2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

FindAnything: Open-Vocabulary and Object-Centric Mapping for Robot Exploration in Any Environment

Sebastián Barbas Laina[1,5,*], Simon Boche[1,*], Sotiris Papatheodorou[1,3,4,5,6,*], Simon Schaefer[1], Jaehyung Jung[1], Helen Oleynikova[2], Stefan Leutenegger[1,2,3,5,6]

***Abstract*— Geometrically accurate and semantically expressive map representations have proven invaluable for robot deployment and task planning in unknown environments. Nevertheless, real-time, open-vocabulary semantic understanding of large-scale unknown environments still presents open challenges, mainly due to computational requirements. In this paper we present *FindAnything*, an open-world mapping framework that incorporates vision-language information into dense volumetric submaps. Thanks to the use of vision-language features, *FindAnything* combines pure geometric and open-vocabulary semantic information for a higher level of understanding. It proposes an efficient storage of open-vocabulary information through the aggregation of features at the object level. Pixel-wise vision-language features are aggregated based on eSAM segments, which are in turn integrated into object-centric volumetric submaps, providing a mapping from open-vocabulary queries to 3D geometry that is scalable also in terms of memory usage. We demonstrate that *FindAnything* performs on par with the state-of-the-art in terms of semantic accuracy while being substantially faster and more memory-efficient, allowing its deployment in large-scale environments and on resource-constrained devices, such as MAVs. We show that the real-time capabilities of *FindAnything* make it useful for downstream tasks, such as autonomous MAV exploration in a simulated Search and Rescue scenario.**
Project Page: https://ethz-mrl.github.io/findanything/.

I. Introduction

One key application of robotics is Search and Rescue (S&R) and disaster response: where robots should take the place of humans in hazardous or inaccessible environments, and provide useful, safety-critical information to first responders. Micro Aerial Vehicles (MAVs) are particularly relevant here, as they move freely in 3D space and can easily access areas out of reach of humans and ground-based robots.

A robot that is useful for emergency situations requires safe operation with minimal human supervision, while giving remote operators live, up-to-date information about its environment: not just geometry and appearance, but also higher-level semantic information and object locations.

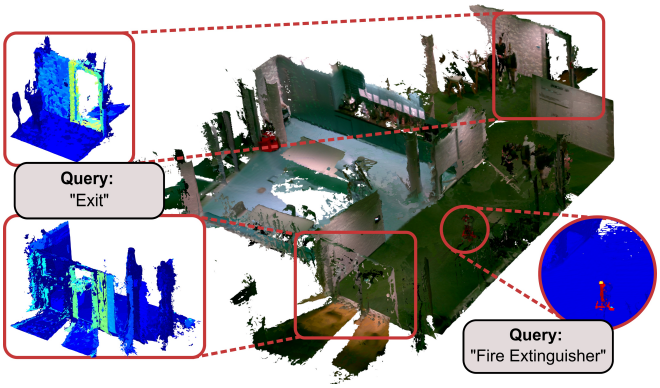

Fig. 1. Real-world MAV exploration demonstration of *FindAnything* in an office environment, including a kitchen. Colored meshes extracted from volumetric maps are shown alongside the 3D CLIP activations in our object-centric map representation for the language queries *"fire extinguisher"* and *"exit"*, which are potential points of interest in a fire scenario.

We propose *FindAnything*, a mapping system that runs online on-board an MAV to address these requirements: creating a scalable, submap-based volumetric map that includes the scene geometry, appearance, and also encodes *open-vocabulary* features in an object-centric way, allowing operators to query the map live for locations of rooms, objects, or properties of the scene.

For instance, consider a fire emergency, where a robot might be assigned to locate objects or tools useful for firefighting, e.g. fire extinguishers, or the nearest exit (see Fig. 1). To succeed in this demanding and highly time-sensitive task, the robot needs the capability to incrementally reconstruct the unknown scene online and in real-time while safely navigating through it. To ensure scalability to large environments (e.g. multistory houses), the underlying map representation must be memory-efficient for deployment on resource-constrained platforms, such as MAVs. Finally, the described task also requires advanced scene understanding capabilities that go beyond purely geometric reasoning but also incorporate generalizable semantic reasoning without relying on prior knowledge of the scene.

Volumetric maps are a common scene representation for MAVs, as they are useful for both on-board navigation and as a map interpretable by robot operators [1], [2]. To go beyond

This work was supported by the Technical University of Munich, MIRMI, a donation by Google, the State of Bavaria through the REACT project, the TUM Innovation Network CoConstruct, ETH Zurich, and the EU Horizon projects DigiForest (101070405) and AUTOASSESS (101120732).

[1]Mobile Robotics Lab, School of Computation, Information and Technology, Technical University of Munich. E-mail addresses: {sebastian.barbas, simon.boche, sotiris.papatheodorou, simon.k.schaefer, jaehyung.jung, stefan.leutenegger}@tum.de
[2]Mobile Robotics Lab, Department of Mechanical and Process Engineering, ETH Zürich. E-mail address: {lestefan, oelena}@ethz.ch
[3]Department of Computing, Imperial College London. E-mail address: s.leutenegger@ic.ac.uk
[4]Department of Electrical and Computer Engineering, University of Patras. E-mail address: s.papatheodorou@ac.upatras.gr
[5]Munich Institute of Robotics and Machine Intelligence (MIRMI).
[6]Munich Center for Machine Learning (MCML).
* Equal contribution.

mere geometry and appearance and enable a higher level of scene understanding, many approaches have been studied to integrate class-level semantic information into volumetric maps [3], [4], [5]. However, these methods require an a priori known set of classes, making the semantic information compact to store (as the number of classes is finite), but limiting the expressiveness and applicability of the maps.

