OVI-MAP: Open-Vocabulary Instance-Semantic Mapping

Supplementary Material

6. Additional Details of the Method

6.1. Super-Point Merging

In the main paper, we addressed under-segmentation in 2D instances. Over-segmentation, however, is handled by merging spatially proximate super-points. Here, we provide details on the merging procedure.

For all voxels containing 3D points in the point cloud $P_{t,j}$, let $\Omega_{j,k}$ denote the number of voxels assigned to the super-point S_k with label k. The spatial proximity $Spa(S_a, S_b)$ measures the overlap between two super-points S_a and S_b by counting how many point clouds significantly intersect both:

$$Spa(S_a, S_b) = \sum_{P_j \in \mathcal{P}} \mathbb{I}[\Omega_{j,a} > \theta_{\text{assoc}} \wedge \Omega_{j,b} > \theta_{\text{assoc}}],$$

where $\mathcal{P} = \{P_{t,j}\}_{j=1}^{N_t}, \forall t \in T$ is the set of all inserted point clouds, and $\mathbb{I}[\cdot]$ is the Iverson bracket which is one of the condition inside holds and zero otherwise. Super-points with overlap $Spa(S_a, S_b)$ exceeding θ_{merge} are considered spatially connected, and merged into a single instance with same label.

6.2. TSDF Map Projection via Ray-Casting

After the incremental TSDF fusion and label stabilization with super-points voting and merging, we obtain a global TSDF-based instance map. We leverage this global instance map with stabilized instance labels across frames for the view-selection and feature extraction.

We obtain the projection of the TSDF map within the current camera frame by casting ray going through each pixel to the TSDF voxels. Combining with the depth prior from the depth input for ray-casting, we get a globally aligned 2D instance mask that taking occlusion and multi-view consistency into account.

7. Ablation Study on the VLM Backbone

Table 7 compares several VLM backbones for semantic feature extraction, including CLIP and multiple SigLIP variants. Results show that the view selection strategy remains effective across different models, indicating that our method is not specific to a single VLM. We observe that larger SigLIP variants [60] yield consistent improvements in mIoU and AP metrics, with moderate parameter growth. Our method uses `siglip-large-patch16-384`, which offers the best trade-off between accuracy and computational cost.

Method	mIoU	mAcc	AP_{25}	AP_{50}	AP_{all}	Param.
clip-vit-large-patch14-336	15.4	24.7	20.6	14.9	6.7	0.4B
siglip-large-patch16-384 (Ours)	**26.9**	33.2	**36.4**	22.0	8.6	0.7B
siglip-so400m-patch14-384	25.2	33.8	32.4	19.1	9.7	0.9B
siglip2-large-patch16-384	26.7	33.9	35.9	22.2	**10.4**	0.9B
siglip2-so400m-patch14-384	26.4	**34.3**	**36.4**	**23.1**	9.4	1B

Table 7. **Semantic feature extraction by** (1) ViT-L, (2) siglip-large (Ours), (3) siglip-so400m, (4) siglip2-large, and (5) siglip2-so400m. The last column shows the number of parameters for each model.

Method	Component	Run-Time / Frame	Thread	Skipped Frames
Ours	RGB Segmentation	964.8 ms	1	30
	Depth Segmentation	88.4 ms	2	
	2D-3D Association	76.3 ms		
	View Selection	140.8 ms		10
	Feature Extraction	131.3 ms	3	
OVO-SLAM	Segmentation	1441.8 ms	1	50
	Mapping	168.8 ms		
	Feature Extraction	433.4 ms	2	

Table 8. **Runtime breakdown.** All components run in parallel across dedicated threads. These steps are performed only on keyframes, meaning only every n-th frame is processed to match real-time performance at 30 FPS. The number of skipped frames for each component are listed in the last column. This pipeline design keeps the overall system real-time despite potentially expensive individual modules.

