RAVEN: Resilient Aerial Navigation via Open-Set Semantic Memory and Behavior Adaptation

Seungchan Kim[1], Omar Alama[1], Dmytro Kurdydyk[2], John Keller[1], Nikhil Keetha[1], Wenshan Wang[1], Yonatan Bisk[1], Sebastian Scherer[1]

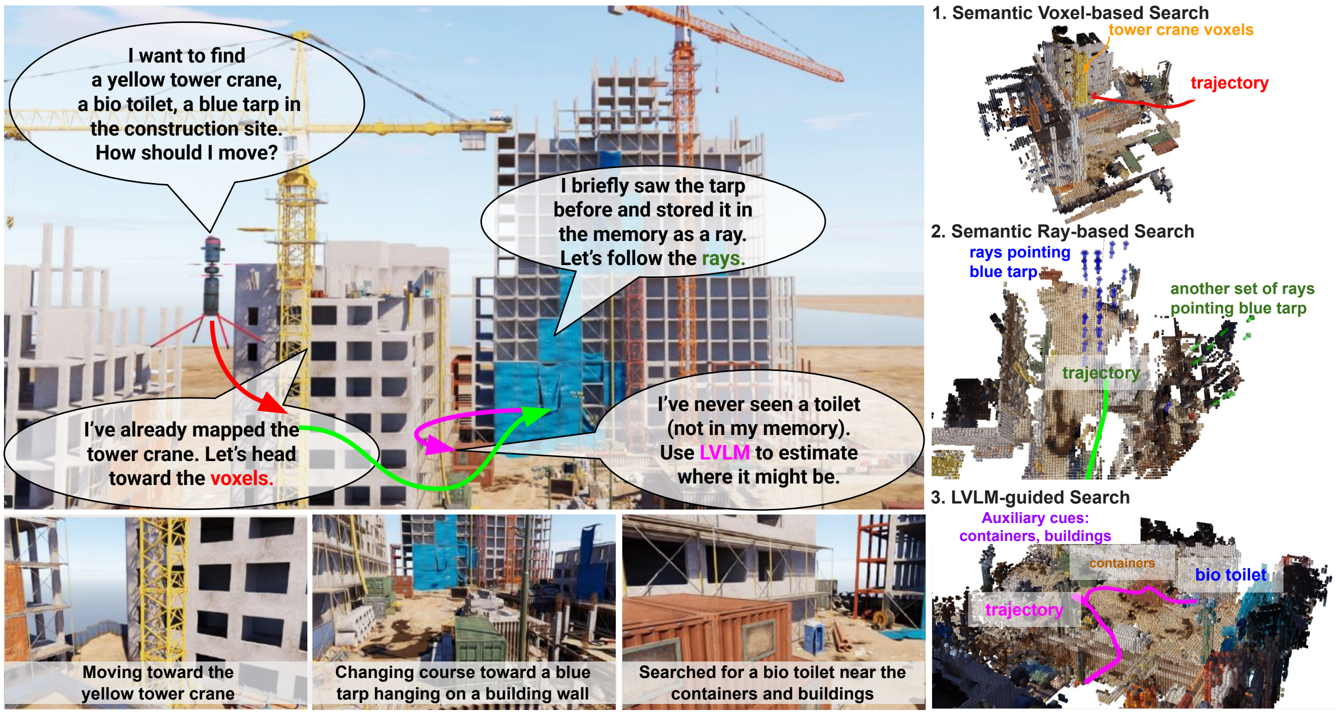

Fig. 1: We present **RAVEN**, a 3D open-set memory-based behavior tree framework for aerial semantic outdoor navigation. **RAVEN** not only navigates reliably toward detected targets, but also performs long-range reasoning to plan toward distant cues and continues informed search even when direct visual evidence is limited. It further supports multi-class object search and on-the-fly task switching within a mission.

***Abstract*—Aerial outdoor semantic navigation requires robots to explore large, unstructured environments to locate target objects. Recent advances in semantic navigation have demonstrated open-set object-goal navigation in indoor settings, but these methods remain limited by constrained spatial ranges and structured layouts, making them unsuitable for long-range outdoor search. While outdoor semantic navigation approaches exist, they either rely on reactive policies based on current observations, which tend to produce short-sighted behaviors, or precompute scene graphs offline for navigation, limiting adaptability to online deployment. We present RAVEN, a 3D memory-based, behavior tree framework for aerial semantic navigation in unstructured outdoor environments. It (1) uses a spatially consistent semantic voxel-ray map as persistent memory, enabling long-horizon planning and avoiding purely reactive behaviors, (2) combines short-range voxel search and long-range ray search to scale to large environments, (3) leverages a large vision-language model to suggest auxiliary cues, mitigating sparsity of outdoor targets. These components are coordinated by a behavior tree, which adaptively switches behaviors for robust operation. We evaluate RAVEN in 10 photorealistic outdoor simulation environments over 100 semantic tasks, encompassing single-object search, multi-class, multi-instance navigation and sequential task changes. Results show RAVEN outperforms baselines by 85.25% in simulation and demonstrate its real-world applicability through deployment on an aerial robot in outdoor field tests.**

