

Open-Vocabulary Online Semantic Mapping for SLAM

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 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 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 them. Along with superior segmentation performance, we also show experimental results of our mapping contributions integrated with two different SLAM backbones (Gaussian-SLAM [65] and ORB-SLAM2 [37]), 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 onboard sensors [4]. While early SLAM research primarily targeted robotics, where it is seen as a fundamental step for autonomy [13], its widespread industrial adoption stemmed from augmented and virtual reality [27]. Today, its applications continue to expand into other domains [14, 36]. Visual SLAM research, however, has mainly focused on geometric models, sensor fusion, processing pipelines and optimizations [5, 39, 45, 62], and much less and mostly recently on the crucial aspect of the scene representation [49, 50, 58], that would further expand its potential for a wider array of tasks.

Online semantic representations have taken various forms, *e.g.*, object annotations in 3D point clouds [10, 17, 63], objects as high-level features [3, 41, 52, 59], semantic segmentations of point cloud maps [7, 32, 50] or implicit 3D representations [29, 31, 71]. All of them, however, are constrained to a predefined closed set of categories, limiting their applicability in real-world scenarios.

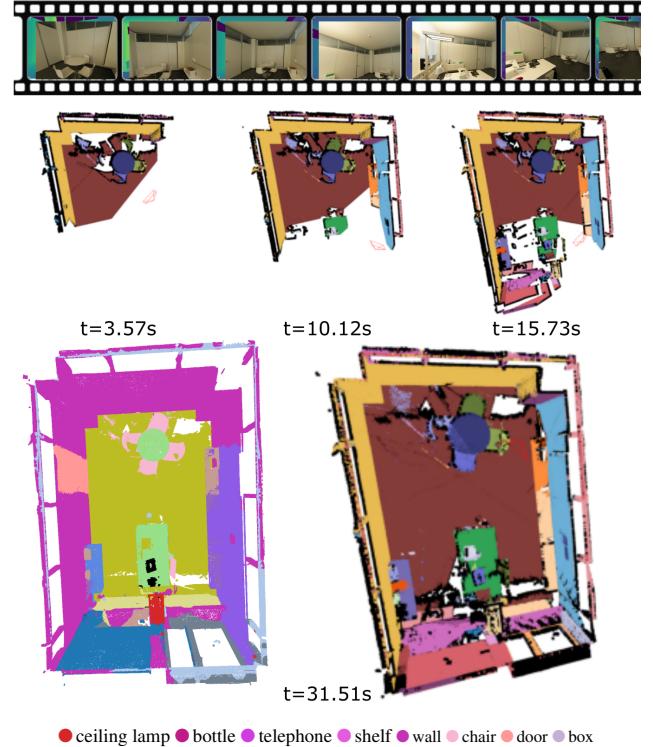


Figure 1. **OVO mapping.** Given an RGB-D input sequence (**Top**), our method successively reconstructs a scene open-vocabulary semantics over time (**Middle**: shows reconstructions at 3 different time instants). At the end of the sequence the full map is recovered (**Bottom**: the left depicts open-vocabulary representation after mapping to the target vocabulary color-coded by the provided legend; the right depicts 3D segments colored by instances).

Offline semantic 3D reconstructions have traditionally used a closed-set approach [1, 28, 29]. However, following the development of CLIP [47], research on open-vocabulary 3D representations has surged [2, 24, 40, 43, 44, 57]. Despite the excellent performance of these recent advancements, their reliance on offline processing limits their applicability in robotics, augmented reality, and virtual reality.

In this paper we present the first *Open-Vocabulary*

Online mapping algorithm, OVO, which we also integrate into different visual SLAM pipelines. Our novel method estimates, from a RGB-D sequence, a set of 3D segments that are assigned a CLIP vector per segment. Specifically, our segments are initialized by back-projecting SAM 2.1 [48] masks, and are tracked by projecting and matching the 3D segments against the 2D masks. The CLIP descriptor of each 3D segment is selected between the descriptors from its keyframes with better visibility. Furthermore, 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 motivated earlier, OVO is the first pipeline for open-vocabulary online 3D semantic mapping. Unlike prior methods that rely on offline optimization with ground-truth camera poses or scene geometry, our approach is compatible with simultaneous localization and mapping pipelines. 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 classified pixels into a fixed set of categories [20, 25]. Seminal work on Contrastive Language Image Pretraining (CLIP) [47], 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, 18, 21, 67] and the granularity of its features attempting to generate dense feature vectors [19, 56, 70] rather than per-image ones. Despite still far from matching the metrics of closed-vocabulary methods on predefined sets of labels, open-vocabularies allows for an optimization-free generalization that is highly relevant.

Offline 2D Open-Vocabulary from RGB. With the raise in popularity of Neural Radiance Fields (NeRFs) [35] and 3D Gaussian Splatting (3DGS) [23], semantics have been increasingly integrated into these representations. LERF [24] embeds per-object multi-scale CLIP features within NeRF representations, enabling 2D image searches using language queries. LangSplat [44] uses the Segment Anything Model (SAM) [26] 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 [23] representation is augmented with the reduced features to render novel viewpoints and query semantic labels on the decoded 2D rasterizations. SAGA [6] builds on LangSplat to compute CLIP descriptors and incorporates affinity fea-

Method	Open Vocabulary	Online	3D semantics
LERF [24]	✓	✗	✗
LangSplat [44]	✓	✗	✗
OpenScene [43]	✓	✗	✓
OpenMask3D [57]	✓	✗	✓
Open3DIS [40]	✓	✗	✓
HOV-SG [61]	✓	✗	✓
OpenNeRF [15]	✓	✗	✓
NEDS-SLAM [22]	✗	✓	✗
NIS-SLAM [66]	✗	✓	✗
SGS-SLAM [31]	✗	✓	✓
Kimera-VIO [50]	✗	✓	✓
OVO (ours)	✓	✓	✓

Table 1. **Overview of open-vocabulary 3D reconstruction baselines.** OVO is the first method that estimates 3D open-vocabulary semantics in an online manner compatible with online estimation of camera poses and scene geometry within a SLAM setting. In contrast, most other works either use a closed set of categories, offline processing, or 2D representations for the semantics.

tures optimized with a multi-view mask graph to cluster 3D targets. However, it has only been validated on 2D tasks. Instead, LEGaussians [54] and Language-Driven Physics-Based Scene Synthesis and Editing via Feature Splatting [46] integrate 3DGS with both CLIP and DINOv2 [42] 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 Open-Vocabulary from 3D point clouds. Most approaches to open-vocabulary 3D semantics assume a known 3D point cloud. OpenScene [43] leverages OpenSeg [19] 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 [57] selects k views per object, crops its 2D SAM mask (similarly to LERF [24]) and computes a CLIP vector per mask. CLIP features are then average-pooled over each crop and view. Open3DIS [40] 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 [57]. 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.

