RayFronts: Open-Set Semantic Ray Frontiers for Online Scene Understanding and Exploration

rayfronts.github.io

Omar Alama, Avigyan Bhattacharya, Haoyang He, Seungchan Kim, Yuheng Qiu, Wenshan Wang, Cherie Ho, Nikhil Keetha, Sebastian Scherer

Carnegie Mellon University

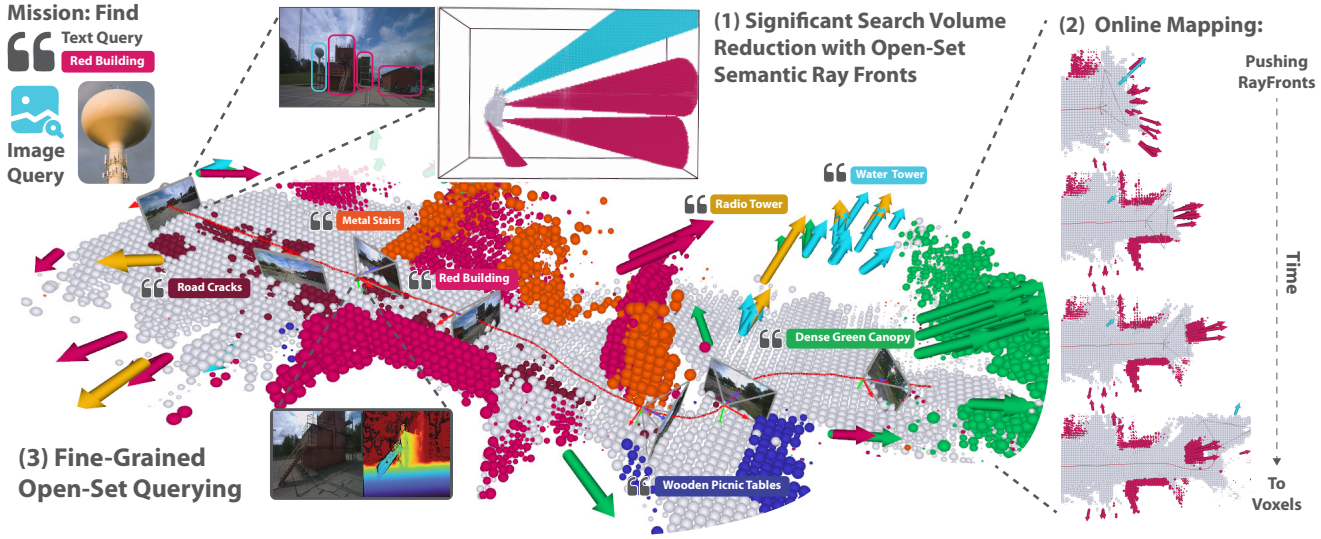

Fig. 1: **`RayFronts` is a real-time semantic mapping system that enables fine-grained scene understanding both within and beyond the depth perception range.** Given an example mission through multi-modal queries to locate red buildings & a water tower, **`RayFronts`** enables: (1) Significant search volume reduction for online exploration (as shown by the red and blue cones at the top) and localization of far-away entities (e.g., the water & radio tower). (2) Online semantic mapping, where prior semantic ray frontiers evolve into semantic voxels as entities enter the depth perception range (e.g., the red buildings query on the right side). (3) Multi-objective fine-grained open-set querying supporting various open-set prompts such as "Road Cracks", "Metal Stairs", and "Green Dense Canopy".

***Abstract*— Open-set semantic mapping is crucial for open-world robots. Current mapping approaches either are limited by the depth range or only map beyond-range entities in constrained settings, where overall they fail to combine within-range and beyond-range observations. Furthermore, these methods make a trade-off between fine-grained semantics and efficiency. We introduce `RayFronts`, a unified representation that enables both dense and beyond-range efficient semantic mapping. `RayFronts` encodes task-agnostic open-set semantics to both in-range voxels and beyond-range rays encoded at map boundaries, empowering the robot to reduce search volumes significantly and make informed decisions both within & beyond sensory range, while running at 8.84 Hz on an Orin AGX. Benchmarking the within-range semantics shows that `RayFronts`'s fine-grained image encoding provides 1.34× zero-shot 3D semantic segmentation performance while improving throughput by 16.5×. Traditionally, online mapping performance is entangled with other system components, complicating evaluation. We propose a planner-agnostic evaluation framework that captures the utility for online beyond-range search and exploration, and show `RayFronts` reduces search volume 2.2× more efficiently than the closest online baselines.**

Authors are with Carnegie Mellon University, Pittsburgh, PA, USA. Emails: {oalama, avigyanb, hhe2, seungch2, yuhengq, wenshanw, cherieh, nkeetha, basti}@andrew.cmu.edu

I. Introduction

Open-set semantic mapping is essential for robotic systems to reason, search, and navigate in open-world environments. The task requires capturing both fine-grained local details and distant beyond-range semantic cues in real-time. For instance, as shown in Fig. 1, an aerial or ground robot may need to localize the water or radio towers over 100 meters beyond its depth perception capability, as well as locate any hazards (road cracks) or interesting structures along the way (red building). This work explores what the most effective semantic mapping system would be to capture this information to imbue the robot with the ability to reason within and beyond depth sensing limitations.

Although there is a growing body of literature on open-set metric semantic mapping [1]–[5], these methods focus primarily on offline mapping for downstream usage in limited environments ignoring efficiency and depth-sensing limitations. Such representations cannot guide the robot in search and exploration tasks as they provide no information about the unmapped region. Other works change the way semantics are typically encoded in a map (point clouds, voxels, and

bounding boxes) to representations that can guide exploration (i.e semantic frontiers [6], [7], and semantic poses [8]). However, existing semantic frontier maps are limited to 2D indoor environments and have limited semantics due to whole-image encoding [6] or using closed-set models [7], whereas semantic poses [8] lack fine-grained reconstructions and can only recognize prominent objects in an image.

In this context, we explore the question, **"How to design an efficient online mapping representation that facilitates fine-grained scene understanding, and be aware of beyond-range semantic entities?"** We introduce **`RayFronts`**, a semantic map representation which seamlessly integrates traditional within-depth mapping with ray-based representations, facilitating both dense mapping within observed depth ranges and perception beyond them. Unlike conventional representations truncated at the depth range, multi-directional semantic ray frontiers retain coarse-grained far-range information, enabling downstream planners (e.g., object-search) to reduce their search volume significantly. Additionally, to assess the utility of the proposed representation, we construct a planner-agnostic benchmark and propose a new metric to measure how effectively an online mapping strategy reduces the search space for fast object localization and exploration. Finally, to avoid image level encoding and expensive pipelines, we introduce a novel image encoder that achieves state-of-the-art performance on zero-shot 3D semantic segmentation enabling a computationally efficient, open-world, and deployable 3D online mapping system.

Our key contributions are as follows:

C1: Unified 3D Map Representation for Within-Depth and Beyond-Depth Perception: We develop the first-of-its-kind open-set semantic ray frontier 3D map, which enables robots to reason in open environments achieving up to 1.85x mIoU in offline zero-shot performance, and are 2.2x more efficient in reducing search volume in online mapping than the closest offline & online baselines respectively.

C2: Planner-Agnostic Online Semantic Mapping Evaluation Framework: We showcase that online semantic mapping systems can be evaluated on their fundamental utility for exploration, without being tightly coupled with a planner, by developing a metric that assesses "correctly reduced search volume".

C3: Efficient real-time open-set online mapping system: can run end to end at 8.84 Hz on an ORIN AGX and our efficient dense vision-language encoder is 16.5x faster than the closest baseline and achieves state-of-the-art on open-vocab zero-shot 3D semantic segmentation mIoU.

II. Related Work

A. Dense 2D Open-Set Semantics

The rapid rise of foundation models [9] has spearheaded progress in tasks requiring fine-grained open-set concepts which are hard to capture with a fixed taxonomy of semantic classes [10], [11]. CLIP [12] and its subsequent variants such as SIGLIP [13] have shown impressive alignment between abstract textual concepts and images. These Visual Language Models (VLMs) initially aligned textual descriptions and images as a whole and not to particular regions or pixels. Subsequently, follow-up work based on supervised and unsupervised regimes has attempted to address this issue [14]. A recurring theme in these methods is the trade-off between efficiency and accuracy, with the most performant approaches often using multiple foundation models like DINOv2 [10], Grounding DINO [15], and SAM [11]. This is not optimal for online real-world deployment and hence we explore the applicability of RADIO [16], a foundation model aligned with various dense visual foundation models. While RADIO's language alignment is to the image as a whole, we find that employing a simple attention trick [17] with its SIGLIP adapter enables us to achieve state-of-the-art pixel-level language alignment and real-time performance on embedded hardware.