I. INTRODUCTION

Aerial robots are increasingly employed for autonomous exploration and search in outdoor environments. Prior research has largely focused on geometry-driven strategies, including frontier-based exploration [1] and volumetric information gain maximization [2]. While these methods efficiently cover the spaces, they rely on geometric representations–such as occupancy grids or volumetric maps–and cannot perform semantic reasoning, for example, locating a yellow car in urban areas or identifying machinery scattered across a construction site.

Semantic navigation has been extensively studied in indoor environments [3–5], where robots are tasked with reaching object goals or following language instructions. With the advent of foundation models, recent approaches have leveraged

[1] Authors are with Carnegie Mellon University, Pittsburgh, PA, USA. {seungch2, oalama, jkeller2, nkeetha, wenshanw, ybisk, basti}@andrew.cmu.edu
[2] Author is with Davidson College, Davidson, NC, USA. dmkurdydyk@davidson.edu

these models for open-set object navigation [6–8]. However, such methods remain largely confined to bounded 2D indoor settings [3–8], where the spatial horizon is limited, the robot's range of possible movements is constrained, and the structured environment provides cues for locating objects.

By contrast, outdoor semantic navigation poses two major challenges compared to indoor settings. First, the much larger spatial extent of outdoor scenes requires long-range search strategies; second, the sparse distribution of target objects, combined with the absence of a structured hierarchy (e.g., building $\rightarrow$ floor $\rightarrow$ room $\rightarrow$ object) demands careful strategic planning. Some prior methods [9–11] pursue map-free navigation with reactive policies that leverage short-term observation histories, which often generate short-sighted behaviors. Other works attempt semantic navigation in outdoor scenes using graph representations; however, they construct these graphs offline, which limits their online applicability [12, 13].

We propose **RAVEN**, a **R**esilient **A**erial **V**oxel-Ray Memory **E**mpowered **N**avigation, a novel framework for outdoor semantic search. (1) **RAVEN** builds upon recent advances in open-set semantic voxel and ray representations [14], leveraging them as *persistent internal memory*. This open-set voxel-ray memory not only avoids purely reactive behaviors and enables long-horizon planning, but also supports multi-class search and accommodates dynamic task switching, owing to its task-agnostic nature. (2) Beyond voxel-based search for close and reliable cues, **RAVEN** performs ray-based search to register and plan over long-range distance cues, thereby addressing the challenge of long-range reasoning in outdoor settings. (3) To overcome the semantic sparsity of outdoor scenes, it invokes a large vision language model (LVLM)-guided search when needed to incorporate auxiliary cues. These search behaviors are integrated through a behavior tree, which allows the system to adaptively select behaviors depending on situations and to maintain resiliency by switching strategies when one fails. We validate **RAVEN** in extensive simulation environments, and demonstrate its promising performance through real-robot deployment testing.

In summary, our contributions are as follows:

- We present a new aerial semantic navigation framework that leverages an open-set voxel–ray memory as an internal representation of the world and employs a behavior tree to robustly adapt across multiple search behaviors.
- We address the challenges of long-range reasoning and sparse semantic cues via ray-based search and LVLM guidance within the behavior tree. Our method further supports multi-class search and on-the-fly task switching, an underexplored aspect of semantic navigation.
- We validate the framework by demonstrating superior performance over existing baselines across diverse tasks in simulation and further showcase its real-world feasibility through deployment on a physical robot.

II. RELATED WORKS

Mobile robot autonomy spans diverse tasks, from exploration to navigation, and building on these foundations, recent works have explored the semantic navigation paradigm [3–5], where robots understand and reason about semantic information in the environment, such as object types, language, and context, to guide their navigation.

With the rise of Vision Foundation Models (VFMs) [15] and LVLMs [16], semantic navigation has seen tremendous growth [17, 18]. Tasks in semantic navigation include vision-language navigation (VLN) [17, 19, 20], where a robot executes a vision-grounded language instruction to follow paths; object-goal navigation (OGN) [3, 6, 21], where a robot locates and navigates toward objects based on their categories or descriptions; embodied question answering (EQA) [18], where a robot actively explores to answer questions about the environment.