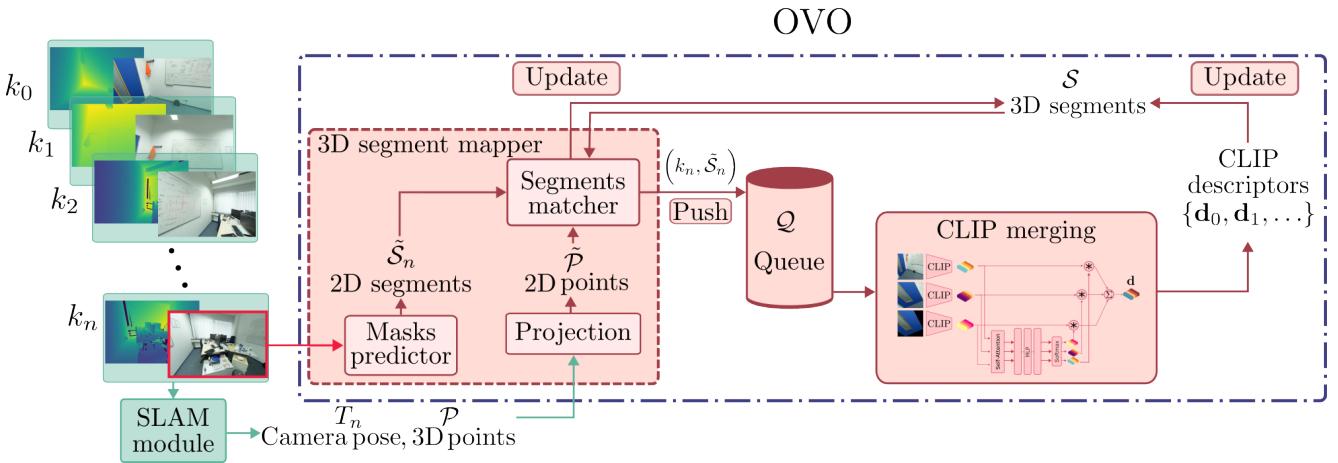


Figure 2. **OVO overview.** From a stream of RGB-D keyframes, OVO builds, online, a 3D semantic representation of the scene. It relies on a 3D segment mapper to cluster 3D points into 3D segments; a queue to distribute the CLIP extraction computation, and a novel CLIP merging method to aggregate CLIP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

Offline 3D Open-Vocabulary 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 NeRF then renders novel views focusing on areas with high covariance to compute additional OpenSeg features and refine the model. Highlight that HOV-SG incremental hierarchical fusion, requires to precompute the 3D segments and features of all the frames. Therefore, unlike OVO, it is offline and not SLAM-compatible. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation (HOV-SG) [61] relies on a hierarchical global fusion. First the full point-cloud of the scene is precomputed from RGB+D using DBSCAN [16] to filter noise. Then, SAM and CLIP are used to compute local 2D segments with corresponding CLIP vectors, which are projected to a 3D global map. Then, the 3D segments and features are fused hierarchically merging segments observed from consecutive frames. Furthermore, they argue that relying solely on masked segments, as in LangSplat [44], lacks critical contextual information. Hence, they introduced a novel approach to compute a descriptor merging three CLIP descriptors for each mask: (1) of the full image, (2) of the masked segment without background, and (3) of the masked segment with background.

Closed-Vocabulary Online 3D Semantics. To date, online semantic methods have focused exclusively on closed vocabularies. SemanticFusion [32] was one of the first semantic SLAM pipelines, predicting per-pixel closed-set categories and fusing predictions from different views in 3D space. Fusion++ [33] uses Mask-RCNN [20] to initial-

ize per-object Truncated Signed Distance Functions (TSDFs), building a persistent object-graph representation. In contrast, PanopticFusion [38] 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. [34] and ALSTER [60], jointly reconstruct geometry and semantics in a SLAM framework. Additionally, NIS-SLAM [66] trains a multi-resolution tetrahedron NeRF to encode color, depth and semantics. NEDS-SLAM [22] 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 [30] and SGS-SLAM [31] 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.

3. OVO

Figure 2 shows an overview of OVO. Our approach takes as input a set of RGB-D keyframes ($\{k_0, \dots, k_n\}$ in the figure) and their respective poses. Keyframe-based SLAM pipelines are common in the literature and our OVO in principle can be integrated with them, as we will show in the experimental results. From this 3D representation, OVO extracts and tracks first a set of 3D segments covering the whole representation (*3D segment mapper* in the figure, detailed in Section 3.2). We then assign a CLIP descriptor per 3D segment, that comes from merging of CLIPs extracted from the closest keyframes to this particular segment (*CLIP merging* in the figure, detailed in Section 3.4).

Algorithm 1 3D Segment Mapper

```

1: function 3D_SEGMENT_MAPPER( $\mathcal{P}, \mathcal{S}, k_n, T_n$ )
2:    $\tilde{\mathcal{S}}_n \leftarrow \text{segment\_keyframe}(k_n)$ 
3:    $\tilde{\mathcal{P}}_n \leftarrow \text{project\_point\_cloud}(\mathcal{P}, T_n)$ 
4:    $M = \{\}$ 
5:   for  $s$  in  $\tilde{\mathcal{S}}_n$  do            $\triangleright$  For every 2D segment in  $k_n$ 
6:      $m, v \leftarrow \text{get\_label\_mode\_and\_votes}(\tilde{\mathcal{P}}_n, s)$ 
7:      $M = M \cup \{m\}$ 
8:     if  $v > \epsilon$  then       $\triangleright$  #votes greater than threshold
9:       if  $m = -1$  then
10:         $S_{i+1} \leftarrow \text{new\_3D\_segment}(i + 1, n, s)$ 
11:         $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_{i+1}\}$ 
12:      else
13:         $S \leftarrow \text{update\_3D\_segment}(S_m, n, s)$ 
14:      end if
15:    end if
16:   end for
17:    $\tilde{\mathcal{S}}_n \leftarrow \text{merge\_2D\_segments}(\tilde{\mathcal{S}}_n, M)$ 
18:    $\mathcal{P} \leftarrow \text{update\_point\_cloud\_labels}(\mathcal{P})$ 
19:   return  $\mathcal{P}, \mathcal{S}, \tilde{\mathcal{S}}_n$ 
20: end function

```

3.1. Map Definition

OVO assumes a parallel-tracking-and-mapping architecture, as first defined by Klein and Murray [27] and adopted by most visual SLAM implementations [5]. Its input is a RGB-D video $\mathcal{V} = \{f_0, \dots, f_\tau\}$, $f_\tau \in \mathbb{N}_{\leq 255}^{w \times h \times 3} \times \mathbb{R}_{>0}$ standing for the RGB-D frame of size $w \times h$ captured at time step τ . A SLAM front-end estimates in real-time the pose 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 the keyframe poses \mathcal{T} , a point cloud $\mathcal{P} = \{P_0, \dots, P_m\}$ and a set of 3D segments $\mathcal{S} = \{S_0, \dots, S_q\}$ ¹. Every map point $P = ([x \ y \ z]^\top, l)$ is defined by its 3D coordinates $[x \ y \ z]^\top \in \mathbb{R}^3$ 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 $\tilde{\mathcal{S}}_n =$

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

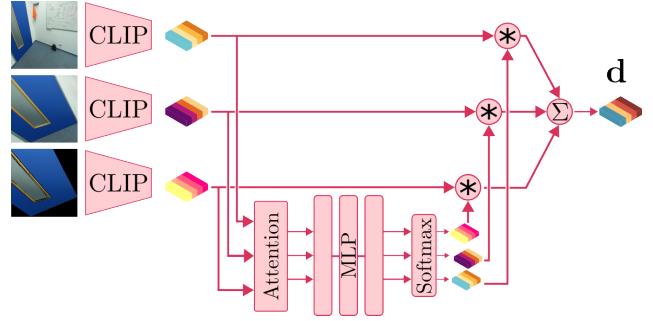


Figure 3. **CLIP merging.** The model predicts a weight for each dimension of each of the three input CLIP descriptors. The final descriptor \mathbf{d} is computed as a weighted sum of the input vectors.

