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 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 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 [36]), 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 [26]. Today, its applications continue to expand into other domains [14, 35]. Visual SLAM research, however, has mainly focused on geometric models, sensor fusion, processing pipelines and optimizations [5, 38, 44, 62], and much less and mostly recently on the crucial aspect of the scene representation [48, 49, 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, 40, 51, 59], semantic segmentations of point cloud maps [7, 31, 49] or implicit 3D representations [28, 30, 71]. All of them, however, are constrained to a predefined closed set of categories, limiting their applicability in real-world scenarios.

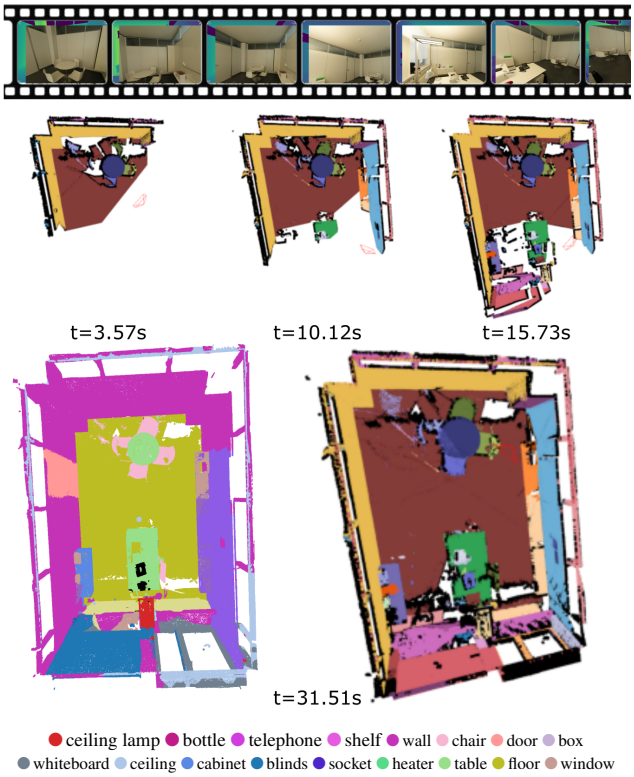


Figure 1. **OVO mapping.** Given an RGB-D set of keyframes (**top**), our method successively reconstructs a 3D open-vocabulary semantic representation of a scene over time (**middle**). At any moment of the sequence both semantic labels (**bottom left**) as well as instance labels (**bottom right**)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

Offline semantic 3D reconstructions have traditionally used a closed-set approach [1, 27, 28]. However, following the development of CLIP [46], research on open-vocabulary 3D representations has surged [2, 24, 39, 42, 43, 56, 57]. Despite the good 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 an Open-Vocabulary Online mapping algorithm, OVO, which we also integrate into two different visual SLAM pipelines. See Fig. 1 for an illus-

tration of our online reconstruction results. Our method estimates, from a RGB-D set of keyframes, a set of 3D segments that are assigned a CLIP vector per segment. Specifically, our segments are initialized by back-projecting SAM 2.1 [47] 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 relevant baselines.

2. Related Work

Unlike previous methods that rely on offline or expensive optimizations with ground-truth camera poses or predefined scene geometry, our online approach seamlessly integrates with simultaneous localization and mapping (SLAM) pipelines. Tab. 1 provides a comparative summary of recent related 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) [46], 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 [9, 18, 21, 67] and improved feature granularity, aiming to generate dense feature vectors [19, 55, 70] rather than per-image representations. While closed-vocabulary methods outperform on predefined sets, open-vocabulary offers optimization-free generalization, highly relevant for diverse applications.

2D Open-Vocabulary from RGB and RGB-D. With the advent of Neural Radiance Fields (NeRFs) [34] and 3D Gaussian Splatting (3DGS) [23], semantics have become increasingly integrated into them. LERF [24] embeds per-object multi-scale CLIP features within NeRF, enabling 2D image searches via language queries. LangSplat [43] applies the Segment Anything Model (SAM) [25] on each viewpoint to generate 2D masks, remove their background, and encode them into CLIP vectors. Features are then compressed into 3 channels and incorporated into a 3DGS [23] representation, allowing 2D semantic segmentation. Subsequent works build on LangSplat by incorporating affinity features optimized with a multi-view mask graph [6] or by integrating both CLIP and DINOv2 [41] features [45, 53]. O_2V-Mapping [57] combines CLIP and SAM within NICE-SLAM [72] for an online reconstruction although only evaluating 2D semantics. The dependence of all these methods on multi-point 3D-2D transformations (via rendering or rasterization) for computing 2D semantic features limits their semantic representation to 2D–evidenced by the lack of proper 3D evaluation.