In this work, we focus on object-goal navigation. While prior studies have primarily focused on indoor, ground robots, and single-object scenarios, we aim to explore OGN in outdoor settings, with aerial robots, and multi-class, multi-object scenarios–an important yet widely underexplored domain.

A. Classical Exploration and Search

Early work on robotic exploration began with frontier-based methods using 2D occupancy grids [22]. Extensions to 3D aerial scenarios include frontier-based fast exploration [1] and sampling-based approaches driven by information gain [2]. While these methods have shown strong performance in real-world deployments, their primary focus is on mapping and covering unknown areas, often neglecting semantic information. Recent efforts [23] employ aerial robots to explore regions and attempt to detect target objects. However, they treat object discovery merely as a byproduct of map coverage rather than performing object-goal-driven planning. In contrast, **RAVEN** leverages open-set semantic voxel-ray memory and LVLM to guide planning, while retaining the robustness of traditional methods when semantic guidance is insufficient.

B. Map-free Semantic Navigation

Building on the success of vision language models (VLMs) and multi-modal LLMs, recent works explored zero-shot semantic navigation that uses semantic priors to interpret observations and directly generate control commands. Early approaches relied on contrastive VLMs such as CLIP [15] to sequentially guide the robot [7]. Later works leverage LLMs' reasoning capabilities [24] and emphasize incorporating memory to go beyond reactive planning, e.g., by buffering past FPV frames [10, 20] or maintaining textual summaries of past observations and actions [17]. This map-free paradigm has also been applied to VLN [9, 10] and OGN [11] in aerial robotics.

However, the FPV-based map-free paradigm typically produces discrete control commands, bypassing motion continuity and showing limitations in collision avoidance. Moreover, relying on latest frames or heuristically selected history suffers from information loss [17], limiting applicability to long-horizon planning. In contrast, our approach leverages a 3D spatially grounded open-set semantic memory that efficiently aggregates in-range and out-of-range observations into a persistent internal representation, enabling long-horizon planning.

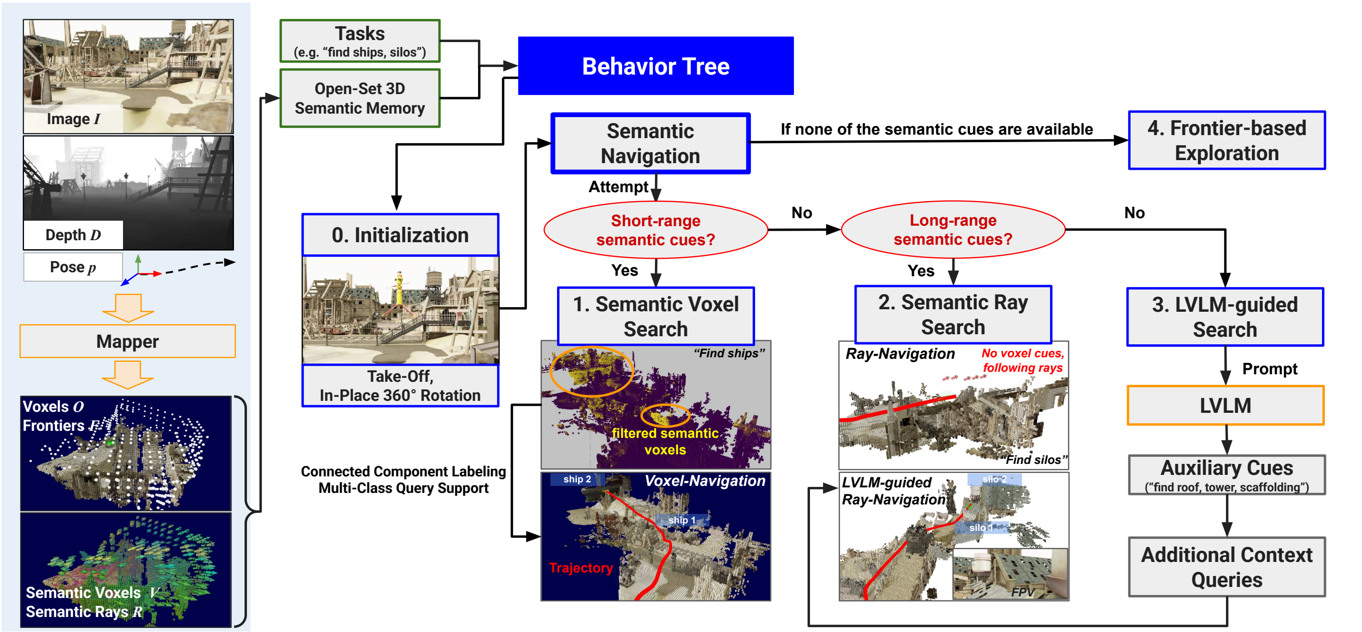

Fig. 2: Overview of **RAVEN**. From image, depth, and pose inputs, the mapper builds an open-set 3D semantic voxel-ray map that serves as persistent memory. A behavior tree adapts the robot's actions: it performs semantic voxel-based search when reliable cues exist within depth range, switches to semantic ray-based search when only long-range directional hints are available, invokes an LVLM to suggest auxiliary objects when no target is visible, and defaults to frontier-based exploration if all strategies fail.