$\{s_0, s_1, \dots\}$. 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 $p \in \tilde{\mathcal{P}}_n = ([x \ y]^\top, l)$. 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 predefined threshold ϵ , we discard the mask s . If not, two possibilities can happen:

1. If the 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 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 were matched to the same 3D segment S are merged.

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

3.3. CLIP Descriptors

When a tuple $(k_q, \tilde{\mathcal{S}}_q)$ 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 by merging three different descriptors. Then, the final descriptor for a 3D segment S is selected between the 2D segments in the 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} .

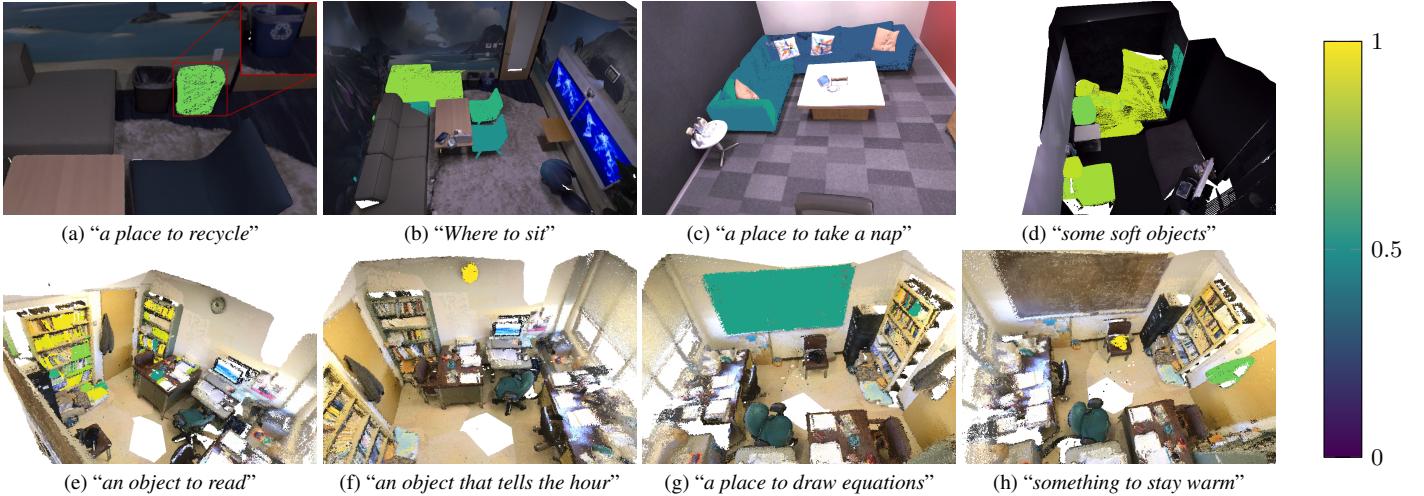


Figure 4. **Out-of-distribution queries.** From left to right, top to bottom, observe how the descriptors similarity with common-language queries allow to differentiate between two bins based on a recycling symbol; that both a sofa and a chair are places to sit; match a sofa to a place to take a nap, some pillows and a couch as soft objects, books as objects to read, a clock to something one should look to tell the hour, a blackboard as a place to draw equations, and a jacket as to something to stay warm. Colorbar shows similarity strength.

3.4. CLIP Merging

Similarly to HOV-SG [61], for each 2D segment we compute three CLIP descriptors: 1) one for the full keyframe, 2) one for the segment masking the rest of the image out, and 3) one for the minimum bounding box that contains the segment. The CLIP descriptor \mathbf{d} of a 2D segment is the result of fusing the three of them using a weighted average.

HOV-SG relies on a pre-tuned set of weights, using the same weight for all dimensions of the descriptors, despite they may encode fairly different concepts of different relevance. Instead, we propose to pre-train a model to predict per-dimension weights for each descriptor, and compute a per-dimension weighted average. Fig. 3 illustrates our novel Weights Predictor architecture. Specifically, the model takes as input the three CLIP descriptors. These are first passed by a transformer encoder, and the output is flattened and fed to an MLP that predicts per-dimension weights for each of the three CLIPs in the input, and a softmax, forcing that the three weights of each dimension sum one. Finally, the per-dimensional weighted average is computed multiplying the input CLIPs by the predicted weights and adding together corresponding dimensions.

Our Weights Predictor is pre-trained following SigLIP [67]. For a mini-batch $\mathcal{B} = \{(s_0, T_0), (s_1, T_1), \dots\}$ of (2D segments, semantic classes) pairs, we minimize the sigmoid cosine similarity loss function

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

between the merged CLIP descriptor \mathbf{d} , and the CLIP em-

bedding of the semantic class associated to its 2D mask, \mathbf{y}_j . Where z_{ij} is the label for a given image and class input, which equals 1 if they are paired and -1 otherwise, and b and t are the learnable bias and temperature parameters used to compensate the imbalance coming from negative pairs dominating the loss.

4. Experiments

Experimental setup. We implemented three different configurations for OVO. 1) **OVO-mapping**, for which we use ground-truth camera poses and unproject the RGB-D values to generate the 3D point cloud, 2) **OVO-Gaussian-SLAM**, for which we integrate our contributions within Gaussian-SLAM [65] pipeline, and 3) **OVO-ORB-SLAM2** for which we use the tracking and mapping implementation of ORB-SLAM2 [37]. For the three configurations, we relied on SAM2.1-1 for 2D instance segmentation, and SigLip ViT-SO400 for CLIP descriptors, except when stated otherwise. For further details of Weights Predictor and OVO see Appendix A.2 of the supplementary material.

Baselines. As detailed in Section 2, existing pipelines for semantic SLAM pipelines do not build a 3D representation that can be benchmarked using 3D metrics on open-set classes. They either are based on 2D semantic representations [22, 66], or optimize assuming known 2D semantic labels on a closed-vocabulary [31, 66]. Hence, we compare OVO against the offline open-vocabulary baselines OpenScene [43], OpenNeRF [15], Open3DIS [40] and HOV-SG [61]. We compare computational cost with HOV-SG and OpenNeRF, but not to Open3DIS and OpenScene which rely on pre-processed 3D geometry and features.

Method	All		Head		Common		Tail	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
OpenScene [43] (Distilled)†‡	14.8	23.0	30.2	41.1	12.8	21.3	1.4	6.7
OpenScene [43] (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 [61]†	22.5	34.2	35.9	44.2	23.6	42.3	8.0	16.1
Open3DIS [40] (SigLip)†‡	25.6	38.7	49.7	64.4	22.1	42.4	4.9	9.4
OVO-mapping (ours) (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-Gaussian-SLAM (ours)	27.1	38.6	44.1	58.0	25.0	39.0	12.1	18.9
OVO-ORB-SLAM2 (ours)	26.8	38.4	44.0	58.7	24.7	38.0	11.8	18.7

Table 2. **Evaluation on Replica.** OVO gives competitive results and on average outperforms all baselines. †Offline. ‡Known 3D geometry.