Vision-Language (VL) models such as CLIP [6] do not have these restrictions and instead store semantic information as a high-dimensional *feature embedding* that can be queried in real time for matching a human language description. While this gives such models incredible flexibility, it also comes with a computational and memory cost: where a class label can be stored as a single integer, feature embeddings are generally hundreds of floating-point values. This makes them challenging to aggregate into a 3D map on resource-constrained hardware. To address this, recent works have explored more compact feature representations [7] or combined VL and segmentation foundation models [8], [9], [10], such as *SegmentAnything* (SAM) [11] or more efficient variants of it, e.g. *EfficientSAM* (eSAM) [12]. However, most of these approaches still fall short of the computational requirements to build large-scale maps online and on-board an MAV.

We propose *FindAnything*, a system for large-scale volumetric mapping with open-vocabulary semantic capabilities. Dividing the volume into smaller submaps allows for drift-correcting mechanisms such as loop closures, enabling the deployment in large-scale environments where some degree of state estimation drift is inevitable. Our method aggregates open-vocabulary information over time at the level of objects or object parts. Oversegmentation of objects into smaller entities allows for fine-grained queries, while the generalization capabilities of VL features preserve an understanding of larger objects or concepts. The object-centric map representation significantly reduces the memory requirements while allowing the use of foundation models that trade accuracy for speed. We show that these characteristics make *FindAnything* suitable for downstream tasks, such as the deployment of an MAV for autonomous exploration in a simulated fire emergency. The main contributions of this paper are:

- A method to aggregate high-dimensional VL features into a volumetric map in a memory-efficient, object-centric way: using image-based semantic oversegmentation, segment tracking and association, and feature embedding merging.
- An integration of the proposed object-centric VL feature mapping approach with a submapping-based visual(-inertial) SLAM system, enabling large-scale, online, compute- and memory-efficient mapping even on resource-constrained platforms.
- Evaluation in simulation and real-world benchmarks showing that *FindAnything* achieves semantic accuracy competitive to the state-of-the-art while requiring shorter computational times and up to 60% less memory.
- A showcase of a downstream exploration task on-board an MAV, demonstrating that *FindAnything* can be used to help guide robot exploration using natural language.

II. Related Work

A. Foundation Models

Foundation models, trained on internet-scale data, are a cornerstone for tackling a wide range of tasks in a zero-shot manner. Models like CLIP [6], LLaVA [13], and LSeg [14] enable reasoning about images in natural language by encoding visual and textual information in a single feature space.

While these models were trained using contrastive learning on image patches [6], significant advancements have been made in precisely localizing features in images [15], [16], [14]. However, challenges remain in accurately representing complex object shapes and long-tailed object categories [10].

In this work, we address these limitations by integrating the VL foundation model CLIP [6] with the general-purpose segmentation model eSAM [12]. By accumulating extracted features at the object (or part) level within our probabilistic mapping framework, we effectively harness the extensive open-vocabulary capabilities of the VL model without relying on its ability to produce highly precise feature maps. This approach allows for robust semantic understanding, even for complex and diverse object types.

B. Open-Vocabulary-based Mapping & Navigation

Recent works have explored extending 2D foundation models into 3D without relying on internet-scale 3D datasets. Some methods embed VL features into NeRFs [17], [18] or Gaussian splats [19]. While these approaches generate photorealistic renders of the environment, they require time-consuming training before deployment, limiting their scalability to large environments or their use onboard robots.

To address this limitation, [20] proposed a method to fuse VL features into a monolithic TSDF map, allowing for efficient updates and queries but remaining constrained in map size due to the memory requirements. ConceptFusion [10] and related works [9], [21] directly store features in a point cloud, combining region-level and global embeddings. [22], [23] instead integrate features into a 2D top-down map. More recently, several methods [8], [9], [24], [25], [26] focus on accumulating VL features at the instance level using 3D scene graphs or instance-level point clouds. While these approaches offer improved scalability, they either do not preserve full 3D geometric information, essential for downstream tasks, or are prone to map errors as they do not allow for drift-correcting mechanisms such as loop closures.

In contrast, RayFronts [7] proposes an open-set semantic 3D mapping framework for outdoors robotic exploration. To enhance efficiency and semantic accuracy, they propose a fine-grained dense vision-language encoder based on RADIO [27]. Nevertheless, fusing VL features at the voxel level into a VDB-based occupancy map results in high memory usage, limiting its applicability to larger environments.

FindAnything aims to combine the best of both worlds. Similar to [8], we use segmentation foundation models. However, instead of scene graphs or point clouds we utilize instance-level volumetric occupancy maps with adaptive resolution. Also, instead of tracking in 3D, our segment