C. Map-based Semantic Navigation

To address limitations of LLMs' spatial reasoning [25] and support longer-term observation storage, many works adopt spatial mapping approaches. In these methods, semantic information from observations is encoded and aggregated into either: (1) dense representations, such as 2D bird's eye view (BEV) grids–often combined with semantically scored frontier maps [3–6]–or 3D voxel grids [14, 26]; or (2) graph-based representations, including scene graphs that associate semantics with scene entities [27–29].

Graph-based approaches have a lower memory footprint but are often computationally expensive to construct, with many generated offline before use [12, 13, 26–29]. Furthermore, their heuristic-driven graph structures typically assume regular indoor layouts, making them difficult to generalize to outdoor environments [30]. Similarly, 2D BEV-based approaches [3–6, 8], while capable of online guidance, are largely applied to indoor ground-robot scenarios, assuming constrained spatial ranges and hierarchical indoor structures.

Recently, RayFronts [14] introduced a real-time, online mapping framework that generates a dense semantic representation using a single forward pass. It encodes in-range observations as voxels and out-of-range observations as rays, enabling object localization given a **fixed trajectory** and **explicit queries**. We leverage this representation as a memory to **actively guide** an aerial robot toward goal objects and **generate reasoning-based queries** via an LVLM, enabling the system not only to navigate toward targets, but also to strategically identify auxiliary cues to support efficient search.

III. METHOD

The core of **RAVEN** is a behavior tree that adapts search behaviors, leveraging a 3D open-set semantic voxel-ray representation as a memory. We first provide a brief overview of the voxel-ray mapping, and then focus on behavior modules in detail. The overall pipeline is illustrated in Figure 2.

A. Preliminary: Open-Set Semantic Voxel-Ray Mapping

We build upon the encoder and the open-set 3D voxel and ray mapping pipeline from RayFronts [14]. Specifically, at each timestep t the robot receives an RGB image I_t and depth measurement D_t. The encoder E processes the image to produce vision-language aligned features $\mathbf{f}_t = E(I_t)$. From the depth sensor, we construct a set of 3D occupancy voxels O_t and detect frontier regions $\mathcal{F}_t$. The image features are projected onto the voxel grid, resulting in semantic voxels:

$$V_t = \textsc{Project}(\mathbf{f}_t, O_t). \tag{1}$$

In parallel, we cast outward rays from $\mathcal{F}_t$, and similarly project the features $\mathbf{f}_t$ onto them, yielding semantic rays:

$$R_t = \textsc{RayProject}(\mathbf{f}_t, \mathcal{F}_t). \tag{2}$$

Rays serve to encode semantic information beyond the depth coverage, complementing V_t. Unlike a frontier $f \in \mathcal{F}_t$, which is represented as a single 3D point, each ray $r_i \in R_t$ has its own 3D origin and *directional* vector. Multiple rays may share the same frontier as their origin, with each ray pointing a different direction. These directions are computed via image projection, preserving accurate orientation. Rays also provide a much sparser representation than the full set of image observations, enabling efficient encoding of distant semantic cues.

B. ***RAVEN***: *Behavior Tree for Resilient Adaptation*

This subsection presents our main contribution: a set of modular behaviors for 3D aerial outdoor semantic navigation. We describe each module, and conclude by explaining how they are integrated into a single behavior tree architecture.

Initialization: At the beginning of each episode, the robot takes off from its starting location and ascends to 5m. As in [6], it also performs a 360° rotation in its place to obtain a holistic view of the environment. This step ensures a consistent start across trials and sufficient altitude for safe exploration.

Frontier-based Exploration: The aerial robot defaults to frontier-based exploration when no relevant semantic cues are available. Frontiers $\mathcal{F}_t$ are clustered using DBSCAN with parameters ε and *min_samples*, yielding a set of frontier centroids C. Each centroid is scored by a weighted combination of its distance from the robot's position and a momentum-based penalty for sharp heading changes. The centroid with the lowest score is then selected as the next waypoint.