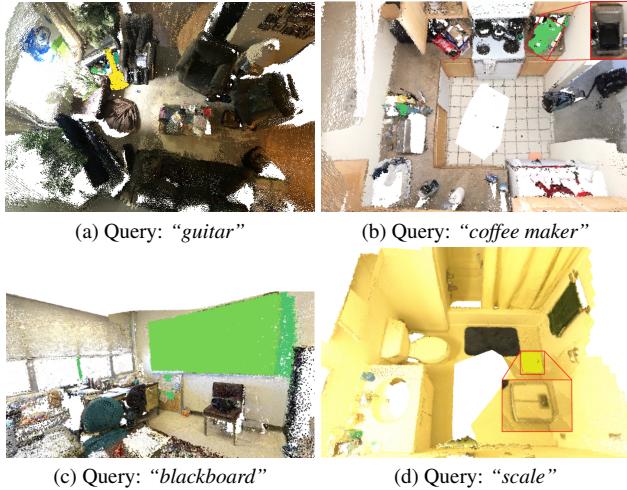


Figure 5. **Out-Of-Distribution queries.** Highlighted 3D points with mid and high similarity.

Datasets. We use ScanNet++ top 100 semantic labels to train the Weights Predictor, evaluating on the full set of more than 1.6K semantic classes. We also evaluate OVO open-vocabulary 3D semantic segmentation on ScanNetv2 [11] and Replica [55]. In ScanNetv2 we use both the original set of annotations with 20 classes (ScanNet20), and the expanded set with 200 classes (ScanNet200) [51]. We evaluate on the whole validation set of 312 scenes (FVS), and on the 5 scenes subset used by HOV-SG (HVS). For Replica we use the common 8 scenes subset (*office-0...4, room-0...2*), evaluating using the 51 classes on it. For further datasets information see Appendix A.1.

Metrics. Semantic segmentation is typically benchmarked using *mean Intersection Over Union* (mIoU) and *mean Accuracy* (mAcc). While we evaluate our Weights Predictor in 2D to isolate other effects, the full OVO is evaluated in 3D, by labeling the vertices of ground-truth meshes and comparing them against the ground-truth 3D labels. On Replica,

following OpenNerf [15], we also report mIoU and mAcc splitting the labels in tertiles based on their frequency (*head, common, tail*). In ScanNetv2 and ScanNet++ we additionally show the metrics weighted by the frequency of the labels in the ground-truth (f-mIoU and f-mAcc). Finally, we show results for the computational footprints of our implementations. We measure clock wall time required to optimize Replica scenes’, mean and max GPU vRAM and max computer RAM used in GigaBytes (GB). For OVO, we also report the final representation size in MegaBytes (MB) and the time to process each keyframe in seconds (KFPS). On each table we highlight first, second, and third best.

4.1. CLIP Merging

Table 3 shows segmentation results for our Weights Predictor and HOV-SG’s merging approach on novel scenes from ScanNet++ with an expanded set of labels. We outperform HOV-SG’s merging, by a significant margin in frequency-weighted metrics.

Method	mIoU	mAcc	f-mIoU	f-mAcc
HOV-SG	9.4	15.9	12.8	15.9
Weights Predictor	10.7	16.9	36.1	45.3

Table 3. Segmentation results of our Weights Predictor vs. HOV-SG’s merging on ScanNet++, using 1.6k queries.

Notably, our Weights Predictor does not have an impact on the rich semantic encoding of CLIP descriptors, and our merged CLIPs generalize to out-of-distribution classes. Observe in Figure 5 several accurate 3D detections in Replica and ScanNetv2 of classes that have not been seen during training (specifically, *guitar, coffee maker, blackboard, and scale*). The mIoU for these examples is over 60%. We further illustrate this aspect in Figure 4, where we test our representation with zero-shot complex language queries. Observe in these qualitative samples how our merged CLIP vectors retain objects’ properties and affordances outside

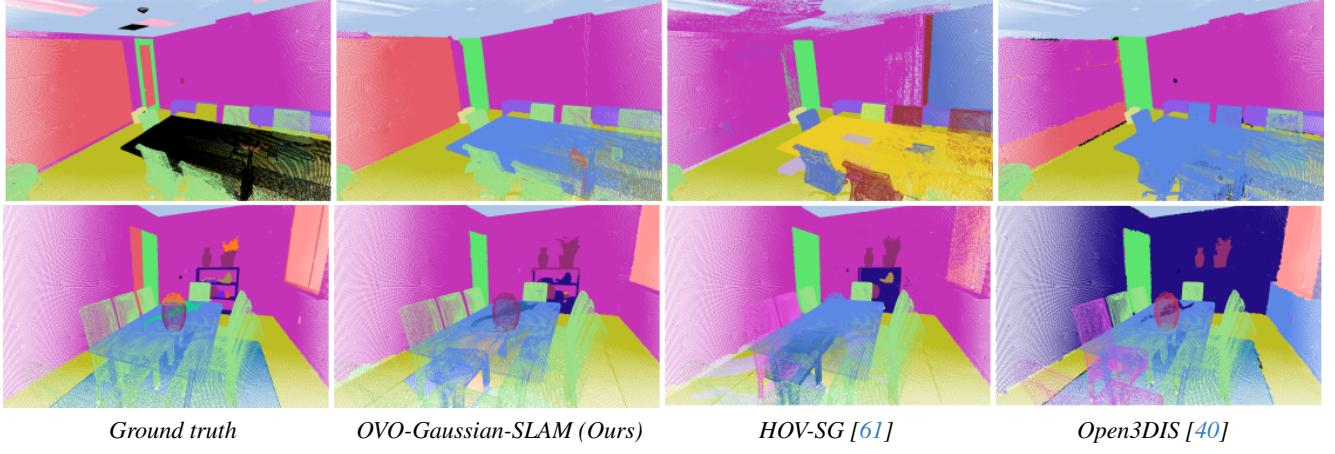


Figure 6. **3D semantic segmentation on Replica.** OVO yields on average more accurate results in comparison to two offline baselines.

Method	ScanNet20				ScanNet200				
	mIoU	mAcc	f-mIoU	f-mAcc	mIoU	mAcc	f-mIoU	f-mAcc	
HVS	HOV-SG [61]†	34.4	51.1	47.3	61.8	11.2	18.7	27.7	37.6
	Open3DIS [40] (SigLip)‡‡	37.3	52.8	57.0	67.9	17.8	23.7	27.9	34.1
	OpenScene(Ensemble) ‡‡	44.6	61.9	57.6	71.0	9.4	12.6	27.8	32.0
	OVO-mapping (ours)	38.1	50.5	57.6	70.5	17.2	25.3	45.4	56.4
	OVO-Gaussian-SLAM (ours)	29.3	41.1	43.0	59.5	11.8	18.8	30.1	42.6
	OVO-ORB-SLAM2 (ours)	31.2	42.9	49.6	65.5	12.0	19.2	36.8	49.5
FVS	Open3DIS [40] (SigLip)‡	24.7	40.9	32.5	45.3	9.4	17.0	22.9	32.2
	OpenScene‡‡	47.0	70.3	57.7	69.8	11.6	22.8	24.5	29.2
	OVO-mapping (ours)	37.3	58.9	55.13	69.4	17.4	35.9	44.3	57.8

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

of the training distribution. OVO allows us to query 3D representations simultaneously to its estimation, which is not possible in previous offline methods. For example, the descriptor are able to differentiate between two trash bins based on the recycling symbol in one of them, despite both having “bin” as ground-truth classification.