Method	Open Vocabulary	Online	3D semantics
LERF [24]	✓	✗	✗
LangSplat [43]	✓	✗	✗
OpenScene [42]	✓	✗	✓
OpenMask3D [56]	✓	✗	✓
Open3DIS [39]	✓	✗	✓
HOV-SG [61]	✓	✗	✓
OpenNeRF [15]	✓	✗	✓
NEDS-SLAM [22]	✗	✓	✗
NIS-SLAM [66]	✗	✓	✗
SGS-SLAM [30]	✗	✓	✓
Kimera-VIO [49]	✗	✓	✓
O_2V-Mapping [57]	✓	✓	✓
OVO (ours)	✓	✓	✓

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

Offline 3D Open-Vocabulary from 3D point clouds. Most open-vocabulary 3D semantic approaches assume a known 3D point cloud. OpenScene [42] leverages OpenSeg [19] 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 [56] selects k views per object, crops its 2D SAM mask to compute a CLIP features, and then features areaverage-pooled across crops and views. Open3DIS [39] integrates SuperPoint [12] with 2D instance segmentations and a 3D instance segmentator to generate multiple 3D instance proposals, describing each with CLIP features following OpenMask3D [56]. In contrast, OpenYolo-3D [2] 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 [15] 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. In contrast, Hierarchical Open-Vocabulary

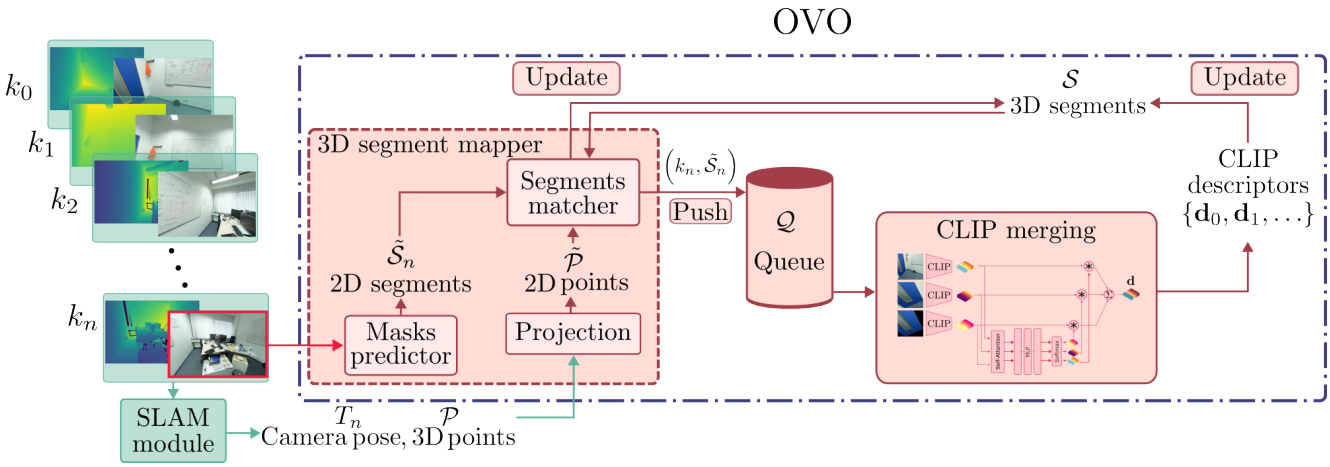


Figure 2. **OVO overview.** From a stream of RGB-D keyframes, OVO builds, online, a 3D semantic representation of the scene. It relies on a 3D segment mapper to cluster 3D points into 3D segments; a queue to distribute the CLIP extraction computation, and a novel CLIP merging method to aggregate CLIP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

3D Scene Graphs for Language-Grounded Robot Navigation (HOV-SG)[61] relies on a hierarchical global fusion approach that requires precomputing 3D segments and features for all frames. Consequently, unlike OVO, it is an offline method and incompatible with SLAM. HOV-SG first reconstructs the full point cloud from RGB+D data, using DBSCAN[16] to filter noise. Then, SAM and CLIP extract local 2D segments and corresponding CLIP vectors, which are projected onto a 3D global map. These 3D segments and features are incrementally fused by merging observations across consecutive frames. Additionally, the authors argue that relying solely on masked segments, as in LangSplat [43], 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.

Closed-Vocabulary Online Semantics. To date, online semantic methods have focused exclusively on closed vocabularies. SemanticFusion [31] was one of the first semantic SLAM pipelines, predicting per-pixel closed-set categories and fusing predictions from different views in 3D space. Fusion++ [32] uses Mask-RCNN [20] to initialize per-object Truncated Signed Distance Functions (TSDFs), building a persistent object-graph representation. In contrast, PanopticFusion [37] 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. [33] 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 [29] and SGS-SLAM [30] 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 Methodology

Fig. 2 shows an overview of OVO. Our approach takes as input a set of RGB-D keyframes ($\{k_0, \ldots, k_n\}$ 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).