Semantic Voxel-based Search: As the robot explores, information captured in the RGB images and falls within the depth range is stored in semantic voxels V_t. Each voxel $v \in V_t$ contains a semantic feature $\mathbf{f}(v)$ produced by RayFronts encoder. For comparison with target object classes, we adapt these features into the SIGLIP embedding [31] space. For a given task, we compute the cosine similarity between SIGLIP-adapted voxel feature $\tilde{\mathbf{f}}(v)$ and the SIGLIP embedding of each target object class query $\mathbf{q}_j$ (e.g., SIGLIP("water tower")):

$$s(v, \mathbf{q}_j) = \frac{\tilde{\mathbf{f}}(v) \cdot \mathbf{q}_j}{\|\tilde{\mathbf{f}}(v)\| \, \|\mathbf{q}_j\|}, \quad j = 1, \dots, J \tag{3}$$

where J is number of object classes in the task. Because multiple text queries can be evaluated in parallel, this formulation naturally supports multi-class navigation. Voxels exceeding a similarity threshold ε_{vox} are retained:

$$V_{\text{filtered}} = \{ v \in V_t \mid \exists \ j \quad s.t. \ s(v, \mathbf{q}_j) > \varepsilon_{\text{vox}} \} \tag{4}$$

To approximate object-level regions, connected component labeling (CCL) is applied to V_{filtered}, grouping adjacent voxels into clusters while filtering out small outliers.

$$\{C_1, \dots, C_N\} \leftarrow \text{CCL}(V_{\text{filtered}}) \qquad s.t. \ |C_n| \geq \tau_{\min} \tag{5}$$

These clusters serve as candidate object hypotheses and are visited sequentially, starting from the closest. For safe navigation, a global waypoint is set just outside the cluster's bounding box, offset outward by Ω.

Semantic Ray-based Search: While V_t stores information about nearby objects, outdoor navigation often requires reasoning about objects that lie far beyond the depth range. To capture such long-range cues, we employ semantic rays R_t, which encodes information about distant objects that are briefly visible in the RGB images but outside depth coverage.

A set of rays $R_t = \{r_1, .., r_N\}$ is cast outward in multiple directions beyond the current frontiers $\mathcal{F}_t$. Each ray r_i is associated with semantic features $\mathbf{f}(r_i)$, which is adapted to SIGLIP embeddings $\tilde{\mathbf{f}}(r_i)$. For a given task, we compute cosine similarity between the ray features $\tilde{\mathbf{f}}(r_i)$ and the SIGLIP embedding features of the text queries $\mathbf{q}_j$ corresponding to the target objects:

$$s(r_i, \mathbf{q}_j) = \frac{\tilde{\mathbf{f}}(r_i) \cdot \mathbf{q}_j}{\|\tilde{\mathbf{f}}(r_i)\| \|\mathbf{q}_j\|} \tag{6}$$

Rays with similarity above a threshold ε_{ray} form the subset:

$$R_{\text{filtered}} = \{ r_i \in R_t \,|\, \max_j s(r_i, \mathbf{q}_j) \geq \varepsilon_{\text{ray}} \} \tag{7}$$

When $R_{\text{filtered}} \neq \emptyset$, ray-based search is initiated. The rays in R_{filtered} are grouped into angular bins $\{B_1, B_2, ..., B_K\}$. Each group B_k is scored using a weighted combination of proximity $\phi_{\text{prox}}(B_k)$ and density $\phi_{\text{dens}}(B_k)$:

$$Score(B_k) = \alpha \cdot \phi_{\text{prox}}(B_k) + \beta \cdot \phi_{\text{dens}}(B_k) \tag{8}$$

The robot selects the highest-scoring B_k as a global waypoint.

As the robot follows selected rays and approaches the object, semantic voxels naturally begin to form, at which point the system transitions smoothly into voxel-based search for more precise localization (Fig. 3). This ray-based search enables long-range reasoning beyond what voxels alone can achieve and provides additional benefit–it introduces a persistent memory: once observed, an object's information remains encoded in the rays even if it is no longer visible in the current image. This motivates our next discussion on memory.