4.2. 3D Semantic Segmentation

Replica. Table 2 shows segmentation results for OVO configurations and relevant baselines. OVO outperforms all baselines on the aggregated mIoU and mAcc (‘All’ column). Both OVO-Gaussian-SLAM and OVO-ORB-SLAM2 also outperform offline baselines. This is notable, as both our implementations estimate the camera poses and scene geometry, while several baselines (indicated in the table) use ground-truth geometry. Both implementations estimate sharp and accurate reconstructions, showing that different camera trackers do not significantly impact our pipeline. Thanks to the generalization of Weights Predictor,

all OVO implementations have a significantly better performance on Tail categories. As a consequence, OVO-mapping is able to slightly outperform the more expensive HOV-SG using the same backbones (SAM and ViT-H/14).

As seen in Fig. 6, OVO is able to properly segment and classify 3D instances like chairs and tables, that are confused by other baselines that naively add too much context information into CLIP descriptors. OVO even outperforms the ground-truth on some instances. See on “office4” scene’s ground-truth (Fig. 6 top left) how the table’s label is missing and one chair has been miss-classified as floor. This highlights the benefit of open-set system on situations that would lead to failure of previous SLAM algorithms, that rely on known 2D semantics [31, 66].

ScanNetv2. Results, summarized in Tab. 4, show how OVO-mapping outperforms 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-

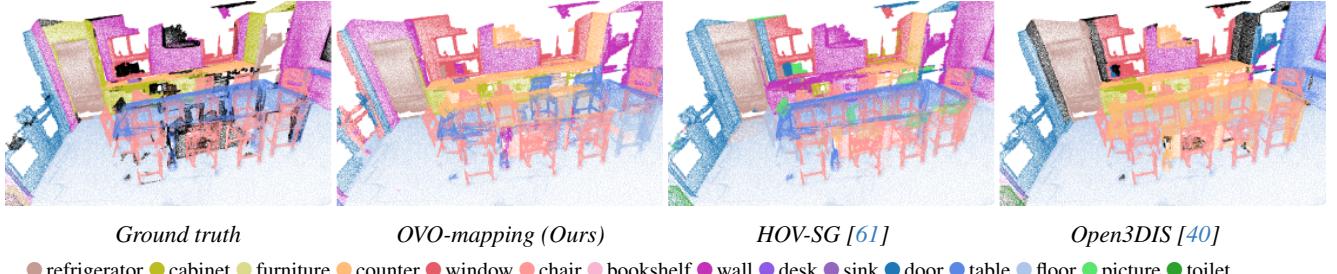


Figure 7. **Qualitative results on ScanNetv2 [11]**. We visualize the 3D semantic segmented point cloud of two SotA offline baselines with the ground-truth on ScanNetv2 *scene0011_00*. Our method achieves competitive results despite the more challenging online setting.

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. Qualitative analysis, see bottom Fig. 7, 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; HOV-SG is predicting parts of the chairs as *picture* and classifying most of the counter and furniture as *wall*, while Open3DIS completely miss-classified the table as *counter* and a cabinet as *refrigerator*.

The difference between OVO’s two SLAM versions and OVO-mapping is bigger in ScanNetv2 than in Replica (compare Tab. 2 and Tab. 4). This is due to image blur and noisy depths in ScanNetv2, that propagates to the estimated camera poses and scene maps. One one-side, due to the blurred images and depth Gaussian-SLAM optimizes its Gaussian Splatting representation generating a sparse point-cloud resulting in a coarse 3D segmentation. On the other, on scenes where ORB-SLAM2 performs a pose-graph optimization, this is not synchronized with OVO’s semantics.

Runtime cost. OVO’s efficiency, by design, makes it 3× faster than OpenNerf and 80× faster than HOV-SG, see Tab. 5. In contrast with HOV-SG that, relies on an expensive hierarchical merging of segments, requiring almost ×10 more RAM, OVO’s design is reflected on the lowest RAM and GPU vRAM usage. Further statistics, in Tab. 6,

Method	avg / max vRAM	max RAM	Avg time
HOV-SG [61]	6 / 12 GB	139 GB	11h 12m
OpenNeRF [15]	4 / 22 GB	44 GB	19m 3s
OVO-mapping (ours)	4 / 8 GB	12 GB	6m 7s

Table 5. **Runtime statistics on Replica**. OVO is significantly faster to reconstruct a scene, requiring less RAM and GPU vRAM.

show the system runs at ∼ 0.5 – 1 keyframes per second on both Replica and ScanNetv2. Therefore, it is compatible with real-time SLAM system, where the critical camera

tracking runs at video rate while the mapping runs at lower frequencies. Finally, highlight how the low GPU usage and representation size enable its use on consumer GPUs.

Dataset	avg / max vRAM	checkpoint	KFPS	#KF
Replica	4.2 / 8.1 [GB]	12±1 [MB]	0.5±0.2	200±0
ScanNet	4.4 / 7.7 [GB]	15±5 [MB]	0.8±0.3	755±833

Table 6. **OVO-mapping runtime statistics**. GPU vRAM, checkpoint size, keyframes per sec. and number of keyframes.

4.3. Limitations

Despite state-of-the-art performance on 3D semantic segmentation, OVO’s instance detection and tracking has margin for improvement. For example, it lacks a deeper integration between the semantic and SLAM modules that supports loop closures. Furthermore, its current cost limits its application to those where semantics can be computed at 0.5 – 1 FPS rather than real-time. Finally, our Weights Predictor has a slight bias towards the most common CLIP descriptors and classes, despite its generalization abilities. A more robust training, observing a wider set of classes and including also semantic properties, would strengthen its capabilities.

5. Conclusions

In this paper we present OVO, the first open-vocabulary online mapping 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 neural network, which we show to be more effective than alternative approaches while maintaining generalization capabilities. We outperform offline baselines in both computation and segmentation metrics on the Replica and Scan-

Net 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] 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)
- [6] 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)
- [7] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. Suma++: Efficient LiDAR-based semantic SLAM. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537. IEEE, 2019. (see page: 1)
- [8] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. (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: 6, 7, and 8)
- [12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabинovich. 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, Shehyrar 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, 3, 5, 6, 8, and 1)
- [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] Yingchun Fan, Qichi Zhang, Yuliang Tang, Shaofen Liu, and Hong Han. Blitz-slam: A semantic slam in dynamic environments. *Pattern Recognition*, 121:108225, 2022. (see page: 1)
- [18] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. In *ICLR*, 2024. (see pages: 2, 5)
- [19] 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)
- [20] 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)
- [21] 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)
- [22] 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 5)
- [23] 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)
- [24] 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, and 3)
- [25] 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)
- [26] 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, 1, and 5)
- [27] 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)
- [28] 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)
- [29] 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)
- [30] 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)
- [31] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pages 163–179. Springer, 2024. (see pages: 1, 2, 3, 5, and 7)
- [32] 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)
- [33] 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)
- [34] 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)
- [35] 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)
- [36] 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)
- [37] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. (see pages: 1, 5)
- [38] 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)
- [39] 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)
- [40] 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, 7, and 8)
- [41] 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)
- [42] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin 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, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. (see page: 2)
- [43] 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)
- [44] 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, and 3)
- [45] 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)
- [46] 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)
- [47] 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, and 3)
- [48] 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, 5)
- [49] 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)
- [50] 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)
- [51] 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. (see pages: 6, 1)
- [52] 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)
- [53] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. (see page: 2)
- [54] 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)
- [55] 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: 6)
- [56] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alphaclip: 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)
- [57] 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)
- [58] 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)
- [59] 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)
- [60] 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)
- [61] 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, 7, 8, and 1)
- [62] 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)
- [63] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019. (see page: 1)
- [64] 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: 2, 4, and 5)
- [65] 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 pages: 1, 5)
- [66] 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, 5, and 7)
- [67] 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, 5, and 3)
- [68] Zhuoyang Zhang, Han Cai, and Song Han. Efficientvit-sam: Accelerated segment anything model without performance loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7859–7863, 2024. (see page: 2)
- [69] Xu Zhao, Wenchoao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023. (see page: 2)