B. Offline & Bounded Open-Set Semantic Mapping

Traditional semantic mapping systems have relied on learning-based methods to detect and segment a fixed set of concepts, with performance limited by vocabulary size and training distribution [18]–[24]. With the rise of dense 2D open-set semantics, interest has grown in open-vocab semantic mapping systems using representations like point clouds, voxels, and scene graphs [25]. These systems have shown strong open-world capabilities for navigation, manipulation, and scene understanding [1]–[5], [26]–[28]. However, most focus on offline database maps and lack online utility for robotics, with many design choices making real-time deployment infeasible. To address this, we introduce a computationally efficient, fast, and deployable 3D mapping system for online scene understanding.

C. Online & Unbounded Open-Set Semantic Mapping

While offline and bounded semantic mapping has excelled in indoor scenes, it struggles with outdoor, unbounded, and unstructured environments, where limited depth perception becomes a challenge. An effective online semantic mapping system must support both efficient exploration and fine-grained scene understanding. VLFM [6] addresses this by encoding semantics on 2D frontiers for object goal navigation, but it is limited to a single object at a time and only works in indoor settings. Similarly, Embedding Pose Graph (EPG) [8] encodes semantics into rays from pose nodes, but lacks fine-grained mapping and condenses the entire image into one feature vector, risking the loss of subtle details.

In contrast, we propose a novel representation combining semantic voxels with ray-based frontiers, capturing multiple viewing directions and open-set features. This approach enables efficient online search and rough triangulation of distant objects, allowing us to capture both in-range and beyond-range semantic entities. Our synergy of metric-map-based semantic voxels and direction-based ray frontiers supports fine-grained scene understanding and efficient exploration.

III. Method

We present **`RayFronts`**, a unified 3D semantic mapping system for multi-modal open-set semantic querying of both

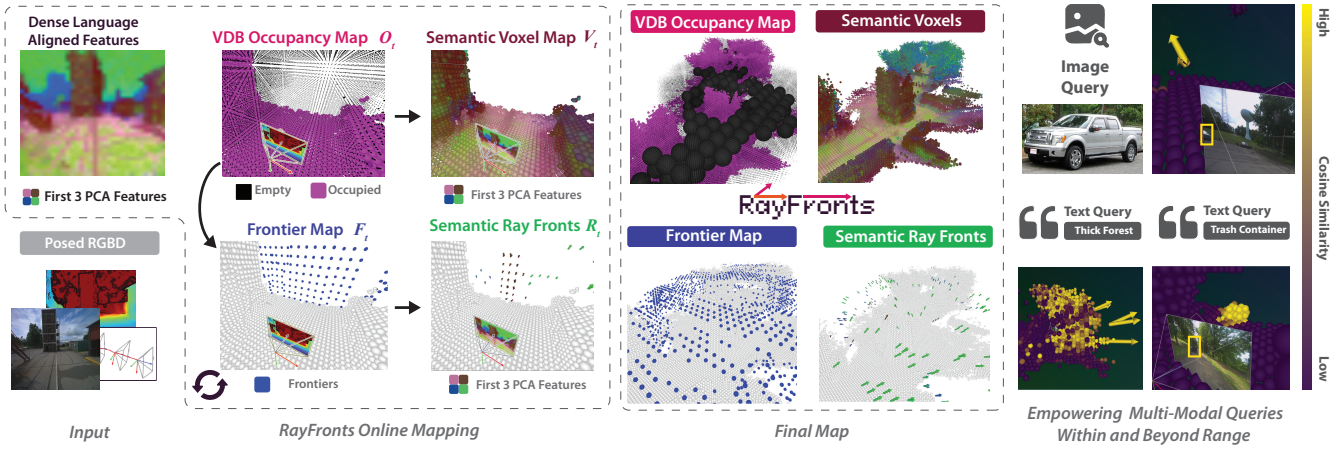

Fig. 2: **Overview of our online mapping system, `RayFronts` is designed for multi-objective & multi-modal open-set querying of both in-range and beyond-range semantic entities.** Given posed RGB-D images, we first extract dense features with our fast language-aligned image encoder. Then, posed depth information and features are used to construct a semantic voxel map for in-range queries. In parallel, **`RayFronts`** also maintains a VDB-based occupancy map to generate frontiers, which are further associated with multi-directional semantic rays. These semantic ray fronts enable us to perform beyond-range querying of open-set concepts in the unobserved region.

in-range and beyond-range semantic entities. **`RayFronts`** maintains a semantic voxel map $\mathscr{V}_t$ containing voxel coordinates and semantic features for within-range entities, an occupancy VDB map $\mathscr{O}_t$, a set of frontiers $\mathscr{F}_t$ denoting subsampled boundary voxels between observed and unobserved spaces, and semantic ray fronts $\mathscr{R}_t$, a ray-based representation on the frontiers, which contains features for beyond-range semantic reasoning.

`RayFronts` operates in four steps: (1) extracting dense, language-aligned features from RGB input through our efficient encoding pipeline, (2) fusing within-range featurized points into a sparse semantic voxel map, (3) maintaining an occupancy map for frontier computation and semantic voxel pruning, and (4) ray casting beyond-range semantics onto frontiers to semantically reason beyond the observed map. Our system is optimized for parallel computing and online mapping, leveraging PyTorch tensors for $\mathscr{V}_t, \mathscr{F}_t, \mathscr{R}_t$ on the GPU and OpenVDB [29] for $\mathscr{O}_t$ on the CPU. This design ensures efficient querying, seamless feature integration, and adaptability to evolving environments. The pipeline and outputs of **`RayFronts`** are illustrated in Fig. 2.

A. Extracting Dense Language-Aligned Features

There has been a rapid growth of methods that extract dense language aligned features from RGB images. However, existing methods fall short by (1) lacking generalization due to limited supervision, (2) sacrificing efficiency with multi-model multi-stage pipelines, or (3) prioritizing efficiency and generalization at the cost of segmentation quality. In this work, we adopt RADIO [16], a VFM that distills key features from CLIP [12], DINOv2 [10], and SAM [11]. This integration enhances feature representation and segmentation performance. However, since RADIO leverages vanilla ViT [30], it struggles with fine-grained localization of visual features, a critical challenge in semantic scene understanding. To address this, we integrate the explicit spatial attention mechanism proposed by NACLIP [17] and modify the RADIO encoder accordingly. Specifically, we augment the attention layer of the final ViT block by introducing a locality constraint via an unnormalized multivariate Gaussian kernel centered around each patch essentially pushing the model to attend to its neighboring patches and improving locality.

To densely align the RADIO feature space with language, we explore the available pre-trained MLP-based adaptor heads provided by RADIO. Simply following RADIO's original distillation approach– projecting spatial features onto CLIP or SIGLIP space using their respective adapters–yields subpar performance. Instead, we use the SIGLIP summary feature adapter to project spatial features to the SIGLIP `CLS` token space, thus resulting in a spatially consistent and language-aligned feature map and observe significant performance improvements over existing methods.

B. Semantic Voxels for Dense Within-Depth Queries

Given a pose $P_t \in \mathbb{SE}3$ and depth map $D_t \in \mathbb{R}^{H \times W}$, we initialize a voxel grid retaining only points within the frustrum, yielding Q_t. We transform the points Q_t into the camera frame and classify its occupancy based on depth D_t. For each occupied point, we find the associated feature via nearest-neighbor interpolation yielding $\mathscr{P}_t^{local} = \{(\mathbf{p}_i, \mathbf{f}_i)\}_{i=1}^{M}$ where $\mathbf{p}_i \in \mathbb{R}^3$ are point coordinates and $\mathbf{f}_i \in \mathbb{R}^{3+D+1}$ (3 for RGB, D is feature dimension, and 1 for the hit count). Local updates are accumulated into a buffer of m frames before being voxelized at resolution α and integrated into the global map $\mathscr{V}$.

Feature Fusion and Aggregation: Rather than complex fusion methods used in [2], [3], we employ a simple weighted average, where each voxel's hit count serves as the weight when fusing features within the same voxel. To achieve this, we concatenate coordinate and feature tensors of accumulated local updates $\mathscr{P}_t^{local}$ with those of global voxel map $\mathscr{V}_t$. A parallel scatter-reduce operation fuses features at the same discretized coordinates into a single voxel.

C. Occupancy Mapping for Frontiers and Pruning

To represent occupancy map $\mathscr{O}_t$ efficiently, we employ OpenVDB [29], recently used in modern 3D frontier-based exploration works [31]–[33] for its sparse tree representation and multi-resolution capability. Following standard practice, we store log-odds occupancy o_j in a signed byte. To better tolerate dynamic environments and to avoid overflow, we limit $prob_{occ}(o_j)$ to lower and upper limits. Fig. 2 shows the OpenVDB map with large free voxels showing the multi-resolution aspect of the occupancy representation.

Pruning Semantic Voxels: When accumulating voxels over long distances and time periods, odometry drifts and dynamic objects can introduce inconsistencies, not to mention the growing memory consumption. To mitigate this, we prune invalid semantic voxels by querying the occupancy map $\mathscr{O}_t$ and removing those with occupancy below 0.5.