3.1. Map Definition

OVO assumes a parallel-tracking-and-mapping architecture, as first defined by Klein and Murray [26] and adopted by most visual SLAM implementations [5]. Its input is a RGB-D video $\mathcal{V} = \{f_0, \ldots, f_\tau\}$, $f_\tau \in \mathbb{N}^{w \times h \times 3}_{\leq 255} \times \mathbb{R}_{>0}$ standing for the RGB-D frame of size $w \times h$ captured at time step τ. 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 $\mathcal{K} =$

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)
        $l_s \leftarrow z_l$
      end if
    end if
  end for
  $\tilde{\mathcal{S}}_n \leftarrow$ 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 $\mathcal{P}, \mathcal{S}, \tilde{\mathcal{S}}_n$
end function

$\{k_0, \dots k_n\} \subset \mathcal{V}$ 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 $P = \left(\begin{bmatrix} x & y & z \end{bmatrix}^\top, l_p\right)$ is defined by its 3D coordinates $\begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3$ and a discrete label $l_p \in \{-1, 0, 1, \dots, q\}$, that when $l_p > -1$ indicates the 3D segment the point belongs to, and when $l_p = -1$ indicates that it is unassigned to any of them. Every 3D segment $S_q = (\mathbf{d}, \kappa)$ has a unique identifier q, its semantics are described by a CLIP feature $\mathbf{d} \in \mathbb{R}^d$, 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 $\tilde{\mathcal{S}}_n = \{(s_0, l_{s0}), (s_1, l_{s1}), \dots\}$, each segment being composed of a mask s and a label l_s, that is initialized as $l_s = -1$. We then select the 3D map points in k_n's frustum and remove occluded points. Those remaining are projected to k_n obtaining 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 happen:

1. If $z_l = -1$, we set $z_l = q+1$ and initialize a new 3D segment S_{q+1} with an empty $\mathbf{d}$ (filled 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. Otherwise, the 2D segment s is a match for the 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}.

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.

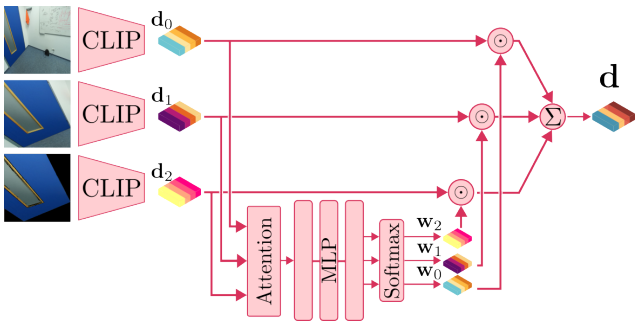


Figure 3. **CLIP merging.** Our model predicts weights $\mathbf{w}_{i=\{0,1,2\}}$ for each input CLIP descriptor $\mathbf{d}_{i=\{0,1,2\}}$. The final descriptor $\mathbf{d} = \sum_{i=0}^{2} \mathbf{w}_i \odot \mathbf{d}_i$ is the weighted average of the input ones.

3.3. CLIP Descriptors

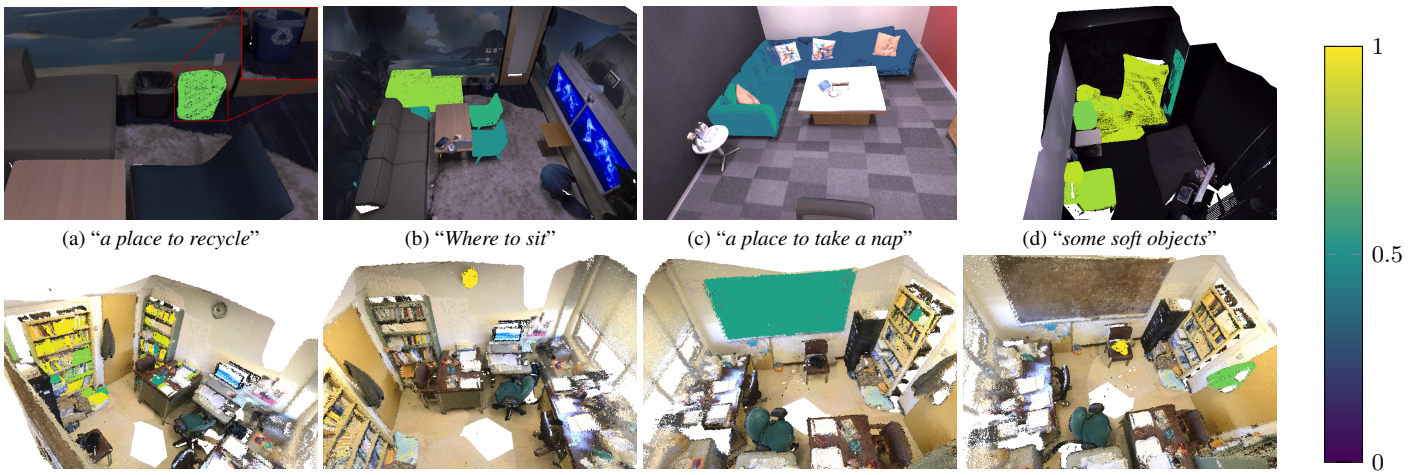
When a tuple $\left(k_q, \tilde{\mathcal{S}}_q\right)$ is popped from $\mathcal{Q}$, only the matched 2D segments for which k_q is still in the κ of their 3D instance S are selected. A CLIP descriptor $\mathbf{d}$ is computed for each of them by merging three different descriptors. Then, the final descriptor 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 $\mathcal{S}$.