- [70] 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](#))
- [71] 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](#))
- [72] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022. (see page: [1](#))

Open-Vocabulary Online Semantic Mapping for SLAM

Supplementary Material

A. Evaluation details

A.1. Datasets

ScanNet++ contains 1752×1168 RGB-D images of real indoor scenes with ground-truth 3D meshes and instance and semantic annotations. For training, we use the top 100 semantic labels from the more than 1.6K annotated semantic classes, and evaluate on the whole set of 1.6K. 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 sets of annotations, the original with 20 classes (ScanNet20), and an expanded set with 200 classes (ScanNet200) [51]. We evaluate on the 5 scenes subset used by HOV-SG[61](HVS), and on the whole validation set of 312 scenes (FVS). Despite some overlap in physical scenes, ScanNet and ScanNet++ were captured years apart, with different trajectories and sensors, making images and reconstructions significantly different. 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 . For Replica we use the common 8 scenes subset (*office-0...4, room-0...2*) with Nice-SLAM camera trajectories [72].

A.2. Implementation

The Weights Predictor has a 5 layers transformer encoder with 8 heads and a 4 layers MLP. It was trained on ScanNet++ train set for 15 epochs, with batch size 512, on 4 V100 GPUs. As pre-processing, we computed segmentation masks on images, matched these with their ground-truth 2D semantic labels, and pre-computed input and target CLIP embeddings to speed-up training.

Regarding OVO, we use the pixel size of segmented 2D masks as metric of viewpoints quality, and show results selecting the final descriptor between the 10 best keyframes of each 3D segment. Except when stated otherwise, we relied on SAM2.1-1 for 2D instance segmentation, and SigLip ViT-SO400 for CLIP descriptors. We query the models with the set of classes of each dataset using the template “This is a photo of a *{class}*”. For fairness in OVO evaluation, we reproduce previous approaches’ [15, 43, 57, 61] keyframes selection and querying. We select as keyframes 1 every 10

frames. The representation is queried with each dataset’s semantic classes, and each 3D segment is matched to the class with higher similarity. Following HOV-SG, the vertices of our estimated point-cloud are matched to the vertices of ground-truth meshes using KD-tree search with 5 neighbors. Profiling experiments were run on Ubuntu 20, with an i7-11700K CPU, an RTX-3090 GPU, 64 GB of RAM and 150 GB of swap.

Due to slight differences in metrics computation, we reproduced HOV-SG, and Open3DIS, in both Replica and ScanNetv2. For a fairer comparison with Open-3DIS we implemented it we SigLIP ViT-SO400M rather than its base CLIP ViT-L/14. We report OpenNeRF official metrics on Replica, and were not able to make it converge in ScanNetv2.

B. System ablations

In this section we report minor ablations and experiments performed during OVO development using ScanNet++ training set. First we report an ablation of different foundation models for 2D instance segmentation, and Language-Image features extraction. Then, we ablate the algorithm to merge different CLIP descriptors and validate our proposed Weights Predictor. We profit from the Weights Predictor to reduce the number of CLIPs descriptors computations and evaluate the impact of the number of views on the selection of the final descriptor of 3D instances. After that, we present a mask bleeding problem that arises from depth estimation inaccuracies, and how we tackled it. Finally, we report an overall profiling of the system using different previously ablated components.

While the segmentation backbones where ablated on a single scene from ScanNet++, we used an extended set of five scenes for CLIP [47] models and similarity computation, to ablate the set of fixed weights, the evaluation of the number of viewpoints, and the mask bleeding. Then we used a different set of 10 scenes for the overall profiling to avoid overfitting on the previous set. Regarding Weights Predictor training was done using the 230 scenes from ScanNet++ training set, and validation against baselines was performed on ScanNet++ 50 scenes validation set, and on ADE20K-150. We measured mean Intersection over Union (mIoU) of the 3D semantic segmentation.

As starting point, segmentation masks are computed using SAM 2 [26]; CLIP vectors are computed from masks using SigLIP-384; for each mask three vectors are computed and weighted together as introduced by HOV-SG [61]; each 3D object gets assigned the CLIP vector from the view that minimizes the L1 distance to its other views. Finally seman-

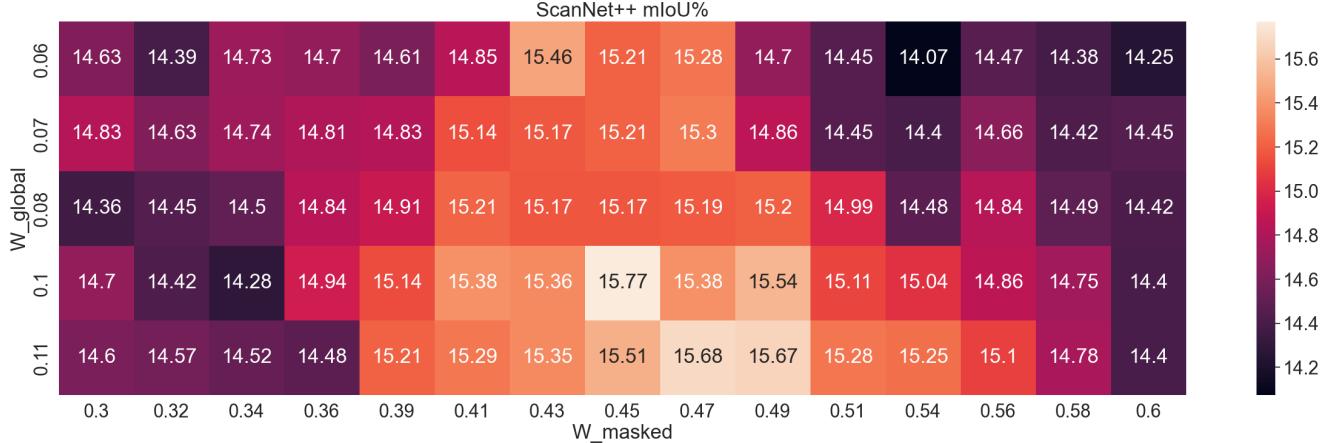


Figure 8. Grid search for CLIP weights merging on five scenes from ScanNet++ [64].

tic classes are matched to each 3D object using the similarity approach presented by LangSplat [44].

B.1. Foundation Models

SAM Since its release, Segment Anything Model (SAM) [26] has been the state-of-the-art for out-of-the box instance segmentation on different fields. It’s segment-everything mode extracts multiple masks from a single image, taking an input a grid of point on the image. Nevertheless, this mode has a low throughput mainly due to the post-processing required to filter duplicate and bad segmentation masks. Although several methods claim up to $\times 100$ speed-ups with respect to SAM, these speed-ups are measured when segmenting a single object on the image, and do not measure the segment-everything mode and its post-processing.