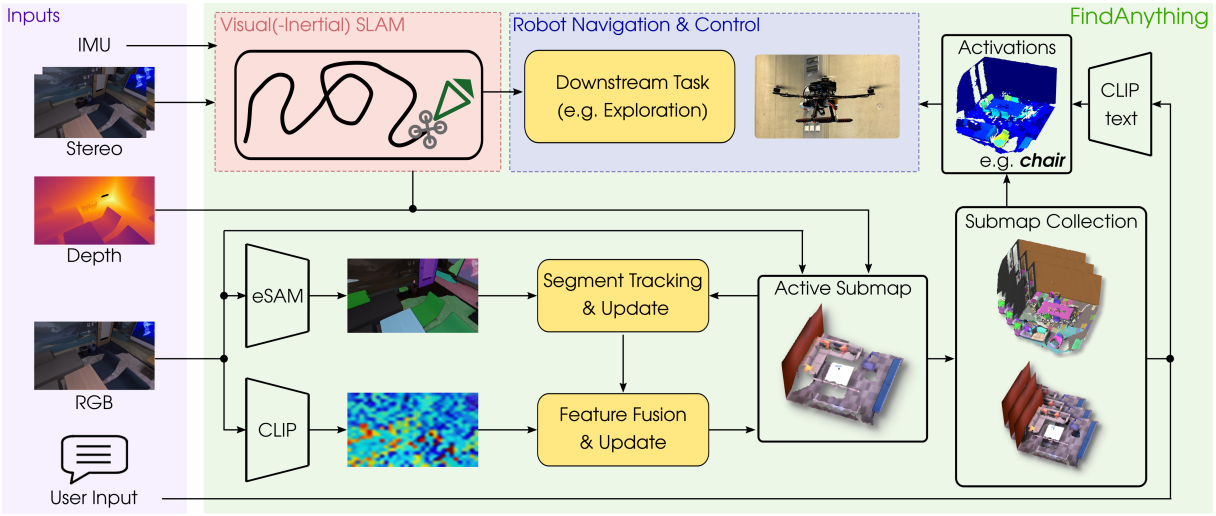

Fig. 2. Overview of *FindAnything*: VI-SLAM provides estimated poses to integrate depth and RGB images into volumetric occupancy submaps. eSAM [12] creates object proposals from the RGB images which are tracked against the current map. CLIP [6] features are aggregated per object mask and fused into the current submap. The submaps can be directly used in downstream tasks, such as autonomous exploration. In addition, natural language user input is translated into a CLIP feature to find regions of high activation in the map and guide the exploration towards those areas of interest.

tracking is done in image space by projecting map objects into the image plane. This alleviates volumetric discrepancies between an object and its current view. This approach enables accumulating VL features in real-time, efficient natural language queries, and error corrections by coupling mapping with the Simultaneous Localisation and Mapping (SLAM) pipeline. Aggregating features at the instance-level significantly reduces memory usage and improves scalability. Unlike prior works, we show that our system runs completely online, even on resource-constrained devices. By preserving detailed 3D geometric as well as object-level semantic information, our method overcomes the limitations of prior works and is well-suited for large-scale environments, and applications such as exploration.

III. Methodology

In this section, we present the modules that form our system, a schematic overview of which is presented in Fig. 2.

A. Notation and Definitions

The used Visual-Inertial SLAM (VI-SLAM) system tracks a moving body with a mounted IMU and several cameras relative to a static world coordinate frame $\mathcal{F}_W$. The robot pose, expressed in the IMU frame $\mathcal{F}_S$, is denoted as $\boldsymbol{T}_{WS}$. Images are presented in matrix form, $\mathbf{I}$, and we obtain values at a pixel coordinate $\mathbf{u}$ (or pixel range as mask) via $\mathbf{I}[\mathbf{u}]$.

B. VI-SLAM

Our state estimation module is based on the multi-sensor SLAM system OKVIS2-X [28], an extension of *OKVIS2* [29], that incorporates depth information to improve the state-estimation accuracy. SLAM poses are used to integrate depth information into volumetric occupancy submaps. A new state $\mathbf{x}_l$ is estimated for a new pair of stereo images at timestamp l. The state-estimator performs real-time sliding window and loop closure optimization. As a result, for a new pair of frames, the state $\mathbf{x}_l$ is estimated and prior states $\mathbf{x}_{l-k}$ are updated as necessary, including loop closures.

C. Volumetric Occupancy Mapping

The geometric information is represented by partitioning the environment into volumetric submaps using the *supereight2* [30] mapping framework. *Supereight2*'s occupancy map representation allows differentiating between occupied, free and unmapped regions, making it directly usable for safe path planning and navigation. By employing submaps, we achieve scalability to large environments while using shallower data structures for mapping, enabling faster data access. We track the rigid transformations between submaps and update them after each SLAM optimization, including loop closures, following the scheme proposed in [31], where each submap is associated to a keyframe.

D. Vision-Language Feature Fusion

As a VL feature extractor, we use CLIP [6] ViT-L/14 pre-trained for images of 336×336 pixels. Following [32], we obtain an image $\mathbf{F}$ with a 768-dimensional feature embedding per pixel. Aggregating these features into volumetric submaps at a voxel level would lead to vast memory requirements, limiting the scalability of the system to larger environments. Instead, we use an object-centric approach, in which segments obtained from image-based segmentation are associated to voxels. For subsequent frames, we perform an image-to-map segment tracking. This choice allows us to decouple the voxel resolution from the language representation, enabling mapping at high resolutions without significantly increasing memory usage, a limiting factor on resource-constrained robots.

We use the eSAM [12] foundation model, a lightweight SAM [11] version with faster inference time, to obtain a collection of binary segment masks $\mathscr{E}_l \in \{\mathbf{E}_0, \mathbf{E}_1, ..., \mathbf{E}_m\}$. As the agent moves through the environment, we track or split the

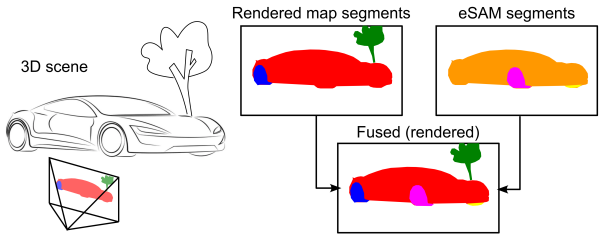

Fig. 3. Schematic of our as-fine-as-possible segmentation strategy. A segment image of the map is rendered from the current camera pose. Rendered segments and eSAM segment proposals are fused into the map favoring the smallest option available.

objects that have been previously added to the submap while adding new segments to previously unsegmented regions.