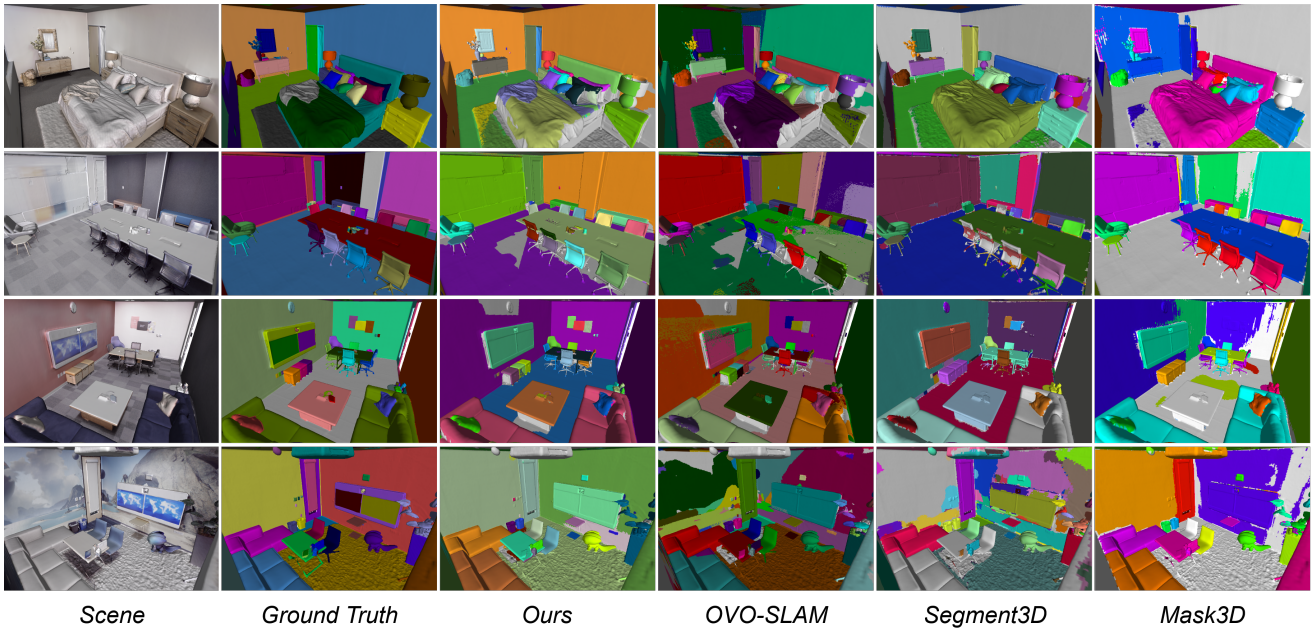
8. Runtime

Table 8 reports the per-frame runtime of the main components, measured on an Nvidia RTX3090 GPU and an Intel Core i7-12700K CPU. The system operates as a multi-threaded pipeline, where segmentation, 2D-3D association, view selection, and semantic feature extraction run in parallel on separate CPU and GPU threads. The 2D-3D association includes 3D lifting of the 2D segments, update of the super-point map, and obtaining the global instance map via ray-casting.

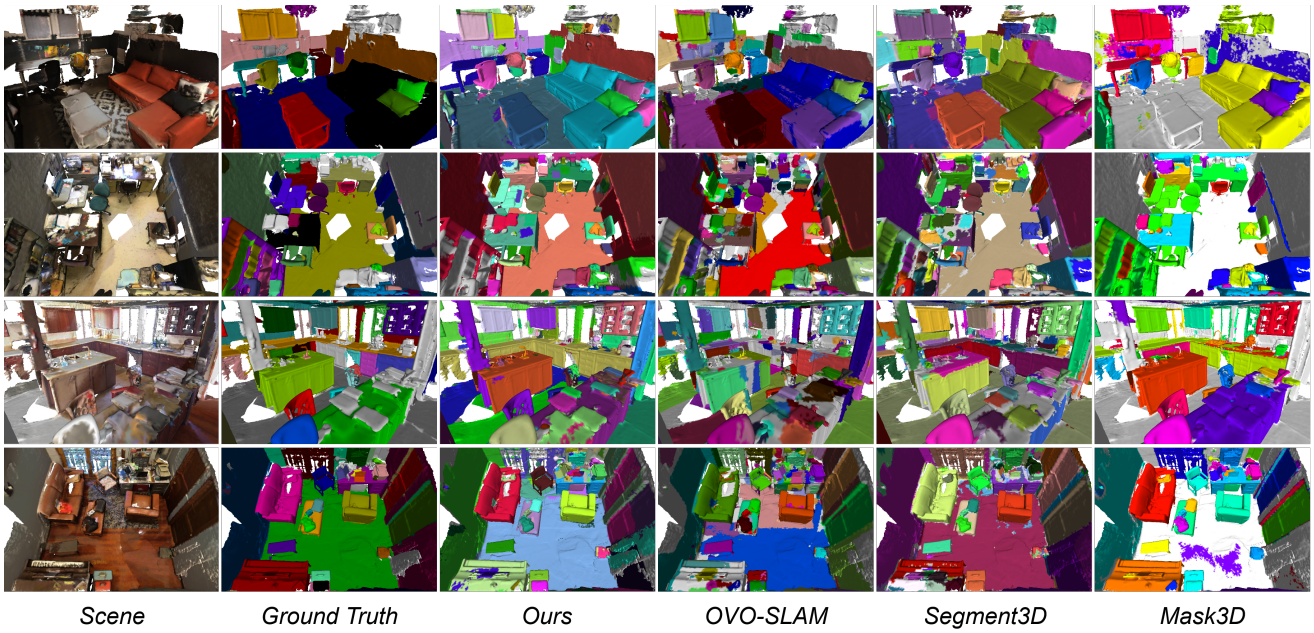
These instance/semantic segmentation steps are executed only on every *n* frames as reported in the main paper rather than every incoming frame, which substantially reduces overall latency.

9. Visualizations of Instance Maps

Figure 7 compares instance maps created by different methods. Online methods (Ours, OVO-SLAM) operate incrementally, while offline methods (Segment3D, Mask3D) process complete meshes. Colors correspond to instance IDs; gray denotes unobserved (for online methods) or unlabeled areas (for offline methods). Our method maintains consistent instance boundaries and achieves dense scene coverage, while offline methods leave large unlabeled regions despite full-scene access.



