster than HOV-SG, as shown in Tab. IV. In contrast with HOV-SG, that relies on an expensive hierarchical merging of segments, requiring almost ×10 more RAM, OVO shows a lower RAM and GPU vRAM usage that enables its use on consumer devices. OVO-ORB-SLAM2 takes on average 0.67 seconds per keyframe on Replica and ScanNetv2, spiking up to 1.4 seconds for the slowest frame, and up to up to 2.5 and 6.1 seconds after loop closure on "scene0011_00" and "scene0231_00" respectively. This is compatible in our experiments with a conservative keyframe creation policy of 1 keyframe every 10 frames. Therefore, it is compatible with real-time SLAM pipelines, in which the critical camera tracking runs at video rate while the mapping runs at lower frequencies.

B. CLIP Merging

In Tab. V, we report evaluation on ImageNet-S, including also Alpha-CLIP, and on which both HOV-SG's and Concept-Fusion's merging approaches are equivalent to just computing the global descriptor due to there being only one mask per image. Tab. VI presents 2D semantic classification results on unseen scenes from ScanNet++, using the expanded label set of 1.6k, of our CLIP merging vs. HOV-SG's and Concept Fusion's (CF) CLIP merging, and the simpler mask crop used by Concept-Graphs.

TABLE V

IMAGENET-S SEMANTIC CLASSIFICATION ACCURACY.

Method	Top-1 mAcc	Top-5 mAcc
Alpha-Clip (ViT-L/14@336)[19]	77.6	94.1
SigLIP-SO400M[16]	82.5	95.7
Mask crop	80.3	93.4
Our CLIP merging	**84.8**	**96.6**

TABLE VI

2D OPEN VOCABULARY SEMANTIC CLASSIFICATION ON SCANNET++.

Method	mIoU	mAcc	f-mIoU	f-mAcc
Concept-Fusion[12]	8.6	**17.0**	10.0	12.8
Mask crop	8.5	**17.0**	10.0	12.8
HOV-SG merging[24]	9.4	15.9	12.8	15.9
Our CLIP merging	**9.5**	14.3	36.9	49.4
CLIP merging variations				
- fused	6.2	11.6	**40.0**	**56.7**
- per-descriptor	9.0	15.6	12.7	15.9

TABLE VII

3D OPEN-VOCABULARY SEMANTIC METRICS ON REPLICA OF OVO-MAPPING WITH ALTERNATIVE CLIP MERGING.

	All		Seen		Unseen	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
w/ HOV-SG's fusion	20.3	38.1	22.3	45.1	**18.3**	**30.9**
w/ our CLIP merging	**27.0**	**39.1**	**36.7**	**54.7**	16.9	22.8
w/ CLIP merging variations						
- fused	20.2	33.2	32.2	52.5	0.8	1.2
- per-descriptor	20.2	38.2	26.6	48.6	13.6	27.4

Overall, ours outperforms baselines, particularly in frequency-weighted metrics, although with slightly worse mAcc in ScanNet++. Alpha-CLIP performs worse than simpler approaches like our CLIP merging. Using a better backbone (SigLIP-SO400M vs ViT-L/14) outperforms a significantly more expensive fine-tuning (our trained two day on 4 V100 vs their on 128 A100 GPUs).

Regarding alternatives, the per-descriptor weights predictor achieves a similar performance to HOV-SG, while directly predicting a fused descriptor achieves better frequency-weighted metrics but significantly worse overall ones, which indicates overfitting. This is further validated evaluating OVO-mapping in Replica, Tab. VII using HOV-SG's merging approach, and the alternatives to our CLIP merging. The fused predictor performance collapses in classes not seen during training, while our proposed CLIP merging has a slightly worse performance than HOV-SG's, while being significantly better on the known classes. The per-CLIP weights is unable to match the performance, highlighting the impact of per-dimension weights.

Finally, we highlight in Fig. 3 how our CLIP merging preserves their rich semantic encoding, allowing our merged CLIPs to generalize to zero-shot complex language queries. For instance, our descriptors distinguish between two trash bins based on a recycling symbol on one of them, despite both being labeled just as *bin* in the ground truth.

C. Limitations

Despite OVO state-of-the-art results on 3D indoor semantic segmentation, generalization to outdoor large-scale scenes may face challenges such as different class distributions, illumination and blur, and higher tracking errors. Our semantic fusion at loop closure effectively corrects odometric drift. However, it sometimes misses instances that should be fused, something that may be fixed by a richer and more accurately localized set of features. We also observed in CLIP merging a slight bias towards classes seen at training, which may be solved with larger training sets.

V. Conclusions

In this paper, we present OVO, an open-vocabulary, online 3D mapping method. Our pipeline extracts 3D segments from 2D masks and tracks them across keyframes. To assign CLIP descriptors to 3D segments, we introduce a novel strategy: each 2D segment receives a single descriptor computed as a weighted sum of embeddings from the full image, the masked region, and its surrounding bounding box. The weights are predicted by a neural network, which outperforms handcrafted heuristics while retaining strong generalization. We also develop a mechanism to fuse instances that are affected by odometric drift after the geometric corrections of a loop closure. OVO outperforms existing baselines in both computational efficiency and segmentation quality across multiple datasets. By bridging SLAM with open-vocabulary representations, we believe that our work broadens the scope of applications in these two domains.