To construct an object-centric map representation, we extend *supereight2* by incorporating segment IDs into occupied voxels. Each segment ID, referred to as k, stores an associated average language feature $\bar{\mathbf{f}}_k$ encoding open-vocabulary semantic information, and the total number of pixels N_k that have been used to update this segment. To update the segment IDs of a submap at time step t, we require a depth image $\mathbf{D}_t$, the robot pose at that time $\boldsymbol{T}_{WS_t}$ and a corresponding image with per-pixel segment IDs $\mathbf{K}_t$. To track objects, we also render a collection of binary segment masks and associated segment IDs, $\mathscr{R}_t \in \{(k_0, \mathbf{R}_0), (k_1, \mathbf{R}_1), ..., (k_p, \mathbf{R}_p)\}$, of the current submap into the image frame of given pose $\boldsymbol{T}_{WS_t}$.

We follow an as-fine-as-possible segmentation strategy to represent the objects in the smallest partition proposed by eSAM over time. Segment tracking and oversegmentation is based on 2D image overlap between eSAM segments $\mathscr{E}_t$ and rendered segments $\mathscr{R}_t$ from the current submap. Our method greedily partitions both segment sources, if possible, by prioritizing smaller segments (above a threshold), producing an image $\mathbf{K}_t$ with per-pixel segment IDs to be integrated into the current submap. A visual schematic of this approach is presented in Fig. 3. VL features are then aggregated per segment over all pixels with an ID in $\mathbf{K}_t$. This strategy allows for long-term tracking and is fully computed on a CPU, freeing the GPU resources for the foundation models. This oversegmentation of objects into smaller entities allows for fine-grained queries, while the VL features preserve a semantic understanding of larger objects or concepts, allowing to group similar segments upon query time. The assumption of semantic consistency, together with that of spatial consistency between submaps, renders segment tracking across maps unnecessary.

After segment tracking, a weighted average update is performed on the per-segment feature vectors using the per-pixel segment ID image $\mathbf{K}_t$ and the CLIP image $\mathbf{F}_t$. For each segment ID k in $\mathbf{K}_t$ we compute the set of pixel coordinates $\mathscr{N} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N\}$ where $\mathbf{K}_t$ has the value k. The VL feature is then updated as

$$\bar{\mathbf{f}}_k \leftarrow \frac{N_k \bar{\mathbf{f}}_k + \sum_{i=1}^{N} \mathbf{F}_t[\mathbf{u}_i]}{N_k + N}, \tag{1}$$

$$N_k \leftarrow N_k + N, \tag{2}$$

where N_k is the number of pixels previously associated with segment ID k, while N is the number of pixels in the tracked segment in the current frame. The average language feature descriptor of segment k is denoted as $\bar{\mathbf{f}}_k$. These weighted updates from different viewpoints improve the consistency of the VL embedding representation.

IV. Evaluation and Experiments

To showcase the accuracy and computational efficiency of our proposed system, we benchmark *FindAnything* on a number of standard datasets both in indoor (Section IV-B) and outdoor (Section IV-C) scenarios. As motivated by our application requirements, we focus on evaluating semantic map accuracy as well as runtime and memory usage.

We additionally showcase the usefulness of *FindAnything* in an integrated real-world experiment, where *FindAnything* is used as the mapping back-end for online exploration in a simulated firefighting scenario (see Section IV-E).

A. Evaluation Criteria

For the semantic accuracy of the aggregated VL features, we follow the standard evaluation setup of [7], [8], [10], [33] and report accuracy as the class-mean recall (mAcc) and the frequency-weighted mean intersection-over-union (f-mIOU) of closed-set predictions of the semantic classes.

We demonstrate *FindAnything*'s suitability for online deployment in resource-constrained devices by evaluating runtime of our system against different state-of-the-art open-vocabulary 3D mapping approaches on the Replica dataset [34], a typical indoor dataset, in Section IV-B, and additionally memory usage in SemanticKITTI [35], a large-scale outdoor dataset, in Section IV-C.

As a main baseline, we chose to benchmark against RayFronts [7], one of the few established approaches for large-scale volumetric open-vocabulary mapping. As RayFronts runs mainly on the GPU, we report memory as the combined GPU and RAM usage. For fair comparisons, we ensured equal voxel resolution and depth range limits and set a batch size of 1 to mimic online mapping. Furthermore, we disable their ray frontiers representation for mapping beyond depth-sensing range to limit the evaluation to the semantic accuracy and memory usage of the underlying volumetric map. Results for other papers are mainly taken from the respective papers. All results from competitors were obtained using ground-truth (GT) poses. For *FindAnything*, results are shown for both GT and SLAM poses to evaluate both ideal and realistic real-world performance.

We also evaluate *FindAnything*'s applicability for downstream tasks such as exploration, by evaluating mesh completion and accuracy of an exploration approach with and without using semantic information in Section IV-D.

B. Indoor Dataset: Replica

Our first evaluation seeks to answer multiple questions: 1) how does the accuracy of *FindAnything* compare to state-of-the-art semantic mapping approaches, 2) what impact does submapping versus monolithic mapping have in small-scale

TABLE I

3D SEMANTIC MAP ACCURACY (REPLICA).

	Method	mAcc	f-mIoU
GT	ConceptFusion[1] [10]	24.16	31.31
	ConceptFusion[1] [10] + SAM	31.53	38.70
	ConceptGraphs[1] [33]	40.63	35.95
	HOV-SG (Vit-H-14)[1] [9]	30.40	38.60
	Clio-batch[1] [8]	37.95	36.98
	OVO-SLAM† [24]	32.18	46.19
	Octree-Graph (OVSeg)[1] [26]	41.40	55.30
	RayFronts (NACLIP) [7]	26.44	32.66
	RayFronts (NARADIO) [7]	52.90	64.97
	FindAnything †	44.48	62.01
	FindAnything	48.87	62.71
	FindAnything (NARADIO)	**53.55**	**66.91**
SLAM	*FindAnything-Monolithic*	47.57	61.37
	FindAnything	48.80	62.91

[1] Results taken from the respective papers. Words in parentheses indicate VL encoders if different versions are available.
† Only processing every 10th frame.