(a) Qualitative comparison of instance maps on the Replica dataset.



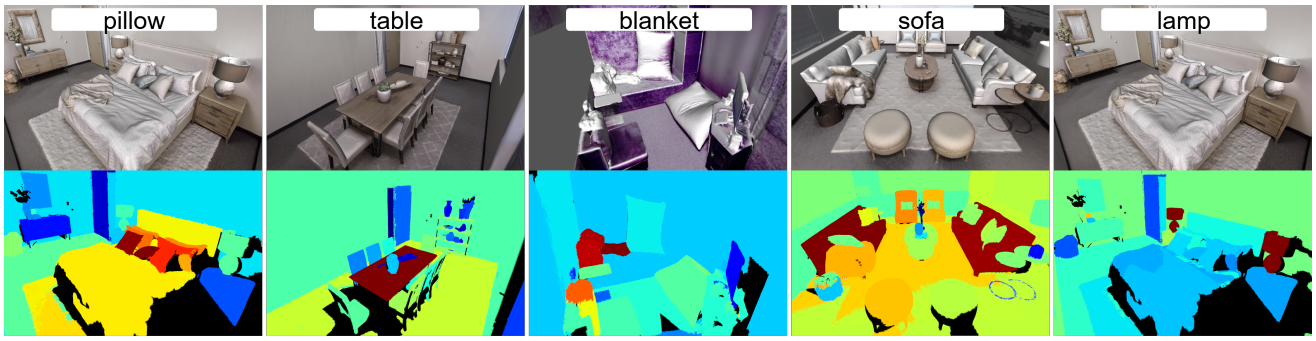
(b) Qualitative comparison of instance maps on the ScanNet dataset.

Figure 7. We compare our method with online [29] and offline [18, 43] approaches on the ***Replica*** (a) and ***ScanNet*** (b) datasets. Colors are randomly assigned for all instance maps according to the instance labels. Gray regions indicate unobserved areas for online methods (Ours and OVO-SLAM), and unlabeled for offline methods (Segment3D, Mask3D). OVI-MAP produces spatially coherent accurate reconstructions, maintaining sharp instance boundaries throughout incremental mapping.

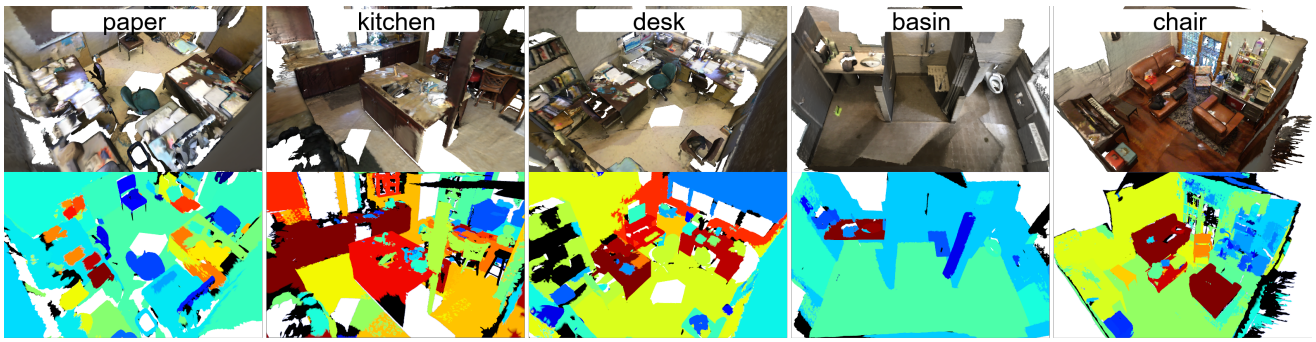
10. Heat Maps

To better understand how the objects are recognized, the heat maps of the instances in the scene are shown in Fig.8. It shows us how well can the objects we query can be distinguished from the others. We calculate the cosine similarities between the semantic features of all instances and the semantic features of the query semantic label, then map the normalized similarities to a color map.

The visualizations using heat maps further demonstrate



(a) Heat maps for semantic querying to the scenes from the Replica dataset.



(b) Heat maps for semantic querying to the scenes from the ScanNet dataset.

Figure 8. **Heat map visualizations of the semantic queries.** The color closer to red indicates the instance is more similar to the query semantic label, while the color closer to blue indicates the instance is less similar to the query semantic label. Unobserved areas are shown in black.

the effectiveness of the proposed method in recognizing the objects in the scene, where the objects that match the query label are highlighted correctly, and other irrelevant objects are not highlighted as much. This application shows the potential of the proposed method in real-world scenarios. For example, the system can locate the objects we are looking for based on the heat map, and update the semantic map accordingly in those areas.

11. Datasets

We perform experiments on the Replica dataset [45] and ScanNet [7], which are widely used benchmarks for 3D scene understanding tasks.

We use 8 scenes including *'office0', 'office1', 'office2', 'office3', 'office4', 'room0', 'room1', 'room2'* for evaluation on Replica, similar to OpenNeRF [10]. We evaluated on 18 scenes for ScanNet, as selected by ConceptFusion [19].