D. Finding the "Fronts": Computing 3D Frontiers

We identify frontiers by iterating over all free observed voxels using efficient OpenVDB iterators and examining their neighbors. A voxel is considered a frontier if its neighbors meet the minimum thresholds for unobserved ($min_{unobsrv}$), occupied (min_{occ}), and free (min_{free}) counts, allowing us to emphasize frontiers near surfaces or open space as needed. To reduce density, we subsample the frontier map using a coarser voxel grid of size β. Fine-grid frontiers are accumulated into a coarser grid, and cells with enough frontiers remain as frontiers.

E. Semantic Ray Frontiers for Beyond-Depth Mapping

Need for richer frontiers: Existing semantic frontier methods have fundamentally constrained beyond-range semantic encoding, where only a single object can be pursued at a time due to feature collisions from distinct objects observed through the same frontier. To enable multi-object semantic guidance for search and exploration, we transition from conventional semantic frontiers $\mathscr{F}_{sem} = \{(\mathbf{p}_k, \mathbf{f}_k)\}_{k=1}^{F}$ to semantic ray frontiers $\mathscr{R}_{sem} = \{(\mathbf{o}_r, \theta_r, \phi_r, \mathbf{f}_r)\}_{r=1}^{R}$, where $\mathbf{o}_r$ is ray origin, $\theta_r \in [-\pi, \pi)$ and $\phi_r \in [0, \pi)$ are azimuthal and zenith angles, and $\mathbf{f}_r$ are semantic features. This shift drastically enhances the mapping system by allowing efficient storage of rich multi-object semantics with minimal feature collisions, enabling rough triangulation of object locations, and reducing the search space volume needed for exploration. We discuss the ray mapping process (observe, associate, discretize & accumulate) and how rays are pruned and propagated below.

Observe: To identify out-of-range regions in the feature map F_t we compute a boolean mask $M_t \in \mathbb{R}^{H \times W}$ from the depth D_t (obtained via stereo, LiDAR, or monocular depth estimation). The mask encompasses either $+\infty$ values from depth sensors or far low-certainty values. M_t is eroded to prevent semantic leakage at object boundaries, and used to select the semantic pixels to propagate as rays $\mathscr{R}_t^{local} = \{(\mathbf{o}_r, \mathbf{d}_r, \mathbf{f}_r)\}_{r=0}^{H_t}$ where $\mathbf{o}_r \in \mathbb{R}^3$ is the ray and camera origin, $\mathbf{d}_r \in \mathbb{R}^3$ is the normalized direction vector, and $\mathbf{f}_r$ represents semantic features.

Associate (Matching Rays to Frontiers): In the presence of depth information, rather than keeping rays at the robot's origin as in [8], we leverage the mapped area to push rays closer to their observed entities, improving localization. For each semantic ray $(\mathbf{o}_r, \mathbf{d}_r, \mathbf{f}_r)$, we select a frontier from the candidate set $\mathscr{F}_{t+1}$ through a two-step filtering process. First, we prune frontiers by (1) removing those not in front of the ray (2) computing the shortest orthogonal distance d_{ortho} between the ray and frontiers, discarding those where $d_{ortho} > \beta$ (exceeding the frontier grid cell size), and (3) calculating the distance from ray origin $\mathbf{o}_r$ to frontier origin $\mathbf{p}$, obtaining d_{orig} and removing frontiers where $d_{orig} > 4 \times depth\_range$.

Next, for the remaining k candidate frontiers, we compute a cost function

$$d_{cost} = \left(\frac{d_{ortho}}{max(\{d^r_{ortho}\}_{r=0}^{k})} + \frac{d_{orig}}{max(\{d^r_{orig}\}_{r=0}^{k})}\right)/2, d_{cost} \in [0,1] \tag{1}$$

We select the frontier with the minimum d_{cost} as the best match. We qualitatively find that utilizing both d_{ortho} and d_{orig} improves results and prevents distant frontiers from receiving noisy semantics.

For further refinement, we optionally apply ray tracing, marching each ray through the occupancy map $\mathscr{O}_{t+1}$ until it reaches its assigned frontier or encounters occupied or unobserved (possibly occupied) cells. At this stage, each semantic ray is associated with a frontier, updating its origin $(\mathbf{o}_r)$ to the corresponding frontier origin $\mathbf{p}$. Since we lack depth information about the underlying semantic entity, we maintain the ray's direction $\mathbf{d}_r$ when shifting its origin.

Discretize and Accumulate ("Ray Binning"): Similar to voxelization techniques, we organize semantic rays into angle bins with a resolution of ψ degrees. The normalized ray directions $\mathbf{d}_r$ are converted to spherical angles using: $\theta_\mathbf{r} = atan2(d_r^1, d_r^0), \quad \phi_\mathbf{r} = acos(d_r^2)$ where $atan2$ is the four-quadrant inverse tangent. We then discretize these angles and merge rays that correspond to the same frontier and the same angle bin from both the local update $\mathscr{R}_t^{local}$ and the global set $\mathscr{R}_t$. We use $1 - d_{cost}$ for weighing the features while merging, assigning lower trust to high-cost associations. This yields the updated ray-frontier $\mathscr{R}_{t+1}$.

Pushing the ray fronts onward: Semantic ray frontiers must be updated as new areas are mapped. After computing the frontier update $\mathscr{F}_{t+1}$, we use a voxel grid to perform a set intersection between all ray origins and frontier origins, similar to pruning voxels, and remove rays no longer associated with active frontiers. If ray tracing is enabled, removed rays are added back to the ray accumulation buffer to be re-cast in the next iteration. This preserves previously observed semantics that may no longer be directly visible as the frontier shifts (e.g., from a side view). However, without ray tracing, removed rays would continue to propagate indefinitely, so we disable the behavior under that setting. In our experiments, we use ray tracing unless otherwise stated.

IV. Experimental Setup

A good online mapping system should (1) intelligently guide the robot toward regions of interest in any environment,

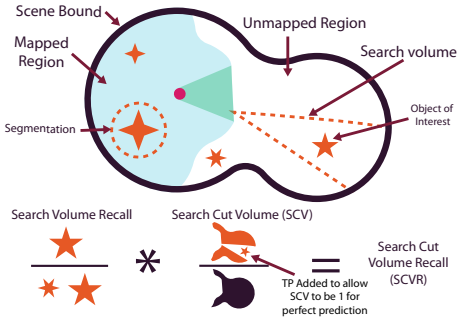

Fig. 3: **An illustration of our proposed planner-agnostic metric (Search Cut Volume Recall) for open-world online search benchmarking.** Intuitively, the metric captures "How much of the search volume is eliminated correctly?" An optimal mapper should promptly and accurately reduce the search space, enabling fast multi-object localization and exploration.

eliminating irrelevant volumes early, (2) accurately capture fine-grained open-set semantics within a metric map, and (3) do so efficiently. In this section, we first introduce our proposed online mapping evaluation framework, which assesses the utility of a mapping system in guiding exploration without a planner in-the-loop, and introduce competitive baseline representations. We then present our extensive offline map evaluation following established protocols. Finally, we conduct a deployability and throughput analysis.

A. Planner-Agnostic Online Semantic Mapping Evaluation

Dataset: Originally designed to challenge visual SLAM with large, cluttered, long-tail objects in indoor and outdoor environments, TartanAirV2 [34] serves as a stress test for our representation. To simulate scenarios with severely limited depth, we choose four large outdoor scenes `AbandonedCableday`, `Factory`, `Downtown` and `ConstructionSiteOvercast` where bounding boxes span approximately 8 million m^3 with a 50m range cutoff. We generate ground truth occupancy (defining the scene volume) and a semantic label map at 1-meter voxel resolution from the provided posed RGBD input.

Baselines: There are no established online mapping baselines for 3D open-world environments. Therefore, we take inspiration from existing works and design the following baselines, keeping the encoder fixed to isolate the impact of our mapping approach:

- **Semantic Poses** (Sem Pose): Emulates EPG [8] by using global encoding for the image, resulting in a single ray per frame located at the robot origin.
- **Semantic Voxels** (Sem Voxels): Subsumes representations that encode only within-range semantics [2]–[4].
- **Semantic Frontiers**: Emulates in 3D the 2D approaches that paint frontiers with semantics [6], [7]. We recognize that there are two ways semantic frontiers can be interpreted; (1) As a spherical region encompassing the semantic entity (i.e **Spherical Sem Fronts**), or (2) as a single ray pointing away from the observed region (i.e **Unidirectional Sem Fronts**). We evaluate both.