In this ablation the evaluated models are SAM [26], SAM 2 [48], FastSAM [69], and EfficientViTSAM [68]. The evaluation in Tab. 7 shows how when segmenting everything these methods do not imply an improvement against a SAM implementation with tuned hyper-parameters.

281bc17764		
SAM backbone	mIoU↑	Latency [s]↓
FastSAM [69]	5.0	0.40 ± 0.27
EfficientViTSAM [68]	17.1	4.19 ± 0.85
EfficientViTSAM [68] - tuned	15.1	0.68 ± 0.05
SAM [26]	19.0	5.43 ± 1.83
SAM [26] - tuned	18.1	0.84 ± 0.13
SAM 2 [48] - tuned	19.1	0.71 ± 0.10

Table 7. Segmentation backbone ablation.

Architecture	Resolution	mIoU [%]	Latency [s]
DFN-ViT-B-16		10.92	0.100 ± 0.022
DFN-ViT-L/14		11.89	0.173 ± 0.031
DFN-ViT-H/14	224 × 224	13.22	0.286 ± 0.054
OpenCLIP ViT-H/14		12.71	0.283 ± 0.053
SigLIP-SO400M		13.78	0.229 ± 0.026
SigLIP-SO400M 384	384 × 384	15.35	0.442 ± 0.080
DFN-ViT-H/14-378	384 × 384	12.96	0.664 ± 0.136

Table 8. CLIP ablation results on 5 scenes from ScanNet++.

Visual-Language descriptors. To compute image-language features we rely on the family of CLIP and its variants. To select the CLIP architecture we evaluate the difference in performance and latency of different SOTA models to compute CLIP embeddings:

- OpenCLIP [21] base ViT-H-14, trained on LAION-2B English [53] at a resolution of 224×224 , using CLIP’s cosine similarity.
- DFN [18] ViT-B-16, ViT-L-14, and ViT-H-14 trained on the dataset DFN-5b [18] with input images of 224×224 , and a ViT-H-14 finetuned at resolution 384×384 , using CLIP’s cosine similarity.
- two SigLIP’s Shape-Optimized 400M parameter ViT (ViT SO-400M), trained on WebLI English dataset [8] at 224×224 , with one fine-tuned at 384×384 , and optimized using SigLIP’s cosine similarity.

In this ablation each backbone is evaluated using the similarity with which they were trained, without ensembling, and using the template “*This is a photo of a {class}*”. The results in Tab. 8 show a clear trade-off between segmentation performance, and model latency. SigLIP-384 achieves the best mIoU, while SigLIP at 224×224 has the best balance between mIoU and speed. Overall, this ablation shows

the importance of selecting the proper CLIP backbone, with a difference of almost 5% between the best and the worst model.

Similarity computation. Initially, CLIP [47] presented the cosine similarity, $\cos(\phi_{\text{qry}}, \phi_{\text{img}})$ to compute the distance between the text, ϕ_{qry} , and image, ϕ_{img} , embeddings. SigLIP [67] adapted it to its loss function, as Sigmoid ($\cos(\phi_{\text{qry}}, \phi_{\text{img}}) \times \tau + b$), including a Sigmoid operation, and the learned inverse temperature, $t = \frac{1}{\tau}$, and bias b parameters. To classify, both approaches assigned to an image the class of the query that generated the highest similarity. Also based on CLIP’s experiments, to compute the cosine similarity HOV-SG [61] computed query embeddings as $\phi_{\text{qry}} = \frac{\phi_{\text{cl}} + \phi_{\text{temp}}}{2}$, where ϕ_{cl} is the text embedding computed from the class name, and ϕ_{temp} is the text embedding computed from the phrase resulting of inserting the class into the template “*There is {class} in the scene*”. In contrast, LERF [24] proposed to compute the cosine similarity between the image and text embeddings as

$$\min_i \frac{\exp(\cos(\phi_{\text{qry}}, \phi_{\text{img}}))}{\exp(\cos(\phi_{\text{qry}}, \phi_{\text{img}})) + \exp(\cos(\phi_{\text{can}}^i, \phi_{\text{img}}))}, \quad (2)$$

where ϕ_{can}^i is the text embedding of one of the predefined canonical queries *object, things, stuff, texture*.

Using the SigLIP ViT-SO400M model to compute CLIP vectors, we compare between:

- computing query embeddings, ϕ_{qry} , only with the template “*This is a photo of a {class}*” or as an ensemble averaging the template embedding with the class embedding;
- and computing SigLIP’s cosine similarity or LERF’s cosine similarity.

Results in Tab. 9 show how the basic configuration of using SigLIP similarity without ensemble achieves the best performance. **From here on, all experiments will proceed using basic cosine similarity without ensemble.**

	Cosine similarity	LERF’s similarity
w ensemble	14.75%	14.75%
w\o ensemble	15.35%	14.98%

Table 9. Similarity computation ablation on 5 scenes from ScanNet++ measuring semantic 3D mIoU.

B.2. CLIP descriptors merging

To focus CLIPs descriptors to elements in an image we start following HOV-SG [61] approach. For each mask segmented by SAM, HOV-SG proposed to compute CLIP embeddings combining the information of the complete image, the masked image without background, and a bounding box

of the mask including background. For each segmentation mask i , its corresponding CLIP vector F_i is computed as

$$F_i = F_{\text{global}} \times w_{\text{global}} + F_{\text{local}_i} \times (1 - w_{\text{global}}), \quad (3)$$

with

$$F_{\text{local}_i} = F_{\text{masked}_i} \times w_{\text{masked}} + F_{\text{bbox}_i} \times (1 - w_{\text{masked}}), \quad (4)$$

combinig the CLIP vector of the whole image, F_{global} , the CLIP vector of only the segmentation mask without background, F_{masked_i} , and the one of the bounding box of the segmentation mask including background, F_{bbox_i} .

HOV-SG [61] used

$$w_{\text{global}} = \text{Softmax}(\cos(F_{\text{global}}, F_i)), \quad (5)$$

and $w_{\text{masked}} = 0.4418$. Nevertheless, the use of the Softmax introduced a dependency between the different embeddings extracted on the same frame. To avoid computing all CLIP embeddings on every frame, we remove the Softmax and perform a grid search of w_{masked} and w_{global} . The best performance is achieved for $w_{\text{global}} = 0.45$ and $w_{\text{masked}} = 0.0975$ as shown in Fig. 8.

Weights Predictor Rather than relying on 3 fixed-weights that ideally should be tunned for each scene, we developed the Weights Predictor to estimate the corresponding weight for each image. After training on ScanNet++ train set with the top 100 semantic labels, we evaluate its performance on the ScanNet++ validation set using the total set of 1.6k queries, both including (w.top 100) and excluding (w/o. top 100) classes seen during training. For a stronger distribution switch we also evaluate on ADE20k-150. Comparing its performance against HOV-SG’s approach, and our variation of HOV-SG’s using three fixed weights, the Weights Predictor outperforms the baselines using all the labels, Tab. 10. Excluding from the metrics the 100 labels seen during training, we can observe how the Weights Predictor performance drops with respect to the baselines.