TABLE II

MEAN REPLICA SEQUENCE PROCESSING TIME

Method	Mean time per sequence
OVO-SLAM† [24]	3m 2s
FindAnything †	**1m 19s**
HOV-SG [9]	11h 12m
Rayfronts [7]	9m 19s
FindAnything	**5m 24s**

HOV-SG timings taken from [24], which used an NVIDIA RTX 3090 GPU. The other timings are obtained using an NVIDIA RTX 4500.
† Only processing every 10th frame.

environments, and 3) how much faster is *FindAnything* in processing a complete sequence than existing approaches?

To answer these questions, we evaluate on the Replica [34] dataset, using the same pre-rendered sequences as [10]. To use Replica (a monocular dataset) in our stereo SLAM system, we also render a synthetic color image 6 cm to the right of the original one using Habitat-Sim [36]. In our experiments, the voxel resolution was set to 5 cm.

Results for the semantic accuracy are presented in Table I, where we demonstrate that our proposed approach is competitive to the state-of-the-art. In fact, thanks to the modular design of *FindAnything*, it can achieve the highest semantic accuracy when using the same VL encoder (NARADIO) as RayFronts. Despite the higher accuracy of NARADIO, we decided to use CLIP features due to the lower dimensionality of its embeddings, resulting in a lower memory footprint at the expense of minor degradation in semantic accuracy.

As a consequence of the high SLAM accuracy on such small-scale datasets, the difference in semantic accuracy between our method with GT or SLAM poses is negligible. It is important to note that there is a minor randomness in our results. Even with GT poses, the submap creation still depends on SLAM running in the background. Due to the higher global consistency of submapping in the presence of loop closures, there is still a minor improvement over *FindAnything-Monolithic*, a version using a monolithic map.

To demonstrate the real-time capabilities of our system, we show the average runtime per sequence in Table II and provide a detailed time distribution of the different stages of the method in Table IV. As seen in Table II, our method is substantially faster than [9] and faster than other competitors, e.g. OVO-SLAM [24] or RayFronts [7]. As OVO-SLAM only processes every tenth frame, we evaluate *FindAnything* with the same processing strategy and demonstrate that it is faster and semantically more accurate. The minor semantic degradation in this evaluation mainly is due to the limited motion in the dataset, which is not assured in other scenarios.

C. Large-scale outdoor dataset: Semantic KITTI

Our second evaluation mainly focuses on two objectives. First, we evaluate the scalability of *FindAnything* to large environments in terms of semantic map accuracy, processing time and memory usage and benchmark it against RayFronts. Second, we justify our system design through a series of ablation studies. These studies specifically investigate: (a) the choice of the segmentation model; (b) the proposed strategy for VL feature fusion and oversegmentation; and (c) the partitioning of the map into submaps in a real-world scenario using SLAM poses. Table III presents the semantic accuracy as well as the average sequence processing time T_{avg} and the average per-sequence memory usage M_{avg} on the SemanticKITTI dataset [35], a large-scale outdoor dataset. For semantic evaluation, we exclude classes labeled as "moving" or "other". Depth images were obtained by pre-processing stereo images with [37] and the maximum depth integration range is set to 10 m.

Due to the vast GPU memory requirements of RayFronts, quantitative benchmarking of the semantic accuracy was restricted to coarse voxel resolutions (0.5 m). While both approaches have similar runtimes, *FindAnything* uses only 40% of the memory thanks to the aggregation of VL features at the segment level. As *supereight2* sometimes fails to preserve fine-grained geometric structures (e.g. poles) at such a coarse resolution, RayFronts achieves a higher class-mean recall. However, as most classes are coarser structures (e.g. roads, sidewalks), *FindAnything* still yields a higher f-mIoU. In contrast to RayFronts, which fails at finer resolutions (0.1 m) due to insufficient GPU memory, *FindAnything* shows superior scalability and succeeds with a moderate memory usage at this resolution. The overhead in processing times is negligible, as our limitation is the image processing by the networks, while RayFronts shows an expected increase in runtime. Moreover, geometric details are better preserved at 0.1 m resolution, resulting also in a higher semantic accuracy, significantly outperforming RayFronts$_{@0.5\text{m}}$.

To assess the impact of the chosen segmentation model, we evaluate *FindAnything* using eSAM at half image resolution (FA - half-res) and replacing eSAM with SAM2 [38] (FA - SAM2). While there is a big impact on runtime and memory usage, the semantic map accuracy stays largely unaffected. This is mainly because the different pixel accuracy of image segments is nullified when aggregating segments into the volumetric map for tracking. A higher map resolution is

TABLE III

3D SEMANTIC MAP ACCURACY AND RESOURCE USAGE (KITTI).

		mAcc	f-mIoU	T_{avg}[s]	M_{avg}[GB]
GT	RayFronts$_{@0.5m}$	**45.85**	35.60	**277.9**	24.61
	FindAnything $_{@0.5m}$	36.89	**51.48**	288.2	**9.91**
	RayFronts[1]$_{@0.1m}$	-	-	337.8[2]	>24.5
	***FindAnything* (FA)**	**51.24**	53.90	289.4	16.23
	FA - SAM2	49.06	**54.28**	359.4	18.22
	FA - half-res	50.59	53.29	**124.5**	**15.11**
	FA - no Fusion	49.88	50.26	286.3	16.24
SLAM	*FindAnything-Monolithic*	23.81	28.24	**285.7**	**15.58**
	FindAnything	**32.26**	**37.77**	292.5	16.09

[1] RayFronts$_{@0.1m}$ failed on all sequences due to insufficient GPU memory (24.5 GB available).