We define search volume as the unmapped region unless further evidence is provided. Ray-based approaches cast search cones while spherical sem fronts define a sphere volume extending to the nearest frontier. Multiple search volumes are summed in a voxel grid, counts are normalized, and thresholded at 0.05 to get the final search volume for a class. For **Unidirectional Sem Fronts**, frontier directions are inferred using the occupancy map $\mathscr{O}_t$ by computing a weighted combination of all directions around a frontier in a 3x3x3 window where mapped voxels have a weight of -1 (Pushing away) and unmapped voxels have a weight of +1 (Pulling toward).

Evaluation Protocol: We ask the question "Can an online semantic mapping system's utility for search and exploration be assessed independently of specific planners ?" Yes, the key is examining how accurately and efficiently the map constrains the search space. Traditional mapping metrics such as mIoU, mAcc, F1 measure fine-grained semantic localization but overlook search volume efficiency, as they ignore **true negatives**. In search and exploration, a high true negative rate in the unobserved region reduces wasted search time. Therefore, for beyond-range search volume estimation, we introduce a novel metric below, and to evaluate within-range fine-grained online performance, we use the area under the mIoU-time curve.

Search Cut Volume Recall Metric: Our proposed metric, shown in Figure 3, measures how accurately and efficiently a mapping system cuts search volume. The intuitive definition is to compute total unmapped volume $vol_{unmapped}$ and subtract the search volume from it, however to avoid punishing true positives, we define search cut volume (SCV) as:

$$SCV = 1 - \frac{FP_{unmapped}}{vol_{unmapped}}, \quad SCV \in [0,1] \tag{2}$$

To temper the metric against incorrectly cutting down volume, we compute Recall in the unmapped region and multiply it with SCV yielding the **Search Cut Volume Recall (SCVR)** metric:

$$SCVR = SCV * \frac{TP_{unmapped}}{FN_{unmapped} + TP_{unmapped}}, \quad SCVR \in [0,1] \tag{3}$$

The **SCVR** metric is robust to both naive cases (1) not constraining the search volume, or (2) constraining it to 0 volume, yielding 0 for both. For an aggregate, we compute the area under the SCVR-time curve, stopping time for each class when 50% of it has entered the mapped region. To further assess the robustness of **`RayFronts`**, we vary depth sensing range at 0m, 10m, and 20m.

B. Offline 3D Open-Vocabulary Semantic Segmentation

Datasets: We follow prior work [2]–[4] and evaluate on Replica (`office[0-4]`, `room[0-2]`) and ScanNet (`scene[0011,0050,0231,0378,0518]`). In line with previous protocols, we report results while ignoring background classes ("*floor*", "*wall*", "*ceiling*", "*door*", "*window*"). However, we additionally evaluate across all classes to demonstrate our ability to handle background seamlessly.

TABLE I: Online & Unbounded Semantic Mapping Benchmarking on TartanAirV2 [34]. Ranking shown as **first**, second, and third.

Methods	0m Depth (AUC)				10m Depth (AUC)				20m Depth (AUC)			
	mIoU(%)	SCV(%)	Recall(%)	SCVR(%)	mIoU(%)	SCV(%)	Recall(%)	SCVR(%)	mIoU(%)	SCV(%)	Recall(%)	SCVR(%)
Sem Poses	0.00	11.37	91.91	4.02	–	–	–	–	–	–	–	–
Sem Voxels	–	–	–	–	20.49	0.00	100.00	0.00	13.03	0.00	100.00	0.00
Spherical Sem Fronts	–	–	–	–	20.49	18.00	82.35	0.40	13.03	13.33	87.02	0.41
Unidirectional Sem Fronts	–	–	–	–	20.49	16.12	85.93	3.15	13.03	11.58	89.54	2.07
RayFronts (Ours)	0.00	36.59	75.37	**16.27**	20.49	22.94	81.15	**7.08**	13.03	14.32	88.69	**4.56**

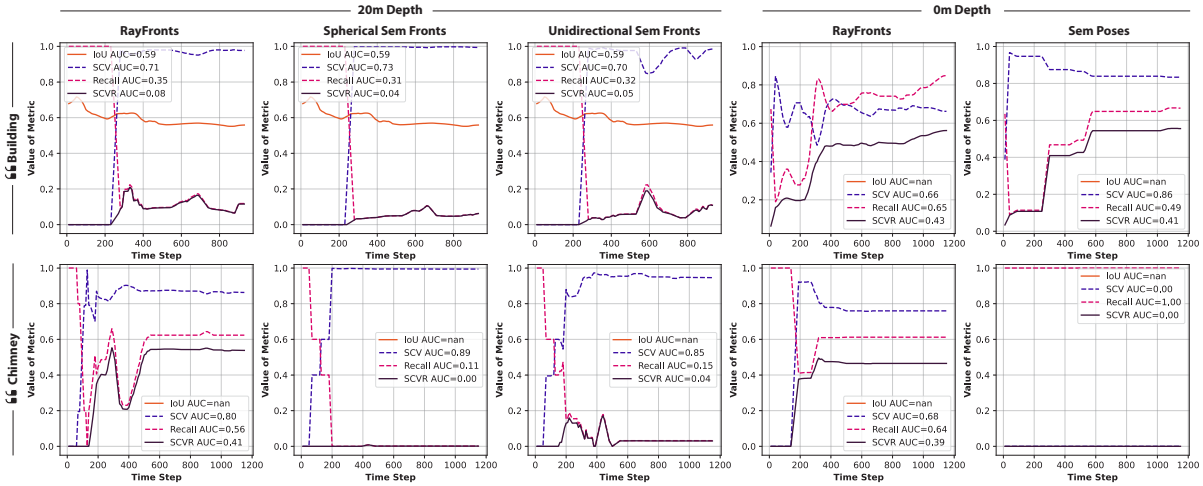

Fig. 4: **RayFronts consistently surpasses baselines for online semantic mapping**. Two query scenarios are shown: (1) querying for a prominent object (i.e Building) that enters depth range, and (2) a distant object (i.e Chimney) that remains beyond range. Through unified dense voxel mapping, and beyond-range semantic ray frontiers, **RayFronts** sets the upper-bound in both scenarios.

Moreover, to showcase **RayFronts**'s effectiveness in outdoor, unstructured, "in-the-wild" environments, we further evaluate on the TartanAirV2 [34] scenes referenced in IV-A excluding methods that cannot function outdoors.

Baselines: We compare our method with two categories of approaches: (1) vision-language representations that create 3D semantic maps, namely, ConceptFusion [2], ConceptGraphs [4], and HOV-SG [3]; and (2) zero-shot semantic segmentation encoders, namely, NACLIP [17] and Trident [35]. We extend the latter encoder-based methods to 3D using the same projection and fusion method as our system.

Evaluation Protocol: We follow standard open-vocabulary semantic segmentation evaluation protocols. We generate 3D segmentations by running HOV-SG and ConceptGraph code ensuring an accurate representation of their scene graph method. For all others, we generate segmentations by computing the cosine similarity between the embedded feature and the class-name text embedding, making a voxel prediction if its softmax probability exceeds 0.1. We encode class names using each method's specified templates. For our approach, we follow NACLIP and use 80 templates [17], with a prompt denoising [36] threhsold of 0.5 to suppress irrelevant classes. We also apply k-NN matching (k=5) following HOV-SG [3] protocol, assigning each GT voxel the majority label. All baselines use the ViT-L model architecture for consistency. We resize images to 480x640, apply a frame skip of 10, 5cm voxels for Replica and ScanNet and 1m voxels for TartanAir.

V. Results & Discussion

A. Online Semantic Mapping

Table I summarizes online performance of the five methods in their respective operating ranges. We observe that **RayFronts** excels and is the upper bound across depth ranges. **Sem Poses** fails to capture any fine-grained reconstructions scoring 0 mIoU-AUC, while **Sem Voxels** fails to provide any information about the unmapped region scoring 0 SCVR-AUC. At 0 depth range, **RayFronts** attaches dense semantic rays at each pose as opposed to the global encoding scheme employed by **Sem Poses**. This allows us to encode non-prominent objects seamlessly and results in a $\sim 4\times$ SCVR-AUC than **Sem Poses**. This observation is illustrated in Fig. 4 where for a simple prominent object such as "building", both **Sem Poses** and **RayFronts** perform similarly. However, for a more distant object like "chimney", **Sem Poses** fails to capture its semantics entirely. At higher depth ranges, we observe that **RayFronts** consistently outperforms semantic frontier baselines at $\sim 2.2\times$ the SCVR-AUC. We attribute this to (1) less semantic collisions as distinct objects are unlikely to be fused in the same ray unlike semantic frontiers which can have many collisions, (2) better preservation of the angle at which the semantic entity was observed from, and (3) allowing each frontier to have multiple rays attached, increasing the density of beyond-range semantics. **RayFronts** is superior to all baselines across depth ranges **empowering both fine-grained localization and beyond-range guidance**.

TABLE II: Offline 3D Semantic Segmentation Benchmarking on Indoor Datasets.