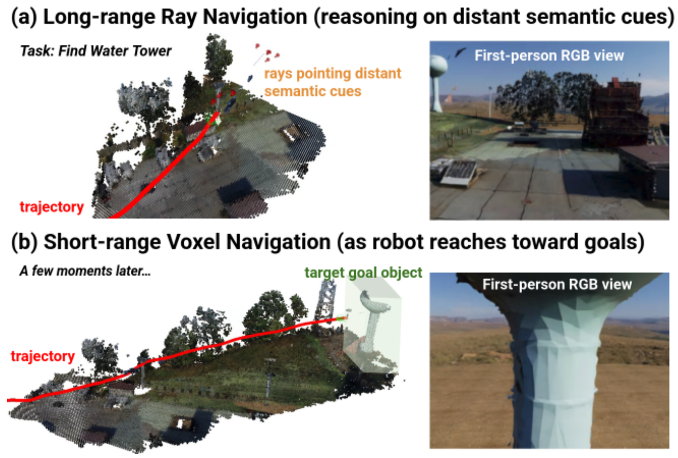

Fig. 3: (a) When no information exists within depth range, the robot activates semantic ray navigation for long-range coarse search, using ray features where the tower was briefly captured in the RGB view. (b) As it follows the rays and approaches the target, voxels naturally form and the search transitions to precise voxel-based navigation.

Task-Agnostic 3D Voxel-Ray Memory: Together, the semantic voxels V_t and semantic rays R_t constitute a unified spatial-semantic memory that accumulates observations over time without overwriting earlier data. This persistent representation explicitly encodes both geometry and semantics, enabling long-horizon planning in large, unstructured environments.

A key feature of our approach is that the voxel-ray memory is task-agnostic, distinguishing it from approaches that rely on task-conditioned 3D maps. In task-conditioned methods, even open-vocabulary models still require a predefined set of

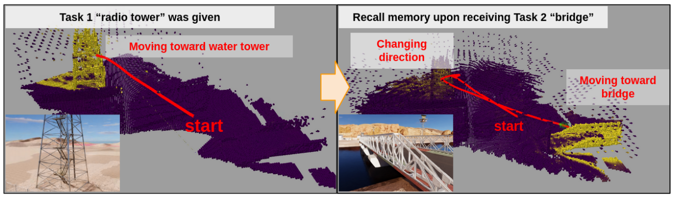

Fig. 4: Open-set semantic voxel-ray map serves as memory. When the first task (“radio tower”) is given, the robot navigates toward it. Upon introduction of a second task (“bridge”), it queries the existing memory and uses highlighted cues to guide search. Yellow voxels and rays indicate task-relevant highlights.

tasks (object types or queries) to guide the mapping process. In contrast, our memory is task-agnostic, meaning that no predefined queries are needed during mapping, and the task can even change on the fly. When a new task arises, the robot can align the new query with the existing voxel-ray memory using a similarity score to select relevant voxels and rays.

This property makes our method particularly suitable for online mapping. As illustrated in Figure 4, while the robot is creating a semantic voxel-ray memory to locate the first task object (e.g., a radio tower), it can seamlessly handle a new task introduced midway (e.g., a bridge) without rebuilding the map. Instead, it simply compares the new query to the existing memory and identifies the most relevant voxels and rays.

LVLM-guided Search: While semantic voxels and rays capture information within and beyond the depth sensor range, both can fail if the target objects never appear, even briefly, in any RGB images. To handle such cases, we introduce an additional search strategy guided by an LVLM. At sparse intervals T_{lvlm}, the LVLM is queried with the current image I_t and a prompt $\mathcal{P}$, producing auxiliary objects aux_j that are semantically related to the targets.

$$\{aux_j\}_{j=1}^{J_{\mathrm{aux}}} \leftarrow \mathrm{LVLM}(\mathcal{P}, I_t) \tag{9}$$

These auxiliary cues are converted into additional text queries and fused into the ray-based search, enabling the robot to pursue contextual objects as guidance toward the target.

$$\begin{gathered} Q = \{\mathbf{q}_j\}_{j=1}^{J} \cup \{\mathbf{q}_{aux_j}\}_{j=1}^{J_{\mathrm{aux}}} \\ s(r_i, \mathbf{q}_j) = \frac{\tilde{\mathbf{f}}(r_i) \cdot \mathbf{q}_j}{\|\tilde{\mathbf{f}}(r_i)\| \, \|\mathbf{q}_j\|}, \quad r_i \in R, \ \mathbf{q}_j \in Q \end{gathered} \tag{10}$$

The remaining procedure follows the standard ray-based search strategy. We employ InternVL3-2B [16] as the LVLM.

Behavior Tree Integration: To coordinate the navigation modules, we use a behavior tree (BT) that structures modular behaviors and ensures robust execution. After initialization, control proceeds through a priority-based sequence:

- **Semantic voxel-based search:** prioritized first, as nearby objects within depth range provide the most reliable cues.
- **Semantic ray-based search:** used when voxel cues are insufficient, using distant observations for long-range coarse guidance.
- **LVLM-guided search:** provides auxiliary cues from sparse LVLM queries when needed.

Fig. 5: Ten simulation environments used for evaluation.