[2] RayFronts$_{@0.1m}$ timings obtained by extrapolating the average processing time until the time of failure.

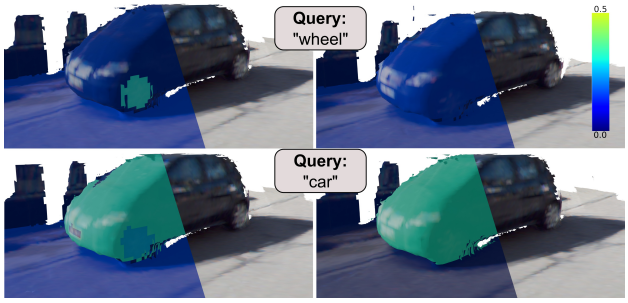

Fig. 4. Qualitative example of the benefit of our oversegmentation approach. The left column shows the activations with our oversegmentation approach and the right column shows the activations for a method that never oversegments the objects in the map. The top row shows the activations for the query "wheel" and the bottom row for the query "car".

needed to benefit from more accurate segmentation.

Furthermore, the effectiveness of proposed feature fusion and oversegmentation strategy is demonstrated. Table III shows that the proposed weighted-mean feature embedding update (Eq. (1)) from different viewpoints improves semantic accuracy compared to storing the first feature vector obtained for each segment (FA - no fusion). Figure 4 qualitatively shows the benefit of the as-fine-as-possible segmentation strategy. Oversegmentation allows distinguishing several parts of an object (e.g. "wheel") while maintaining an understanding of larger concepts or entities (e.g. "car").

These ablations show that the proposed configuration of *FindAnything* offers the best balance between semantic accuracy, runtime, and memory usage.

Finally, we evaluate *FindAnything* with SLAM instead of GT poses. We compute the $\mathrm{Sim}(3)$ alignment between the estimated and ground-truth trajectory and apply it to the maps before evaluation. Semantic accuracy is degraded by drift, which is inevitable in such large-scale scenarios. However, the comparison against *FindAnything-Monolithic* proves the necessity for submaps under the presence of drift, as transforming them based on drift-correcting mechanisms (e.g. loop-closures) removes map inconsistencies.

D. Application: Autonomous MAV Exploration

To demonstrate the usefulness of our mapping approach to downstream tasks we integrate it into an autonomous MAV exploration pipeline. By using the map's VL features to guide the exploration, we achieve high reconstruction accuracy of items or areas of interest that are selected online by natural language queries. Thanks to our mapping method, the resulting exploration pipeline both works in 3D and can respond to on-the-fly queries, while other exploration approaches supporting open-vocabulary queries work only in 2D or require a priori known queries [22], [39], [40], making them unsuitable for 3D exploration.

We extend the sampling-based exploration planner for submaps proposed in [41] to account for our object-level language-based scene representation. In [41], $n_c \in \mathbb{N}^+$ candidate next views are sampled close to frontiers and ranked using an information-gain-over-time utility function. The view with the highest utility is selected as the next goal and the process is repeated once the goal is reached.

The first extension over [41] consists of modifying the candidate next view sampling to take segments into account. Each time a new goal view needs to be selected, the cosine similarity between the embedding $\mathbf{f}_{\text{q}}$ of the current natural language query and the language embedding $\bar{\mathbf{f}}_k$ of each known segment is computed, selecting segments with a similarity greater than a predefined threshold $\beta \in \mathbb{R}^+$. Each segment is associated with a $1\,\text{m}^3$ cube centered on the segment's volumetric centroid. Since objects can be either over-segmented or represented in several submaps, a 3D non-maximum suppression is performed over the cubes, further reducing the number of segments considered. Candidate next views are then sampled inside the segment cubes. Sampling inside the segment cubes stops once n_c candidates have been accepted or if $3n_c$ candidates have been sampled, with the remaining ones being sampled close to frontiers as in [41].

The second modification of [41] is with regards to the utility function used to rank candidate next views. The utility of each candidate j, as computed by [41], is weighted with

$$w_j = \begin{cases} \dfrac{\bar{\mathbf{f}}_k \cdot \mathbf{f}_{\text{q}}}{\|\bar{\mathbf{f}}_k\| \|\mathbf{f}_{\text{q}}\|} & \text{if } \dfrac{\bar{\mathbf{f}}_k \cdot \mathbf{f}_{\text{q}}}{\|\bar{\mathbf{f}}_k\| \|\mathbf{f}_{\text{q}}\|} \geq \beta \\ \dfrac{\beta}{2} & \text{otherwise,} \end{cases} \tag{3}$$

where $\bar{\mathbf{f}}_k$ is the VL embedding of segment k if the candidate is inside its cube or $\bar{\mathbf{f}}_k = \mathbf{0}$ if the candidate is outside all segment cubes, and $\mathbf{f}_{\text{q}}$ is the VL embedding of the query.