3.4. CLIP Merging

Similarly to HOV-SG [61], 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

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



(a) "*a place to recycle*" (b) "*Where to sit*" (c) "*a place to take a nap*" (d) "*some soft objects*"

(e) "*an object to read*" (f) "*an object that tells the hour*" (g) "*a place to draw equations*" (h) "*something to stay warm*"

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.

weighted average with weights $\mathbf{w}_i \in \mathbb{R}^d$ ($\odot$ stands for the Hadamard product). Our weights $\mathbf{w}_{i=\{0,1,2\}}$ are predicted by a neural model, as shown in Fig. 3. Note that HOV-SG's merging is done with hand-crafted scalar weights (*i.e.*, $\mathbf{d} = \sum_{i=0}^{2} w_i \mathbf{d}_i$, $w_i \in \mathbb{R}$, more details in Appendix B.2.

As seen in Fig. 3, our model for CLIP merging takes as input the 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 that $\sum_{i=0}^{2} \mathbf{w}_i = \mathbf{1}^d$.

Our model for CLIP merging is pre-trained following SigLIP [67]. 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

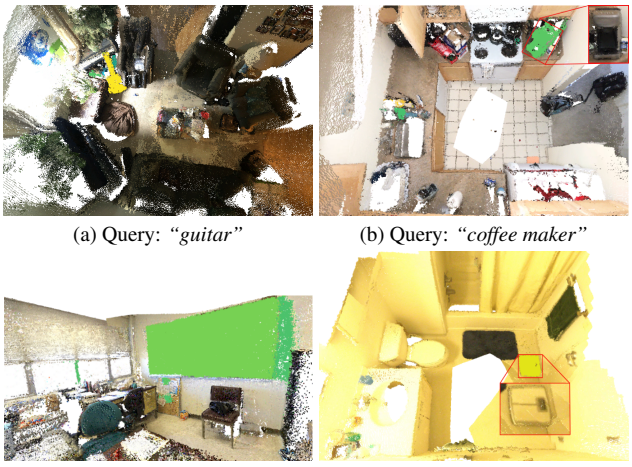
$$L = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \log \left(\frac{1}{1 + \exp(z_{ij}(-t\mathbf{d}_i \cdot \mathbf{y}_j + b))} \right), \tag{1}$$

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.

4. Experiments

We implemented three different configurations for OVO.

1. **OVO-mapping**, that uses ground-truth camera poses.
2. **OVO-Gaussian-SLAM**, for which we integrate our contributions within the Gaussian-SLAM pipeline [65].



(a) Query: "*guitar*" (b) Query: "*coffee maker*"

(c) Query: "*blackboard*" (d) Query: "*scale*"

Figure 5. **Out-Of-Distribution queries.** We highlight 3D points with mid and high similarity to the queries.

3. **OVO-ORB-SLAM2** for which we use the tracking implementation of ORB-SLAM2 [36].

In all three configurations, we use SAM2.1-l for 2D segmentation and SigLip ViT-SO400 for CLIP descriptors. We also implement an OVO-mapping Lite with SAM1-H and CLIP ViT-H/14. For more implementation details on CLIP merging and OVO refer to Appendix A.2.

Baselines. As detailed in Section 2, existing semantic SLAM pipelines do not construct a 3D representation that can be evaluated using 3D metrics for open-set classes. Instead, they either rely on 2D semantic representations [22, 66] or assume known 2D semantic labels within a closed

Method	Online	Pose	Geometry	All		Head		Common		Tail	
				mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
OpenScene [42] (Distilled)	✗	GT	GT	14.8	23.0	30.2	41.1	12.8	21.3	1.4	6.7
OpenScene [42] (Ensemble)	✗	GT	GT	15.9	24.6	31.7	44.8	14.5	22.6	1.5	6.3
OpenNeRF [15]	✗	GT	Est.	20.4	31.7	35.4	46.2	20.1	31.3	5.8	17.6
HOV-SG [61]	✗	GT	Est.	22.5	34.2	35.9	44.2	23.6	42.3	8.0	16.1
Open3DIS [39] (SigLip)	✗	GT	GT	25.6	38.7	**49.7**	**64.4**	22.1	42.4	4.9	9.4
O_2V-mapping [57]	✓	GT	Est.	7.8	13.9	19.3	27.5	4.0	6.7	0.1	7.5
OVO-mapping Lite (ours)	✓	GT	Est.	22.8	35.5	35.2	45.0	22.1	**44.6**	11.0	16.9
OVO-mapping (ours)	✓	GT	Est.	27.0	**39.1**	45.0	59.9	**25.1**	38.5	11.0	18.8
OVO-Gaussian-SLAM (ours)	✓	Est.	Est.	**27.1**	38.6	44.1	58.0	25.0	39.0	**12.1**	**18.9**
OVO-ORB-SLAM2 (ours)	✓	Est.	Est.	26.8	38.4	44.0	58.7	24.7	38.0	11.8	18.7

Table 2. **Results for Replica**. OVO delivers competitive results and, on average, outperforms all baselines.

vocabulary [30, 66]. Therefore, we compare OVO against O_2V-mapping, the only baseline that also builds an online representation, and the offline 3D open-vocabulary baselines OpenScene [42], OpenNeRF [15], Open3DIS [39] and HOV-SG [61]. Additionally, we evaluate our computational cost against O_2V-mapping, HOV-SG and OpenNeRF, but exclude Open3DIS and OpenScene, as they rely on pre-processed 3D geometry and features.