Methods	Replica [37]						ScanNet [38]					
	Without Background			With Background			Without Background			With Background		
	mIoU(%)	f-mIoU(%)	Acc(%)	mIoU(%)	f-mIoU(%)	Acc(%)	mIoU(%)	f-mIoU(%)	Acc(%)	mIoU(%)	f-mIoU(%)	Acc(%)
ConceptFusion [2]	21.07	31.51	35.65	20.38	35.75	41.58	21.76	26.71	34.13	18.57	23.06	28.77
ConceptGraphs [4]	11.63	16.61	19.80	11.72	21.35	28.28	21.62	24.32	31.04	20.83	23.61	35.80
HOV-SG [3]	16.93	31.45	34.74	19.29	30.64	35.17	26.79	36.05	44.17	23.48	28.92	38.52
NACLIP-3D [17]	20.37	35.08	47.47	15.30	16.98	26.23	31.66	39.03	51.65	22.32	24.32	33.46
Trident-3D [35]	21.30	43.34	54.79	20.63	38.53	50.31	29.97	37.62	51.06	24.80	27.77	38.43
RayFronts (Ours)	**39.37**	**62.03**	**68.80**	**27.73**	**43.37**	**54.45**	**41.29**	**46.42**	**56.76**	**32.29**	**39.04**	**49.15**

TABLE III: Offline 3D Semantic Segmentation Benchmarking on an Outdoor Dataset (TartanAirV2 [34]).

Methods	mIoU (%)	f-mIoU (%)	Acc (%)
ConceptFusion [2]	5.84	32.78	39.76
NACLIP-3D [17]	9.66	40.82	54.10
Trident-3D [35]	9.86	**43.56**	55.34
RayFronts (Ours)	**13.22**	43.43	**57.26**

B. Offline 3D Semantic Segmentation

Table II provides a detailed comparison of the performance between our framework and other zero-shot approaches, outlined in Section IV-B. **RayFronts** consistently outperforms the baselines in mIoU, **and achieves SOTA performance** beating the next best baselines by +18.07% and +9.63% mIoU on Replica and Scannet, respectively, excluding background. **RayFronts** is also able to handle background seamlessly with its single-forward pass approach while segment-and-encode approaches fall short.

For outdoor in-the-wild performance on TartanAirV2, Table III shows that **RayFronts** exceeds the performance of the baselines by 3.36% mIoU. While Trident-3D serves as a close second to our approach and achieves a slightly higher f-mIoU on TartanAirV2 by a marginal 0.13%, it does so at the cost of integrating multiple foundational models into their pipeline, which significantly reduces efficiency—an essential factor for online semantic mapping.

C. Encoder & Mapping Throughput Analysis

To assess deployability, we run **RayFronts** on an NVIDIA Jetson AGX Orin and perform a quantitative comparison of image encoder throughput shown in Fig. 5. Our mapping system achieves SOTA performance in 3D open-set semantic segmentation with 1.34x the mIoU of Trident while being 16.5x faster, running at 17.5 Hz, and with only 46% of the parameters. While NACLIP has similar throughput, we surpass it by a significant 1.81x in mIoU. ConceptFusion's 0.03 Hz throughput makes it impractical for real-time use.

Furthermore, we test the end-to-end throughput of **RayFronts** on a real-world outdoor scene using pre-recorded data from a mobile ground robot. We use a resolution of 224x224, 30cm voxel size, and the base encoder model, while compressing features to top 100 PCA components (retaining $\sim 80\%$ variance), and disabling ray-tracing. **RayFronts runs real-time at 8.84 Hz on Orin AGX.**

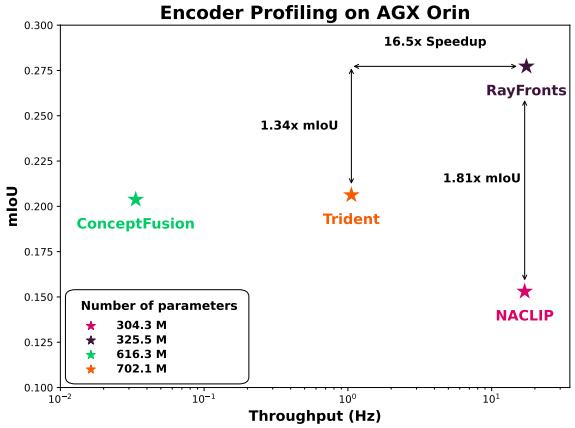

Fig. 5: **RayFronts provides state-of-the-art mIoU & 17.5 Hz throughput on an AGX Orin.** It surpasses Trident with 1.34x higher mIoU and a 16.5x speedup, while achieving 1.81x higher mIoU than NACLIP, which operates at a similar throughput.

D. Qualitative Real-World Study

To evaluate **RayFronts** in unbounded, open-world settings, we record a run through an unstructured fire training facility with a Zed-X camera. As shown in Fig. 1, **RayFronts** accurately reconstructs fine-grained details (e.g., "road cracks") while detecting far-range objects (e.g., "water tower"), demonstrating that **RayFronts empowers robots within and beyond depth-sensing limitations in open-world environments**.

VI. Conclusion

We present **RayFronts**, a real-time semantic mapping system for multi-modal open-set scene understanding for both within- and beyond-range mapping. Our key insight, *semantic ray frontiers*, enables open-set queries about observations beyond depth mapping by associating beyond-depth ray features with the map's frontiers. This allows **RayFronts** to significantly reduce search volumes while retaining fine-grained within-range scene understanding. **RayFronts** improves open-set image encoding with an efficient language-aligned encoder, and introduces a new planner-agnostic metric for open-world search. We achieve state-of-the-art results in 3D open-set semantic segmentation, strong performance in online mapping, and efficient encoder throughput. Our future work aims to include instance differentiation in **RayFronts** and planning integration to facilitate online exploration.

LIMITATIONS

While **RayFronts** is the upper-bound of the online mapping baselines in correct search volume reduction, it also has the highest memory consumption. However, **RayFronts** can be tuned down by reducing ray angle bins down until a single bin (becoming "semantic frontiers"), or reducing depth range down until 0, giving the flexibility to achieve the best trade-off for an application.

ACKNOWLEDGMENTS

This work was supported by Defense Science and Technology Agency (DSTA) Contract #DST000EC124000205, King Abdulaziz University, and National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR) Grant #90IFDV0042. We thank Krishna Murthy Jatavallabhula, Jacob Yeung, Sam Triest and Ayush Jain for insightful discussions and feedback, and Katerina Nikiforova for early encoder exploration.

APPENDIX

A1. CONTRIBUTION STATEMENT

Omar Alama led the research and conceptual design, developed the mapping codebase and online/offline evaluation scripts, wrote major sections of the paper, and created figures, online mapping tables, and videos.

Avigyan Bhattacharya developed the RayFronts encoder code, implemented the ScanNet data loader, and ported Trident for evaluation. Conducted extensive evaluations of multiple offline mapping baselines and wrote the corresponding sections and appendix figures.

Haoyang He ported ConceptFusion and NACLIP for evaluation, performed extensive evaluations of multiple offline mapping baselines, and conducted throughput analyses on the ORIN AGX. Also contributed to writing the corresponding sections.

Seungchan Kim engaged in early discussions, developed the TartanAirV2 data loader, and played a key role in refining the paper writing.

Yuheng Qiu engaged in early discussions, explored datasets for evaluation, and provided writing feedback.

Wenshan Wang participated in early discussions and provided feedback on research direction and paper writing.

Cherie Ho deployed RayFronts on the Orin AGX, assisted in collecting test data across various robot platforms, and contributed extensively to paper writing, coordination, and the design of figures and video.

Nikhil Keetha was heavily involved in brainstorming, discussions, and the conceptual design of the mapping, encoding, and evaluation frameworks. Contributed significantly to paper writing, as well as figures and video design.

Sebastian Scherer shaped the research area, engaged in discussions and brainstorming, and provided valuable feedback on writing, figures, and video.

A2. ONLINE SEMANTIC MAPPING VISUALIZATIONS

Fig. A.1 shows illustrations of the different baselines mentioned in Section IV-A. The top left part of the figure highlights how **RayFronts** can avoid semantic feature collisions (the case where different semantic features are forced to be fused together) by utilizing multiple rays to describe the different semantic entities. Whereas semantic frontier approaches (irrespective of the unidirectional/spherical search volume method) can fail when semantically different entities are observed through the same frontier. The top right part of the figure emphasizes where global encoding approaches like Sem Poses can fail to capture non-prominent objects in the presence of a large centered entity. The illustration provides further explanation for Sem Poses's inability to capture chimneys in the AbandonedCableDay scene as shown in Figs. A.2 and 4. Finally, the bottom row illustrates how each baseline computes its search volume. Spherical Sem Fronts can fail to capture a distant object, with increasing radius cubically increasing search volume. Unidirectional Sem Fronts is highly sensitive to the mapped region topology since it uses it to infer the semantic ray direction, and in the illustrated case, it fails. Sem Poses fails to utilize depth information to push the ray further onto the mapped region boundary for better localization. In contrast with all baselines, **RayFronts** is able to accurately determine the direction of the ray and limit the search volume efficiently, utilizing depth information if available.