- **Frontier-based exploration:** fallback strategy to maintain coverage of unexplored regions when semantic cues are unavailable.

This modular BT design offers two key benefits: (i) it enables autonomous selection of the most suitable behavior for each state, and (ii) it guarantees resilience and robustness by sequentially activating alternative strategies when one fails.

C. Local Trajectory Planning

For local trajectory planning and collision avoidance, we use DROAN [32], a disparity-based algorithm that inflates the configuration space around detected obstacles in the disparity image. The local planner selects the best path from a library of candidate trajectories and, given a global waypoint, generates a collision-free trajectory toward the target.

IV. SIMULATION EXPERIMENTS

A. Simulator, Environments, and Experiment Setup

We conduct photorealistic robot simulation using NVIDIA Isaac Sim. Since no standard benchmark exists for outdoor semantic navigation, we design ten environments: one digital twin of a real-world site and nine designed simulated scenes. To introduce a variety of conditons, each environment is tested with three starting poses. The ten simulation environments are shown in Fig. 5. We deploy our custom aerial robot autonomy stack in these environments. The robot model is based on the Spirit platform from Ascent Aerosystems, and all experiments are run on an NVIDIA RTX 6000 Ada GPU.

B. Tasks

Unlike prior works that typically consider single-OGN tasks, our task formulation generalizes to multi-class, multi-instance settings, as well as sequential task scenarios.

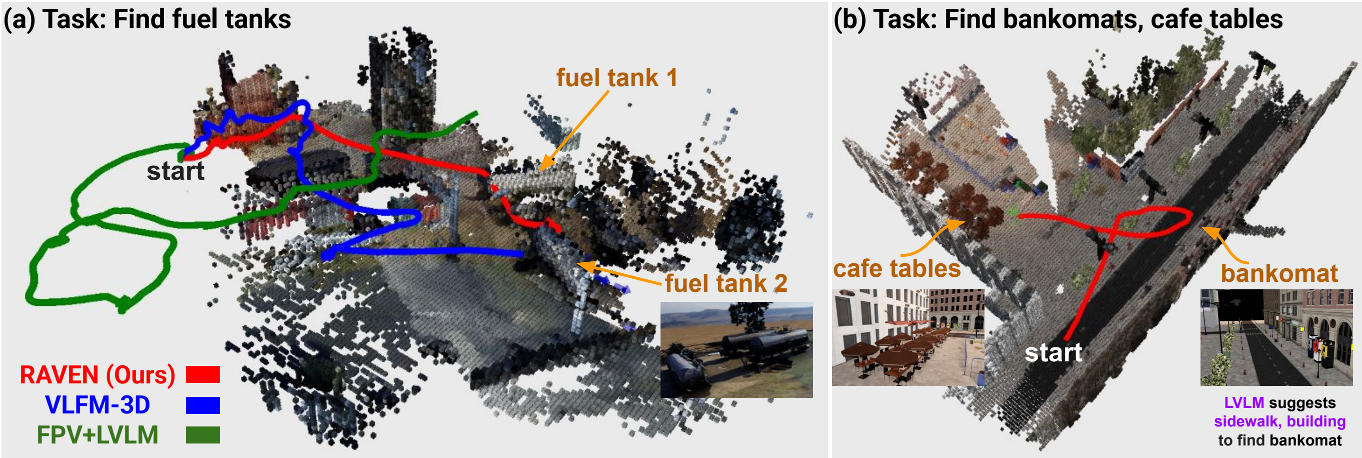

Fig. 6: (a) Trajectories of **RAVEN** (red), VLFM-3D (blue), FPV+LVLM (green) for finding fuel tanks. FPV+LVLM relies on recent views and reactive policies, producing a meandering path. VLFM-3D uses a value map but chases momentary maxima, yielding myopic behavior. **RAVEN** leverages semantic rays for long-range reasoning, generating an efficient path. (b) **RAVEN** trajectory for locating cafe tables and bankomats. Starting with empty memory and failing voxel-ray search, the robot invokes LVLM, which suggests sidewalks and building as auxiliary cues for bankomats. Following rays pointing sidewalk, it finds the bankomats, turns back, spots and navigates to the cafe tables.

Type I: Single-Class Tasks: All target instances belong to a single class $c \in \mathcal{C}$:

$$\mathcal{G} = \{g_m \mid g_m \text{ of class } c,\, m = 1, \ldots, M\}, \quad c \in \mathcal{C}$$

This includes either a single instance ($M = 1$) or multiple instances ($M > 1$) of that class.