Quantitative results are obtained in the `00848-ziup5kvtCCR` scene of the Habitat-Matterport 3D dataset [42], using the semantic annotations from [43] to obtain ground-truth meshes. The MAV platform is simulated using Gazebo [44]. Exploration is evaluated using two natural language queries: the item of interest "bed" and the room "bathroom". These queries were chosen because: i) the Habitat-Matterport 3D dataset [42] contains only house scenes, limiting the potential queries; ii) they appear more than once in the scene; iii) they are contained in the

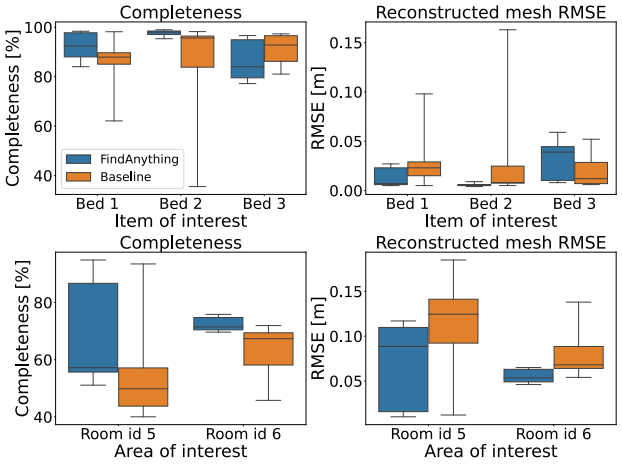

Fig. 5. Mesh completeness (left column) and reconstruction RMSE (right column) results of the 10th-90th percentiles for *FindAnything* (blue) and baseline (orange) for queries "bed" (top row) and "bathroom" (bottom row).

ground-truth annotations of [43] which do not include labels more closely related to S&R scenarios; and iv) they are not visible from the MAV's starting location. The baseline is the state-of-the-art submapping-based exploration method from [41] which lacks semantic information, demonstrating the benefit of incorporating open-vocabulary information. Both *FindAnything* and the baseline [41] use 5 cm voxels.

The mission is run 10 times for each combination of method and query, with each run lasting for 10 minutes, a typical MAV flight time. The final reconstructed map meshes are used to evaluate the completeness and the Root Mean Squared Error (RMSE) against the ground-truth for each individual item or region of interest. Completeness is computed as the percentage of ground-truth mesh vertices with a reconstructed mesh polygon within 5 cm. The results for the 10th to 90th percentiles are presented in Fig. 5. *FindAnything* achieves an overall higher completeness and mesh accuracy for both queries, while also being more consistent. *FindAnything* also yields a better reconstruction of the items of interest, with a larger margin for the area "bathroom" compared to the item "bed". This is because a large part of the item is observable with even a single view, whereas areas of interest require continued observation which *FindAnything* achieves by sampling candidate next views inside the segment cubes.

E. Real-World Experiments

To demonstrate *FindAnything*'s usefulness in potential S&R scenarios, we simulated a firefighting use case by conducting an autonomous exploration experiment in an office environment. Throughout the mission, we defined target objects or areas relevant to fire response through natural language queries. We explored until a "fire extinguisher" was observed and then explored under the query "kitchen", as it is a house area with a higher risk of fire. Figure 1 shows the 3D reconstruction together with map activations for the queries "fire extinguisher" and "exit". It shows that our system can explore the environment and build a volumetric representation suitable for navigation while aggregating the language embeddings in object-centric submaps.

TABLE IV

MEAN PER-FRAME TIMES FOR THE STAGES OF *FindAnything*.

Stage	Mean time [ms]		
	Replica	Exploration Sim	Exploration MAV
CLIP inference	14	14	340
eSAM inference	105	105	171
Render map segments	12	7	16
Segment tracking	109	6	22
Vision-Language fusion	30	14	37
Depth integration	40	24	64
SLAM frontend	25	41	32
SLAM backend	13	24	16

Horizontal lines separate stages that occur in different threads.

We demonstrate that *FindAnything* enables 3D exploration on resource-constrained devices by deploying it on a real-world MAV relying only on onboard computation. The system used in this experiment is a custom-built quadcopter (see Fig. 2) equipped with a RealSense D455 stereo RGB-D camera and an NVIDIA Jetson Orin NX 16 GB computer.

To achieve faster eSAM inference on the on-board computer, we downsample the image to half its original resolution and query 36 equally distributed points. A timing break-down of the individual system components is presented in Table IV. To avoid sparse reconstructions due to slow network inference on the MAV, we allow the integration of depth measurements also in the absence of CLIP features or eSAM segments at a software-imposed rate of 3 Hz, to maintain the spirit of the original algorithm.

V. CONCLUSION

We have presented *FindAnything*, a real-time open-vocabulary object-centric volumetric mapping framework, designed for scalability and online deployment on resource-constrained devices. Using foundation models enables our system to be deployed in unknown environments without prior knowledge of the scene. We have demonstrated *FindAnything*'s semantic mapping accuracy in both simulation and real-world indoor and outdoor environments, while being faster and significantly more memory efficient compared to baselines. These capabilities make *FindAnything* suitable for common robotics downstream tasks, such as exploration in S&R scenarios and allows natural language-based interaction with the robot. As a first of its kind, *FindAnything* has been successfully deployed online on-board a resource-constrained MAV.

In the future, we plan to push the boundaries for robotic exploration even further by incorporating semantic priors or hierarchical map representations to enable informed exploration even when lacking observations in the current scene. Furthermore, considering dynamic objects and people in the scene remains an open challenge to be addressed.

REFERENCES

[1] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, "Fast frontier-based information-driven autonomous exploration with an MAV," in *IEEE International Conference on Robotics and Automation*, 2020.

[2] H. Oleynikova, C. Lanegger, Z. Taylor, M. Pantic, A. Millane, R. Siegwart, and J. Nieto, "An open-source system for vision-based micro-aerial vehicle mapping, planning, and flight in cluttered environments," *Journal of Field Robotics*, vol. 37, no. 4, 2020.

[3] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *IEEE International Conference on Robotics and Automation*, 2017.

[4] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3D object discovery," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, 2019.