Fig. A.2 visualizes the two query scenarios shown in Fig. 4 with ground truth generated at an 80m cutoff (Highest value that fits in our memory) for further clarity. The top block shows search volumes for building at a particular time step. At 20m depth sensing range, it is clear that **RayFronts** achieves the best search volume, having 1.35 $\times$ higher SCVR than Unidirectional Sem Fronts. Spherical Sem Fronts struggles to cast a search volume that encompasses big objects, whereas Unidirectional Sem Fronts has some erroneously inferred ray directions. At 0m range, both Sem Poses and **RayFronts** perform similarly. Furthermore, the bottom block shows the search volume of distant non-prominent objects that never come into the depth sensing range. At 0m range, Sem Poses fails to capture the semantics and fails to reduce search volume, yielding an SCVR of 0, whereas **RayFronts** provides meaningful areas to explore.

A3. OFFLINE SEMANTIC MAPPING VISUALIZATIONS

Fig. A.3 showcases open-vocabulary semantic segmentation samples across different datasets, while Fig. A.4 highlights the open-vocabulary capabilities of **RayFronts** by showing the segmentations of multiple long-tail classes.

A4. RAYFRONTS HYPERPARAMETERS

Table A.1 lists all the **RayFronts** hyperparameters and their descriptions, Table A.2 lists the hyperparameter values used for the online mapping evaluation, Table A.3 lists the hyperparameter values used for the offline mapping evaluation, and Table A.4 lists the hyperparameter values used for the throughput analysis.

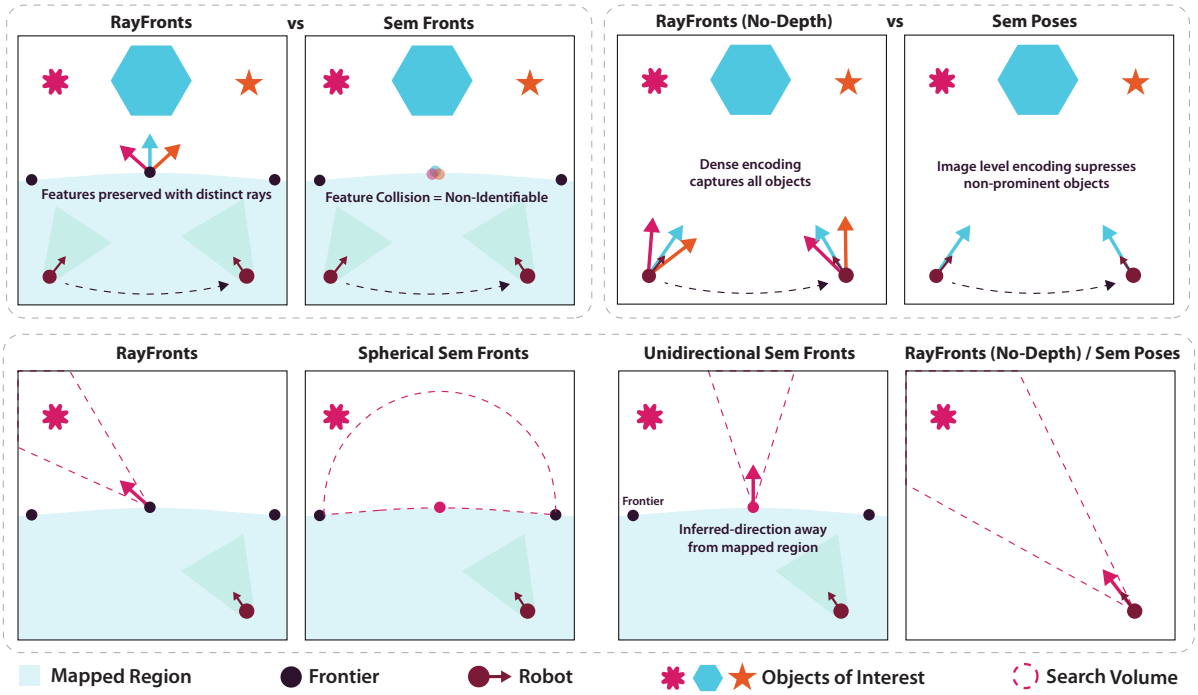

Fig. A.1: Top left shows how **RayFronts** is able to avoid feature collisions through the use of multiple rays that capture distinct semantics observed through the same frontier, where semantic frontier approaches [6], [7] fail. The top right illustrates that even with no depth information, **RayFronts** dense language-aligned encoding can allow it to capture non-prominent semantics where semantic pose approaches [8] fail. The bottom row highlights that **RayFronts** is the upper bound in accurately reducing search volume.

TABLE A.1: **RayFronts** Hyperparameter Descriptions.

parameter	description
backbone	Backbone model used.
resolution	Input RGB and depth resolution.
gauss_std (σ)	Standard deviation of the Gaussian kernel attention for the encoder.
vox_size (α)	Voxel size in meters.
fronti_neighborhood_r	Radius of the neighborhood to look at for computing frontiers.
fronti_min_unobserved($min_{unobsrv}$)	Min # of unobserved cells in the cell neighborhood to be considered a frontier.
fronti_min_occupied (min_{occ})	Min # of occupied cells in the cell neighborhood to be considered a frontier.
fronti_min_empty (min_{free})	Min # of free cells in the cell neighborhood to be considered a frontier.
fronti_subsampling	Subsampling factor of the frontier grid. $\beta = \alpha * fronti\_subsampling$.
fronti_subsampling_min_fronti	# of frontiers to lie in the coarser grid cell to be considered a frontier.
ray_erosion	Half size of the window to use when eroding the out-of-range mask M_t.
ray_tracing	Enable or disable ray tracing when propagating rays.
angle_bin_size (ψ)	Angle bin size used to discretize and aggregate rays within a frontier.
max_occ_cnt	Log-odds upper limit for occupancy.
max_empty_cnt	Log-odds lower limit for occupancy.
occ_observ_weight	How much to increment the log odds buffer with each occupied observation.
occ_thickness	Thickness of a projected occupied surface.
occ_pruning_tolerance	Tolerance of log-odds value to be merged in a super voxel in the VDB map.
max_dirs_per_frame	Max number of rays to cast per frame. (Uniformly samples to enforce).
max_pts_per_frame	Max number of occupied points to unproject. (Uniformly samples to enforce).
max_empty_pts_per_frame	Max number of empty points to unproject. (Uniformly samples to enforce).
vox_accum_period	How many frames should accumulate before aggregating voxels.
ray_accum_period	How many frames should accumulate before casting and aggregating rays.
ray_accum_phase	Ray accumulation delay to offset it from voxel accumulation.
stored_feat_dim	Dimension of map features. If less than encoder output, PCA is used to compress.
sem_pruning_period	How often to prune semantic voxels using the occupancy map.
occ_pruning_period	How often to prune the occupancy map (i.e merge large consistent areas).
prompt_denoising_thresh	Threshold for prompt denoising when classifying voxels/rays).
prediction_thresh	If the softmax value was lower than this threshold, no prediction will be made.
searchvol_thresh	Threshold to select the intersection of multiple search volumes.