Datasets. We train CLIP merging using the top 100 semantic labels from ScanNet++, and evaluate it on the full set of over 1.6K semantic classes. Additionally, we assess OVO for open-vocabulary 3D semantic segmentation on ScanNetv2 [11] and Replica [54]. For ScanNetv2, we use both the original annotation set with 20 classes (ScanNet20) and the expanded set with 200 classes (ScanNet200) [50]. We evaluate on the full validation set of 312 scenes (FVS) and the 5-scene subset used by HOV-SG (HVS). For Replica, we use the standard 8-scene subset (*office-0...4*, *room-0...2*) and evaluate on its 51 annotated classes. For more details on the datasets, refer to Appendix A.1.

Metrics. Semantic segmentation is typically evaluated using *mean Intersection Over Union* (mIoU) and *mean Accuracy* (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 [15], we also report mIoU and mAcc, categorizing labels into tertiles based on their frequency (*head*, *common*, and *tail*). In ScanNetv2 and ScanNet++, we further present metrics weighted by the label frequency in the ground truth (f-mIoU and f-mAcc). Additionally, we analyze our computational 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). For OVO, we also report the final representation size (in MB) and the time to process each keyframe (KFPS). Each table highlights **first**, second, and third best results.

4.1. CLIP Merging

Tab. 3 presents segmentation results for our CLIP merging approach against HOV-SG's, on novel scenes from ScanNet++ with an expanded label set. Ours significantly outperforms HOV-SG's, particularly in frequency-weighted metrics.

Method	mIoU	mAcc	f-mIoU	f-mAcc
HOV-SG merging	9.4	15.9	12.8	15.9
Our CLIP merging	**10.7**	**16.9**	**36.1**	**45.3**

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

Notably, our CLIP merging preserves the rich semantic encoding of CLIP descriptors, allowing our merged CLIPs to generalize to out-of-distribution classes. As shown in Fig. 5, our method accurately detects in 3D several unseen classes across Replica and ScanNetv2, including *guitar*, *coffee maker*, *blackboard*, and *scale*. The mIoU for these examples exceeds 60%. We further highlight this capability in Fig. 4, where we evaluate our representation using zero-shot complex language queries. These qualitative examples demonstrate how our merged CLIP vectors retain object properties and affordances beyond the training distribution. For instance, our descriptors can distinguish between two trash bins based on a recycling symbol on one of them, despite both being labeled simply as "bin" in the ground truth. Unlike previous offline methods, OVO enables real-time querying of the 3D representations while they are being estimated online.

4.2. 3D Semantic Segmentation

Replica. Tab. 2 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-

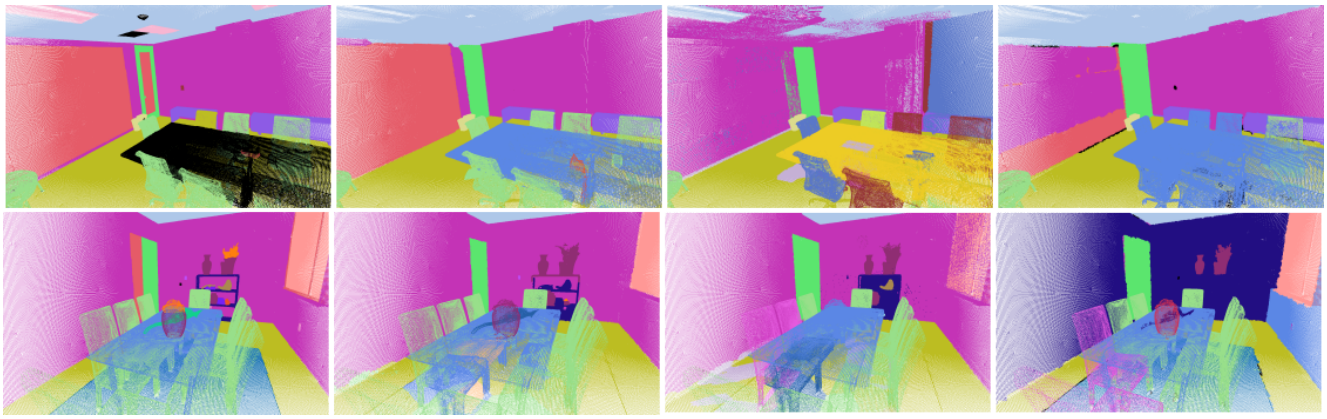


Figure 6. **3D semantic segmentation on Replica.** OVO yields on average more accurate results in comparison to two offline baselines.