[5] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From SLAM to spatial perception with 3D dynamic scene graphs," *The International Journal of Robotics Research*, vol. 40, no. 12–14, 2021.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021.

[7] O. Alama, A. Bhattacharya, H. He, S. Kim, Y. Qiu, W. Wang, C. Ho, N. Keetha, and S. Scherer, "Rayfronts: Open-set semantic ray frontiers for online scene understanding and exploration," *arXiv preprint arXiv:2504.06994*, 2025.

[8] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3D scene graphs," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, 2024.

[9] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation," in *Robotics: Science and Systems*, 2024.

[10] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, G. Iyer, S. Saryazdi, T. Chen, A. Maalouf, S. Li *et al.*, "ConceptFusion: Open-set multimodal 3D mapping," in *Robotics: Science and Systems*, 2023.

[11] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg *et al.*, "Segment anything," in *IEEE/CVF International Conference on Computer Vision*, 2023.

[12] Y. Xiong, B. Varadarajan, L. Wu, X. Xiang, F. Xiao, C. Zhu, X. Dai, D. Wang, F. Sun, F. Iandola *et al.*, "EfficientSAM: Leveraged masked image pretraining for efficient segment anything," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[13] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems*, 2023.

[14] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," in *International Conference on Learning Representations*, 2022.

[15] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling open-vocabulary image segmentation with image-level labels," in *European Conference on Computer Vision*, 2022.

[16] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li *et al.*, "RegionCLIP: Region-based language-image pretraining," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[17] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LERF: Language embedded radiance fields," in *IEEE/CVF International Conference on Computer Vision*, 2023.

[18] G. Liao, K. Zhou, Z. Bao, K. Liu, and Q. Li, "OV-NeRF: Open-vocabulary neural radiance fields with vision and language foundation models for 3D semantic understanding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 12, 2024.

[19] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "LangSplat: 3D language Gaussian splatting," *IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2024.

[20] K. Yamazaki, T. Hanyu, K. Vo, T. Pham, M. Tran, G. Doretto, A. Nguyen, and N. Le, "Open-Fusion: Real-time open-vocabulary 3D mapping and queryable scene representation," in *IEEE International Conference on Robotics and Automation*, 2024.

[21] S. Lu, H. Chang, E. P. Jing, A. Boularias, and K. Bekris, "OVIR-3D: Open-vocabulary 3D instance retrieval without training on 3D data," in *7th Annual Conference on Robot Learning*, 2023.

[22] M. Wei, T. Wang, Y. Chen, H. Wang, J. Pang, and X. Liu, "OVExp: Open vocabulary exploration for object-oriented navigation," *arXiv preprint arXiv:2407.09016*, 2024.

[23] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *IEEE International Conference on Robotics and Automation*, 2023.

[24] T. B. Martins, M. R. Oswald, and J. Civera, "OVO-SLAM: Open-vocabulary online simultaneous localization and mapping," *arXiv preprint arXiv:2411.15043*, 2024.

[25] L. Schmid, M. Abate, Y. Chang, and L. Carlone, "Khronos: A unified approach for spatio-temporal metric-semantic SLAM in dynamic environments," in *Robotics: Science and Systems*, 2024.

[26] Z. W. Wang, Y. Su, C. Li, D. Wang, Y. Huang, B. Z. Zhao, and X. L. Li, "Open-vocabulary octree-graph for 3D scene understanding," in *IEEE/CVF International Conference on Computer Vision*, 2025.

[27] M. Ranzinger, G. Heinrich, J. Kautz, and P. Molchanov, "AM-RADIO: Agglomerative vision foundation model reduce all domains into one," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[28] S. Boche, J. Jung, S. B. Laina, and S. Leutenegger, "OKVIS2-X: Open keyframe-based visual-inertial SLAM configurable with dense depth or LiDAR, and GNSS," *IEEE Transactions on Robotics*, 2025.

[29] S. Leutenegger, "OKVIS2: Realtime scalable visual-inertial SLAM with loop closure," *arXiv preprint arXiv:2202.09199*, 2022.

[30] N. Funk, J. Tarrio, S. Papatheodorou, M. Popović, P. F. Alcantarilla, and S. Leutenegger, "Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, 2021.

[31] S. Barbas Laina, S. Boche, S. Papatheodorou, D. Tzoumanikas, S. Schaefer, H. Chen, and S. Leutenegger, "Scalable autonomous drone flight in the forest with visual-inertial SLAM and dense submaps built without LiDAR," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025.

[32] C. Zhou, C. C. Loy, and B. Dai, "Extract free dense labels from CLIP," in *European Conference on Computer Vision*, 2022.

[33] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis *et al.*, "ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning," in *IEEE International Conference on Robotics and Automation*, 2024.

[34] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren *et al.*, "The Replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[35] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[36] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu *et al.*, "Habitat: A platform for embodied AI research," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[37] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "FoundationStereo: Zero-shot stereo matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[38] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle *et al.*, "SAM 2: Segment anything in images and videos," in *International Conference on Learning Representations*, 2025.

[39] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[40] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "VLFM: Vision-language frontier maps for zero-shot semantic navigation," in *IEEE International Conference on Robotics and Automation*, 2024.

[41] S. Papatheodorou, S. Boche, S. Barbas Laina, and S. Leutenegger, "Efficient submap-based autonomous MAV exploration using visual-inertial SLAM configurable for LiDARs or depth cameras," in *IEEE International Conference on Robotics and Automation*, 2025.

[42] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner *et al.*, "Habitat-Matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI," in *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[43] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet *et al.*, "Habitat-Matterport 3D semantics dataset," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[44] Open Robotics, "Gazebo Fortress," https://gazebosim.org, 2021.