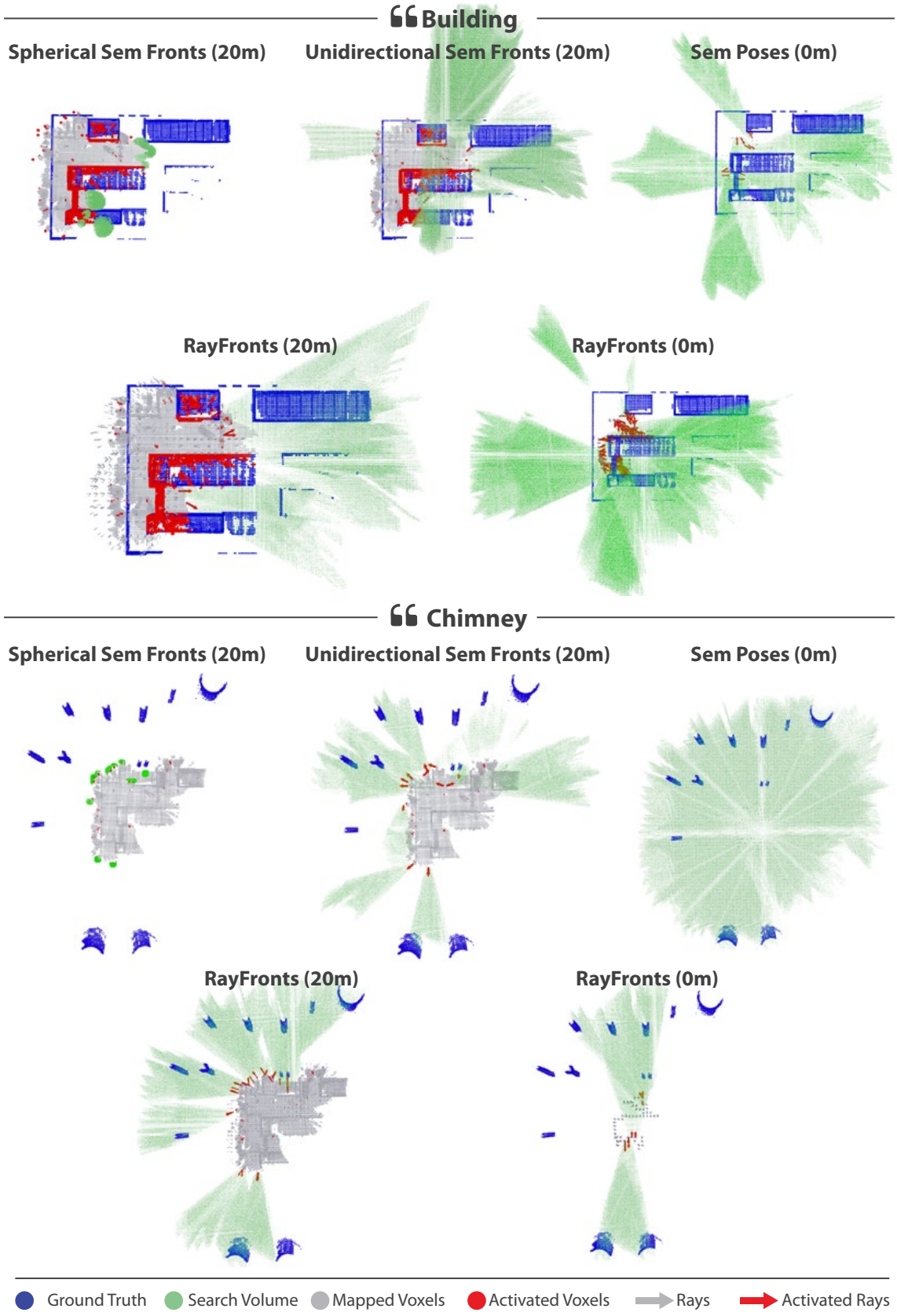

Fig. A.2: Two query scenarios are shown with GT generated at 80m as opposed to 50m cutoff for more clarity: (1) querying for a prominent object (i.e, Building) that enters depth range, and (2) a distant object (i.e, Chimney) that remains beyond range. Through unified dense voxel mapping and beyond-range semantic ray frontiers, **`RayFronts`** sets the upper bound in both scenarios.

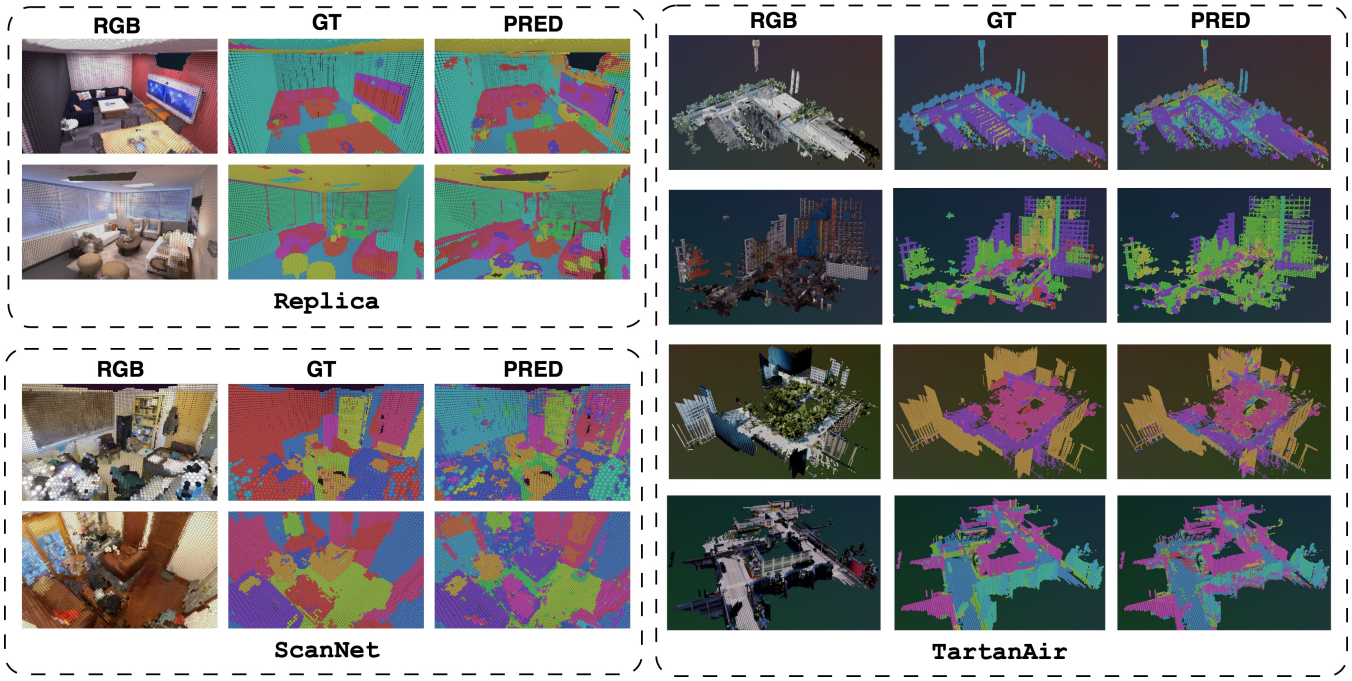

Fig. A.3: Sample visualizations of offline semantic mapping generated by **RayFronts** for scenes from Replica [37] (`room0` and `office2`), ScanNet [38] (`scene0050` and `scene00378`), and the chosen four scenes from TartanAir [34]. "RGB", "GT" and "PRED" refer to the RGB scene reconstruction, Ground Truth semantics, and semantic segmentation prediction by **RayFronts**, respectively, for each corresponding scene. **RayFronts** achieves SOTA mIoU for 3D open-vocabulary semantic segmentation.