	Method	Online	Pose	Geo-metry	ScanNet20 mIoU	ScanNet20 mAcc	ScanNet20 f-mIoU	ScanNet20 f-mAcc	ScanNet200 mIoU	ScanNet200 mAcc	ScanNet200 f-mIoU	ScanNet200 f-mAcc
HVS	HOV-SG [61]	✗	GT	Est.	34.4	51.1	47.3	61.8	11.2	18.7	27.7	37.6
HVS	Open3DIS [39] (SigLip)	✗	GT	GT	37.3	52.8	57.0	67.9	**17.8**	23.7	27.9	34.1
HVS	OpenScene(Ensemble)	✗	GT	GT	**44.6**	**61.9**	**57.6**	**71.0**	9.4	12.6	27.8	32.0
HVS	OVO-mapping (ours)	✓	GT	Est.	38.1	50.5	**57.6**	70.5	17.2	**25.3**	**45.4**	**56.4**
HVS	OVO-Gaussian-SLAM (ours)	✓	Est.	Est.	29.3	41.1	43.0	59.5	11.8	18.8	30.1	42.6
HVS	OVO-ORB-SLAM2 (ours)	✓	Est.	Est.	31.2	42.9	49.6	65.5	12.0	19.2	36.8	49.5
FVS	Open3DIS [39] (SigLip)	✗	GT	GT	24.7	40.9	32.5	45.3	9.4	17.0	22.9	32.2
FVS	OpenScene	✗	GT	GT	**47.0**	**70.3**	**57.7**	**69.8**	11.6	22.8	24.5	29.2
FVS	OVO-mapping (ours)	✓	GT	Est.	37.3	58.9	55.13	69.4	**17.4**	**35.9**	**44.3**	**57.8**

Table 4. **Quantitative results on ScanNetv2.** OVO-mapping outperforms offline baselines, in particular for frequency-weighted metrics and for the ScanNet200 set.

SLAM2 surpass offline baselines. This is particularly noteworthy since both implementations estimate camera poses and scene geometry, whereas several baselines (indicated in the table) rely on ground-truth geometry. Despite using different camera trackers, both implementations produce sharp and accurate reconstructions, demonstrating that tracking variations do not significantly impact our pipeline. Thanks to the strong generalization of our CLIP merging, all OVO implementations have a significantly better performance on *tail* categories. As a result, OVO-mapping Lite slightly outperforms the more computationally expensive HOV-SG using the same backbones, and is $+15$ points better than O_2V, the only other approach that estimates online the 3D semantic representation.

As shown in Fig. 6, OVO effectively segments and classifies 3D instances, such as chairs and tables, that other baselines struggle with and often misclassify them due to the excessive context information incorporated into CLIP descriptors. In fact, OVO even outperforms the ground truth in some instances. For example, in the "office4" scene (top left of Fig. 6), 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 [30, 66], would fail.

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-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 misegmenting several objects. OVO mapping labels part of the table as counter and over segments

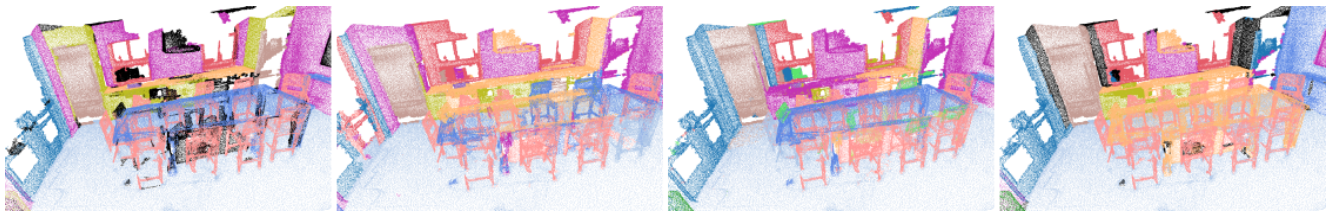


Figure 7. **Qualitative results on ScanNetv2.** We visualize the 3D semantic segmentations of OVO-mapping and two offline baselines against the ground-truth on ScanNetv2 *scene0011_00*. Our method achieves competitive results despite the more challenging online setting.

a window. HOV-SG is predicting parts of the chairs as *picture* and classifying most of the counter and furniture as *wall*. Open3DIS completely misclassified 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. 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.

Computational footprint. OVO is $3\times$ faster than OpenNerf and $80\times$ faster than HOV-SG, as shown in Tab. 5. In contrast with HOV-SG, that relies on an expensive hierarchical merging of segments, requiring almost $\times 10$ more RAM, OVO features the lowest RAM and GPU vRAM usage. Additional results, in Tab. 6, show that

Method	vRAM Avg / Max	RAM Max	Time Avg
HOV-SG [61]	6 / 12 GB	139 GB	∼11h
OpenNeRF [15]	4 / 22 GB	44 GB	∼20m
O_2V [57]	19 / 23 GB	17 GB	∼1h30m
OVO-mapping (ours)	**4 / 8 GB**	**12 GB**	**∼6m**

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

OVO-mapping runs at 0.5–1 keyframes per second on both Replica and ScanNetv2. 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. 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	172±98

Table 6. **OVO-mapping footprint.** GPU vRAM, checkpoint size, keyframes per second and number of keyframes.

Limitations. Despite achieving state-of-the-art results on 3D semantic segmentation, OVO's instance detection and tracking has margin for improvement. For example, a deeper integration between the semantic and SLAM modules is necessary to support loop closures. As demonstrated, OVO-mapping can be integrated into SLAM pipelines with real-time camera tracking. However, OVO operates at $0.5-1$ frames per second, introducing a slight delay in the semantics. Moreover, while CLIP merging generalizes better than the baselines, it still exhibits a slight bias toward the most frequent CLIPs and categories. A more robust training approach, involving a broader set of classes and incorporating semantic properties can further enhance its capabilities.