Despite the slight bias toward classes at training, it still outperform on freq. weighted metrics of classes that weren’t seen during training, and on novel data on the ADE20k-150 dataset.

Although, OVO-mapping evaluation in Replica and ScanNetv2 leave additional segmentation metrics on classes outside the training set (Tab. 11 and Fig. 5) that showcase how the bias does not have an impact on our Weights Predictor’s generalization. **From here on, all experiments will proceed using the Weights Predictor.**

B.3. Additional heuristics

Nº of best views. To reduce the expensive CLIP computation for each frame, we evaluate the impact of using only

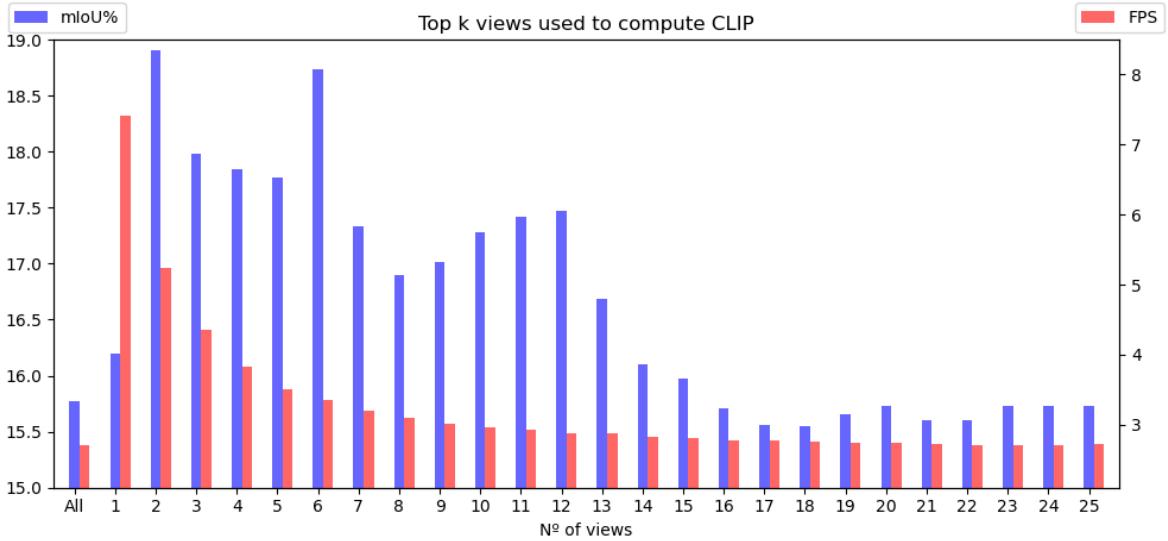


Figure 9. **Evaluation using only top views to compute CLIP on 5 scenes from ScanNet++ [64].** While using more than one view has substantial impact on the runtime, it also improves segmentation accuracy. However, too many views also degrade the segmentation accuracy.

Method	S++ w. top 100				S++ w/o. top100				ADE20k-150			
	mIoU	mAcc	f-mIoU	f-mAcc	mIoU	mAcc	f-mIoU	f-mAcc	mIoU	mAcc	f-mIoU	f-mAcc
HOV-SG	9.4	15.9	12.8	15.9	8.3	15.1	8.4	13.6	21.9	53.7	22.3	34.9
Fixed-weights	9.4	15.9	13.1	16.3	8.3	15.1	8.4	13.8	22.4	53.9	23.1	35.5
CLIP-merger	10.7	16.9	36.1	45.3	7.3	12.8	9.9	15.0	23.4	49.3	28.7	41.2

Table 10. Our Weights Predictor vs. baselines on: ScanNet++ (S++) using 1.6k queries (metrics on observed 495 labels, w. and w/o. the top 100 used at training), and ADE20k with 150 labels. Color indicates First , second , and third best.

scale	toaster	oven	blackboard	coffee	maker	guitar	projector	screen
mIoU%	75.1	78.53	61.4	67.0	62.68	64.1		
mAcc%	81.2	94.07	76.1	86.7	86.79	86.8		

Table 11. **Weights Predictor generalization.** 3D metrics on ScanNetv2 of some classes not seen during training.

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 quality of an image is based on the area of the object’s 2D segmentation in it.

For a sequence of 51 keyframes, we evaluate for $k \in \{1, \dots, 51\}$, being *all* using all the views to compute objects 3D vectors. The results show, see Fig. 9, that neither using only the best nor using all the views are robust enough to noise. For the set of 5 scenes on this experiment, the best values of k are between 2 and 7, achieving an mIoU around 18%, almost 3 points better than using all observa-

tions, although, the perfect value of will probably be scene and object dependent. We decide to set use 10 views as a balance to avoid useless computation of CLIP vectors and being resistant to noisy images.

Masks bleeding. Observing OVO-SLAM matching results, we noticed some problems related with SAM’s masks. When some 3D points are projected on the edges of a 2D mask to which they do not belong, they are wrongly clustered into it and matched to a 3D instance. Then, when these are seen again they will propagate the wrongly assigned ID. This phenomenon can be observed in particular on the edges of objects, where the depth and masks are less accurate, and masks propagate the ID of the object to the background, as seen in Fig. 10. To compensate it we developed two approaches:

- First, we add a filter to only keep matches of 3D points that are assigned to the same object in two consecutive frames;
- second, we apply a low pass filter on the depth map to mask the edges of the objects and avoid matching points

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 12. **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).

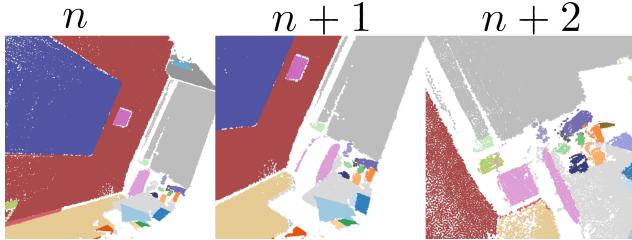


Figure 10. **Mask bleeding** and propagation produced by masks inaccuracy. The edges of the chair (pink) bleed to the background at k_n , and therefore the segment label is wrongly propagated to it in the following keyframes.

of scenes the best 3D segmentation is achieved with the largest models using all points of view. Nevertheless, the best trade-off can be achieved reducing the number of views and the CLIP model.

around them.

Results on Tab. 13 show how while using the depth filter does improve the average mIoU, the limitation to match in consecutive frames does not. As a consequence we keep only the depth filter although it does not completely solve the problem.

Config	mIoU↑
Base	15.80%
w depth filter	16.16%
w consecutive KF filter	15.07%
w both	15.82%

Table 13. **Mask bleeding solutions’ ablation on 5 scenes from ScanNet++ [64] measuring semantic 3D mIoU.**

Overall profiling. Finally, we quantify the latency-quality trade-off in our architecture evaluating selected foundation models and number of views against less powerful alternatives. This evaluation is performed on a different set of 10 scenes from ScanNet++ to avoid over-fitting to the previous 5 scenes. For 2D segmentation we evaluate SAM [26] with ViT-H/14 encoder (1-H), and SAM 2.1 [48] with Hiera large (2.1-L) and Hiera tiny (2.1-t) image encoders. For CLIP extraction, we evaluate DFN ViT-H/14-378 [18] and SigLIP-SO400 [67] both with input images of 384 pixels. The results in Tab. 12 show that in this set