References

[1] S. Y. Bao *et al.*, "Semantic structure from motion," *CVPR*, 2011.

[2] A. Kundu *et al.*, "Panoptic neural fields: A semantic object-aware neural scene representation," *CVPR*, 2022.

[3] J. McCormac *et al.*, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," *ICRA*, 2017.

[4] A. Rosinol *et al.*, "Kimera: an open-source library for real-time metric-semantic localization and mapping," *ICRA*, 2020.

[5] J. Civera *et al.*, "Towards semantic SLAM using a monocular camera," *IROS*, 2011.

[6] S. Zhu *et al.*, "Sni-slam: Semantic neural implicit slam," *CVPR*, 2024.

[7] M. Li *et al.*, "Sgs-slam: Semantic gaussian splatting for neural dense slam," *ECCV*, 2024.

[8] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," *ICML*, 2021.

[9] S. Peng *et al.*, "Openscene: 3d scene understanding with open vocabularies," *CVPR*, 2023.

[10] A. Takmaz *et al.*, "OpenMask3D: Open-Vocabulary 3D Instance Segmentation," *NeurIPS*, 2023.

[11] P. Nguyen *et al.*, "Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance," *CVPR*, 2024.

[12] K. Jatavallabhula *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *RSS*, 2023.

[13] Q. Gu *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *ICRA*, 2024.

[14] K. Yamazaki *et al.*, "Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation," *ICRA*, 2024.

[15] N. R. et al., "Sam 2: Segment anything in images and videos," *arXiv:2408.00714*, 2024.

[16] X. Zhai *et al.*, "Sigmoid loss for language image pre-training," *ICCV*, 2023.

[17] M. Cherti *et al.*, "Reproducible scaling laws for contrastive language-image learning," *CVPR*, 2023.

[18] C. Zhou *et al.*, "Extract free dense labels from clip," *ECCV*, 2022.

[19] Z. Sun *et al.*, "Alpha-clip: A clip model focusing on wherever you want," *CVPR*, 2024.

[20] G. Ghiasi *et al.*, "Scaling open-vocabulary image segmentation with image-level labels," *ECCV*, 2022.

[21] D. DeTone *et al.*, "SuperPoint: Self-Supervised Interest Point Detection and Description," *CVPRW*, 2018.

[22] M. E. A. Boudjoghra *et al.*, "Open-YOLO 3D: Towards Fast and Accurate Open-Vocabulary 3D Instance Segmentation," *ICLR*, 2025.

[23] F. Engelmann *et al.*, "OpenNeRF: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views," *ICLR*, 2024.

[24] A. Werby *et al.*, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," *RSS*, 2024.

[25] J. McCormac *et al.*, "Fusion++: Volumetric object-level slam," *3DV*, 2018.

[26] K. He *et al.*, "Mask R-CNN," *ICCV*, 2017.

[27] G. Narita *et al.*, "Panopticfusion: Online volumetric semantic mapping at the level of stuff and things," *IROS*, 2019.

[28] D. Menini *et al.*, "A real-time online learning framework for joint 3d reconstruction and semantic segmentation of indoor scenes," *RAL*, 2021.

[29] S. Weder *et al.*, "Alster: A local spatio-temporal expert for online 3d semantic reconstruction," *WACV*, 2025.

[30] H. Zhai *et al.*, "Nis-slam: Neural implicit semantic rgb-d slam for 3d consistent scene understanding," *TVCG*, 2024.

[31] Y. Ji *et al.*, "Neds-slam: A neural explicit dense semantic slam framework using 3d gaussian splatting," *RAL*, 2024.

[32] B. Li *et al.*, "Hier-slam: Scaling-up semantics in slam with a hierarchically categorical gaussian splatting," *ICRA*, 2025.

[33] X. Zou *et al.*, "Segment everything everywhere all at once," *NeurIPS*, 2023.

[34] G. Klein *et al.*, "Parallel tracking and mapping for small ar workspaces," *ISMAR*, 2007.

[35] C. Campos *et al.*, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam," *TRO*, 2021.

[36] V. Yugay *et al.*, "Gaussian-slam: Photo-realistic dense slam with gaussian splatting," *arXiv:2312.10070*, 2023.

[37] R. Mur-Artal *et al.*, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *TRO*, 2017.

[38] C. Yeshwanth *et al.*, "Scannet++: A high-fidelity dataset of 3d indoor scenes," *ICCV*, 2023.

[39] S. Gao *et al.*, "Large-scale unsupervised semantic segmentation," *TPAMI*, 2022.

[40] A. Dai *et al.*, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," *CVPR*, 2017.

[41] D. Rozenberszki *et al.*, "Language-grounded indoor 3d semantic segmentation in the wild," *ECCV*, 2022.

[42] J. S. et al., "The Replica dataset: A digital replica of indoor spaces," *arXiv:1906.05797*, 2019.