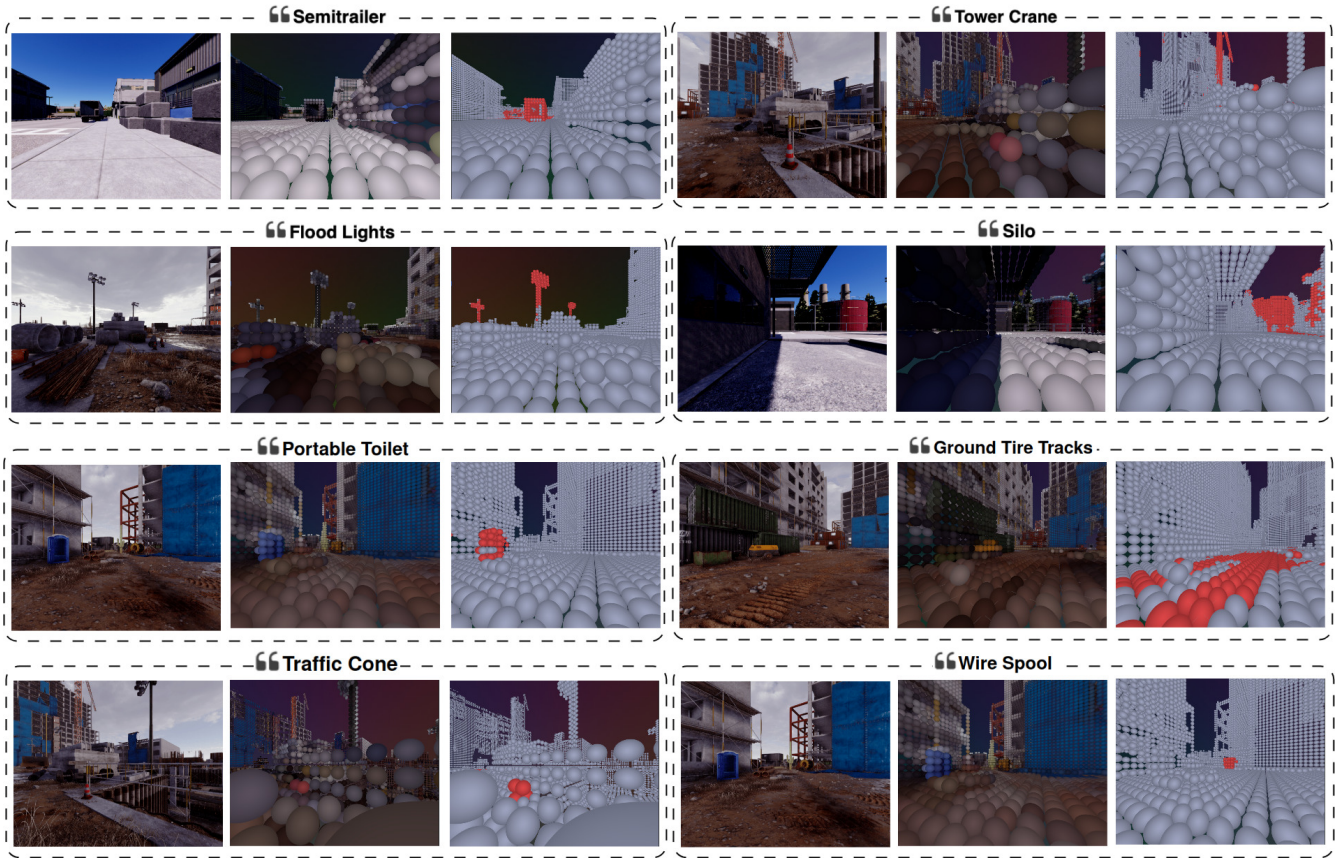

Fig. A.4: Examples of long-tail classes segmented by **RayFronts** across outdoor scenes from TartanAir [34]. We set the voxel size to 0.5 (50cm) for the visualizations. For each set, we present the RGB image, the corresponding 3D reconstructed view, and the classified voxels left to right respectively. **RayFronts** effectively segments long-tail concepts.

TABLE A.2: **RayFronts** Online Evaluation Hyperparameters.

parameter	value
backbone	radio_v2.5-l / SIGLIP
resolution	640x640
gauss_std (σ)	7.0
vox_size (α)	1.0
fronti_neighborhood_r	1
fronti_min_unobserved($min_{unobsrv}$)	9
fronti_min_occupied (min_{occ})	0
fronti_min_empty (min_{free})	4
fronti_subsampling	4
fronti_subsampling_min_fronti	5
ray_erosion	32
ray_tracing	True
angle_bin_size (ψ)	30°
max_occ_cnt	100
max_empty_cnt	-10
occ_observ_weight	100
occ_thickness	2
occ_pruning_tolerance	2
max_dirs_per_frame	10000
max_pts_per_frame	$+\infty$
max_empty_pts_per_frame	$+\infty$
stored_feat_dim	768
prompt_denoising_thresh	0.5
prediction_thresh	0.1
searchvol_thresh	0.05

TABLE A.3: **RayFronts** Offline Evaluation Hyperparameters.

parameter	value
backbone	radio_v2.5-l / SIGLIP
resolution	640x480
gauss_std (σ)	7.0
vox_size (α)	0.05, 1.0
stored_feat_dim	768
prompt_denoising_thresh	0.5
prediction_thresh	0.1

TABLE A.4: **RayFronts** Mapping Throughput Hyperparameters.

parameter	value
backbone	radio_v2.5-b / SIGLIP
resolution	224x224
gauss_std (σ)	7.0
vox_size (α)	0.3
fronti_neighborhood_r	1
fronti_min_unobserved($min_{unobsrv}$)	9
fronti_min_occupied (min_{occ})	0
fronti_min_empty (min_{free})	4
fronti_subsampling	4
fronti_subsampling_min_fronti	5
ray_erosion	0
ray_tracing	False
angle_bin_size (ψ)	30°
max_occ_cnt	100
max_empty_cnt	-10
occ_observ_weight	100
occ_thickness	1
occ_pruning_tolerance	2
max_dirs_per_frame	1000
max_pts_per_frame	3000
max_empty_pts_per_frame	10000
stored_feat_dim	100
vox_accum_period	8
ray_accum_period	8
ray_accum_phase	4
sem_pruning_period	32
occ_pruning_period	32

REFERENCES

[1] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary queryable scene representations for real world planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11509–11522, IEEE, 2023. 1, 2

[2] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha, *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *arXiv preprint arXiv:2302.07241*, 2023. 1, 2, 3, 5, 6, 7

[3] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation," in *Proceedings of Robotics: Science and Systems*, (Delft, Netherlands), July 2024. 1, 2, 3, 5, 6, 7

[4] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028, IEEE, 2024. 1, 2, 5, 6, 7

[5] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, "Openscene: 3d scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 815–824, 2023. 1, 2

[6] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42–48, IEEE, 2024. 2, 5, 9

[7] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, "How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers," 2023. 2, 5, 9

[8] H. Thomas, M. Sivapurapu, and J. Zhang, "Embedding pose graph, enabling 3d foundation model capabilities with a compact representation," *arXiv preprint arXiv:2403.13777*, 2024. 2, 4, 5, 9

[9] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021. 2

[10] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023. 2, 3

[11] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *CVPR*, pp. 4015–4026, 2023. 2, 3

[12] 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*, pp. 8748–8763, PMLR, 2021. 2, 3

[13] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *CVPR*, pp. 11975–11986, 2023. 2

[14] J. Wu, X. Li, S. Xu, H. Yuan, H. Ding, Y. Yang, X. Li, J. Zhang, Y. Tong, X. Jiang, *et al.*, "Towards open vocabulary learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 7, pp. 5092–5113, 2024. 2

[15] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European Conference on Computer Vision*, pp. 38–55, Springer, 2024. 2

[16] M. Ranzinger, G. Heinrich, J. Kautz, and P. Molchanov, "Am-radio: Agglomerative vision foundation model reduce all domains into one," in *CVPR*, pp. 12490–12500, 2024. 2, 3

[17] S. Hajimiri, I. B. Ayed, and J. Dolz, "Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation," *arXiv preprint arXiv:2404.08181*, 2024. 2, 3, 6, 7

[18] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *2017 IEEE International Conference on Robotics and automation (ICRA)*, pp. 4628–4635, IEEE, 2017. 2

[19] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE international symposium on mixed and augmented reality (ISMAR)*, pp. 10–20, IEEE, 2018. 2

[20] 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, pp. 3037–3044, 2019. 2

[21] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *2018 international conference on 3D vision (3DV)*, pp. 32–41, IEEE, 2018. 2

[22] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5231–5237, IEEE, 2019. 2

[23] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," *arXiv preprint arXiv:2201.13360*, 2022. 2

[24] C. Ho, J. Zou, O. Alama, S. M. J. Kumar, B. Chiang, T. Gupta, C. Wang, N. Keetha, K. Sycara, and S. Scherer, "Map it anywhere (mia): Empowering bird's eye view mapping using large-scale public data," *arXiv preprint arXiv:2407.08726*, 2024. 2

[25] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, H.-S. Fang, *et al.*, "Toward general-purpose robots via foundation models: A survey and meta-analysis," *arXiv preprint arXiv:2312.08782*, 2023. 2

[26] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna, and S. Garg, "Anyloc: Towards universal visual place recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1286–1293, 2023. 2

[27] Q. Xie, S. Y. Min, T. Zhang, K. Xu, A. Bajaj, R. Salakhutdinov, M. Johnson-Roberson, and Y. Bisk, "Embodied-rag: General non-parametric embodied memory for retrieval and generation," *arXiv preprint arXiv:2409.18313*, 2024. 2

[28] C. Kassab, M. Mattamala, L. Zhang, and M. Fallon, "Language-extended indoor slam (lexis): A versatile system for real-time visual scene understanding," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15988–15994, IEEE, 2024. 2

[29] K. Museth, J. Lait, J. Johanson, J. Budsberg, R. Henderson, M. Alden, P. Cucka, D. Hill, and A. Pearce, "Openvdb: an open-source data structure and toolkit for high-resolution volumes," in *ACM SIGGRAPH 2013 Courses*, SIGGRAPH '13, (New York, NY, USA), Association for Computing Machinery, 2013. 3, 4

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 3

[31] G. Best, R. Garg, J. Keller, G. A. Hollinger, and S. Scherer, "Resilient multi-sensor exploration of multifarious environments with a team of aerial robots," in *Robotics: Science and Systems (RSS)*, 2022. 4

[32] R. Hagmanns, T. Emter, M. Grosse-Besselmann, and J. Beyerer, "Efficient global occupancy mapping for mobile robots using openvdb," 2022. 4

[33] S. Kim, M. Corah, J. Keller, G. Best, and S. Scherer, "Multi-robot multi-room exploration with geometric cue extraction and circular decomposition," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1190–1197, 2023. 4

[34] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "Tartanair: A dataset to push the limits of visual slam," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916, IEEE, 2020. 5, 6, 7, 11

[35] Y. Shi, M. Dong, and C. Xu, "Harnessing vision foundation models for high-performance, training-free open vocabulary segmentation," *arXiv preprint arXiv:2411.09219*, 2024. 6, 7

[36] C. Zhou, C. C. Loy, and B. Dai, "Extract free dense labels from clip," in *European Conference on Computer Vision*, pp. 696–712, Springer, 2022. 6

[37] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019. 7, 11

[38] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017. 7, 11