5. Conclusions

In this paper we present OVO, an open-vocabulary online 3D mapping method, 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. We develop 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 by a neural network, which we show to be more effective than alternative handcrafted approaches while maintaining generalization capabilities. We outperform relevant baselines in both computation and segmentation metrics on Replica and ScanNet. 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, 3)

[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 page: 6)

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. (see page: 2)

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

[14] Kamak Ebadi, Lukas Bernreiter, Harel Biggie, Gavin Catt, Yun Chang, Arghya Chatterjee, Christopher E Denniston, Simon-Pierre Deschênes, Kyle Harlow, Shehryar Khattak, et al. Present and Future of SLAM in Extreme Environments: The DARPA SubT Challenge. *IEEE Transactions on Robotics*, 2023. (see page: 1)

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

[26] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. (see pages: 1, 3)

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

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

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

[30] 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, 6, and 7)

[31] 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)

[32] 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)

[33] 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)

[34] 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)

[35] 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)

[36] 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)

[37] 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)

[38] 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)

[39] 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, 6, 7, and 8)

[40] 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)

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

[42] 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, and 6)

[43] 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)

[44] 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)

[45] 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)

[46] 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)

[47] 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)

[48] 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)

[49] 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)

[50] 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)

[51] 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)

[52] 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)

[53] 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)

[54] 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)

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

[56] 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)

[57] Muer Tie, Julong Wei, Ke Wu, Zhengjun Wang, Shanshuai Yuan, Kaizhao Zhang, Jie Jia, Jieru Zhao, Zhongxue Gan, and Wenchao Ding. O2v-mapping: Online open-vocabulary mapping with neural implicit representation. In *European Conference on Computer Vision*, pages 318–333. Springer, 2024. (see pages: 1, 2, 6, and 8)

[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, 4, 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, 6, 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, Wenchao 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 pages: 2, 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 labels. 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 set with 20 classes (ScanNet20), and an expanded set with 200 classes (ScanNet200) [50]. 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 ScanNet 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

Our CLIP merging has a 5-layer transformer encoder with 8 heads and a 4-layer 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 the training process.

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-l 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, 42, 56, 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 with SigLIP ViT-SO400M rather than its base CLIP ViT-L/14. We reproduced O_2V in Replica using CLIP ViT/B-16 due to crashing with Out Of Memory errors when using bigger backbones like CLIP ViT/H-14 or SigLIP ViT-SO400M. We where unable to make O_2V and OpenNerf converge in ScanNetv2, probably due to the impact of its noisy GT camera poses in NeRFs convergence. We report OpenNeRF official metrics on Replica.

B. System ablations

In this section we report minor ablations and experiments performed during OVO's 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 CLIP merging. We profit from the CLIP merging 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 [46] 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 CLIP merging 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

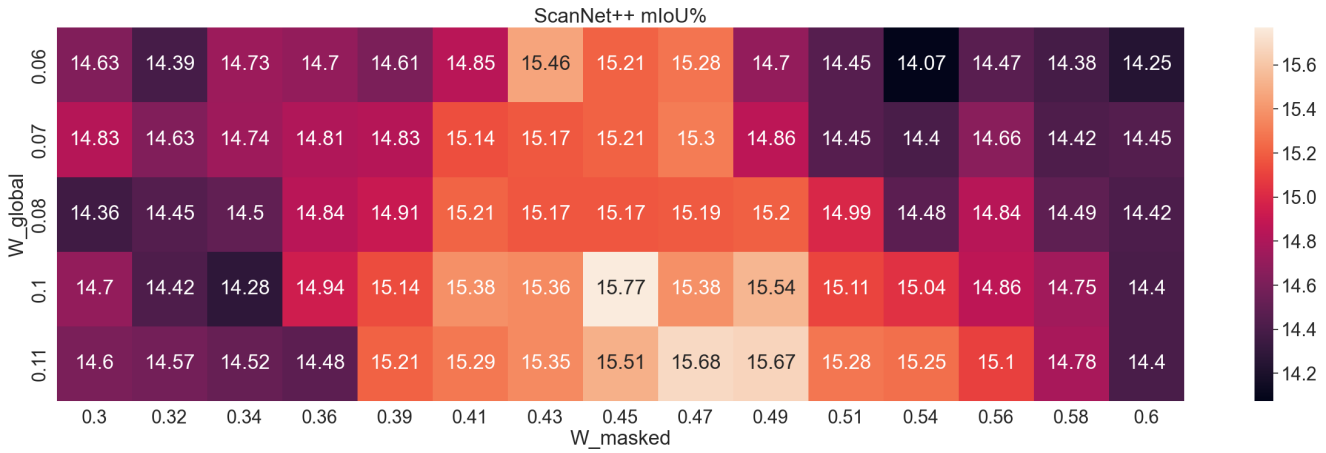


Figure 8. Grid search for CLIP weights merging on five scenes from ScanNet++ [64].

SAM 2 [25]; 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 semantic classes are matched to each 3D object using the similarity approach presented by LangSplat [43].

B.1. Foundation Models

SAM Since its release, Segment Anything Model (SAM) [25] has been the state-of-the art for out-of-the box instance segmentation on different fields. Its 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 duplicated 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 [25], SAM 2 [47], 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.

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 [52] 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 bal-

	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 [25]	19.0	5.43 ± 1.83
SAM [25] - tuned	18.1	0.84 ± 0.13
SAM 2 [47] - tuned	**19.1**	0.71 ± 0.10

Table 7. Segmentation backbone ablation.

Architecture	Resolution	mIoU [%]	Latency [s]
DFN-ViT-B-16	224×224	10.92	$\mathbf{0.100 \pm 0.022}$
DFN-ViT-L/14		11.89	0.173 ± 0.031
DFN-ViT-H/14		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		12.96	0.664 ± 0.136

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

ance 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 [46] 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 $\text{Sigmoid}(\cos(\phi_{\text{qry}}, \phi_{\text{img}}) \times \tau + b)$, 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

$$\min_i \frac{\exp\left(\cos\left(\phi_{\text{qry}}, \phi_{\text{img}}\right)\right)}{\exp\left(\cos\left(\phi_{\text{qry}}, \phi_{\text{img}}\right)\right) + \exp\left(\cos\left(\phi^i_{\text{can}}, \phi_{\text{img}}\right)\right)}, \tag{2}$$

where ϕ^i_{can} 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 CLIP descriptors to elements in an image, we follow HOV-SG's [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

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

with

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

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

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

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.

CLIP merging Rather than relying on 3 fixed-weights that ideally should be tunned for each scene, we developed the CLIP merging 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 CLIP merging outperforms the baselines using all the labels, Tab. 10. Excluding from the metrics the the 100 labels seen during training, we can observe how the CLIP merging 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 CLIP merging's generalization. **From here on, all experiments will proceed using the CLIP merging.**

B.3. Additional heuristics

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

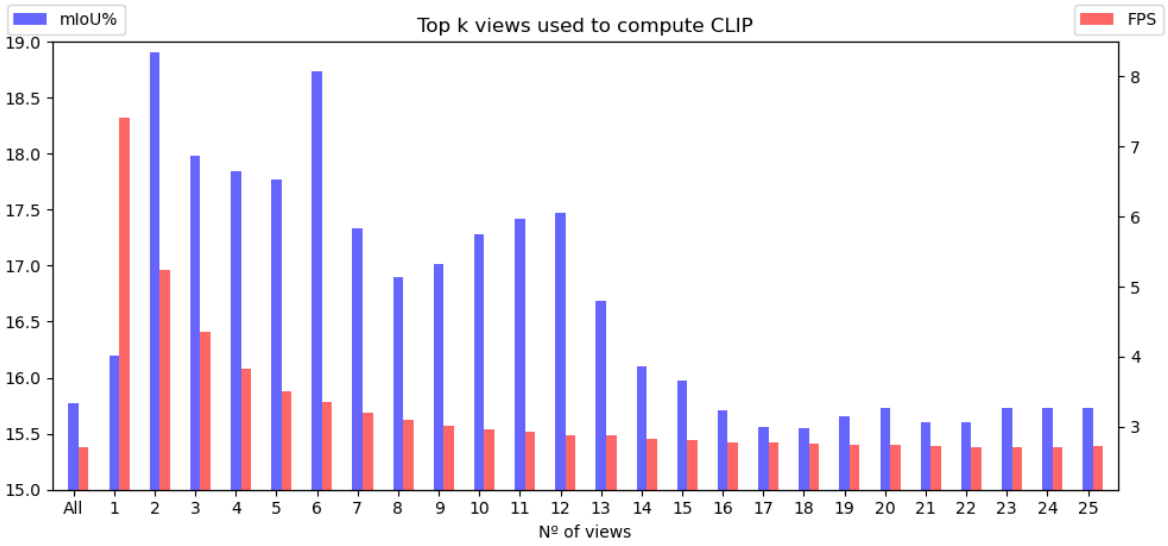


Figure 9. **Evaluation using only top views to compute CLIP on 5 scenes from ScanNet++ [64].** While using more than one view has substantial impact on the runtime, it also improves segmentation accuracy. However, too many views also degrade the segmentation accuracy.

	S++ w. top 100				S++ w/o. top100				ADE20k–150			
Method	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 CLIP merging 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. **CLIP merging 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 $k \in \{1, \ldots, 51\}$, 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 observations, 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 to the depth map to mask the edges of the objects and avoid matching points around them.

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).

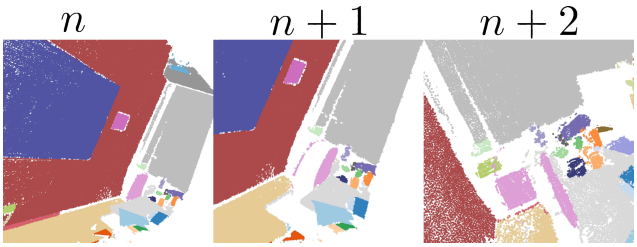


Figure 10. **Mask bleeding** and propagation produced by masks inaccuracy. The edges of the chair (pink) bleed to the background at k_n, and therefore the segment label is wrongly propagated to the it in the following keyframes.

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].

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 [25] with ViT-H/14 encoder (1-H), and SAM 2.1 [47] 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 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.