Type II: Multi-Class Tasks: Target instances span multiple classes $\mathcal{C}_{\text{target}} \subseteq \mathcal{C}$:

$$\mathcal{G} = \bigcup_{c \in \mathcal{C}_{\text{target}}} \{g_m \mid g_m \text{ of class } c\}$$

The agent must navigate to all instances in these classes.

Type III: Sequential Dual-Class Tasks: Target classes are revealed sequentially. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ denote the first and second class instances, respectively. The robot must first find at least one instance of $\mathcal{G}_1$; once found, $\mathcal{G}_2$ is revealed.

$$\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2, \quad \text{with } \mathcal{G}_2 \text{ revealed after first } \mathcal{G}_1 \text{ is found}$$

This task evaluates the robot's ability to exploit internal maps and past observations to quickly search for the second class.

C. Metrics

Single-OGN is commonly evaluted by *Success Rate (SR)*–whether the robot reaches the goal within budget–and its path-efficiency variant *Success weighted by Path Length (SPL)* [3, 5, 6]. In multi-OGN tasks, however, a binary success criterion fails to capture partial completion. We therefore adopt the *Progress* and *Progress weighted by Path Length (PPL)* metrics, introduced in [21].

Progress: Let a task specify M target object goals $\mathcal{G} = \{g_1, \ldots, g_M\}$ (each instance counts separately). If the robot reaches K of them within the budget, we define

$$\text{Progress} = \frac{K}{M}.$$

In the single-object case, this reduces to the standard SR.

Progress weighted by Path Length (PPL): Let p be the length of the executed trajectory up until the K-th distinct goal is reached (set PPL$=0$ if $K = 0$). PPL normalizes progress by trajectory length p and compares it to the optimal path length d_K visiting the best subset of K goals in the best order.

$$\text{PPL} = \frac{d_K}{p} \cdot \frac{K}{M}.$$

In single OGN, PPL reduces to standard SPL.

D. Baselines

We select three baseline categories for comparison: one representing classical exploration and search, one representing map-free semantic navigation, and one representing map-based semantic navigation as the state-of-the-art baseline.

- Frontier-3D [23]: A classical frontier-based exploration method, extended to 3D aerial robot settings.
- FPV+LVLM: A map-free semantic navigation approach that maintains a short history of recent FPV frames and prompts LVLM [16] to output discrete action commands.
- VLFM-3D [6]: A 3D extension of the SOTA 2D BEV online OGN method, VLFM. The image encoder and local planning are adopted from our approach.

We also ablate the impact of each component of our method:

- Semantic Voxels: Excludes rays and LVLM to assess the effect of ignoring long-range observation encoding
- Semantic Rays: Excludes voxels and LVLM to assess the impact of lacking precise in-range localization
- Semantic Voxels + Rays: Uses both voxels and rays but no LVLM to measure the benefit of auxiliary cues.

E. Comparison to Baseline Methods

We compare the performance of **RAVEN** with all baselines. **RAVEN** achieves the best performance across all three task

TABLE I: Comparison of **RAVEN** with baselines. Progress and PPL are reported for each task type.

	Task Type I (40 tasks)		Task Type II (30 tasks)		Task Type III (30 tasks)		All Tasks (100 tasks)	
	Progress(%)	PPL(%)	Progress(%)	PPL(%)	Progress(%)	PPL(%)	Progress(%)	PPL(%)
Frontier-3D [23]	6.33	3.18	4.95	2.52	3.33	2.09	5.02	2.66
FPV+LVLM [16]	17.09	7.19	11.88	5.57	8.33	3.32	12.91	5.54
VLFM-3D [6]	35.34	19.79	27.59	15.48	24.45	10.78	29.75	15.79
RAVEN	**54.19**	**37.28**	**42.08**	**25.04**	**52.78**	**31.55**	**50.14**	**31.89**

TABLE II: Ablation results of **RAVEN**. Progress and PPL are reported for each task type.

	Task Type I (40 tasks)		Task Type II (30 tasks)		Task Type III (30 tasks)		All Tasks (100 tasks)	
	Progress(%)	PPL(%)	Progress(%)	PPL(%)	Progress(%)	PPL(%)	Progress(%)	PPL(%)
Semantic Voxels (No Rays, LVLM)	19.26	11.92	12.52	8.21	12.22	7.64	14.63	9.52
Semantic Rays (No Voxels, LVLM)	39.75	28.76	30.77	17.13	41.67	20.52	34.47	22.81
Semantic Voxels + Rays (No LVLM)	50.05	34.55	38.87	22.99	**53.89**	30.83	47.85	29.96
RAVEN (Sem Voxels + Rays + LVLM)	**54.19**	**37.28**	**42.08**	**25.04**	52.78	**31.55**	**50.14**	**31.89**

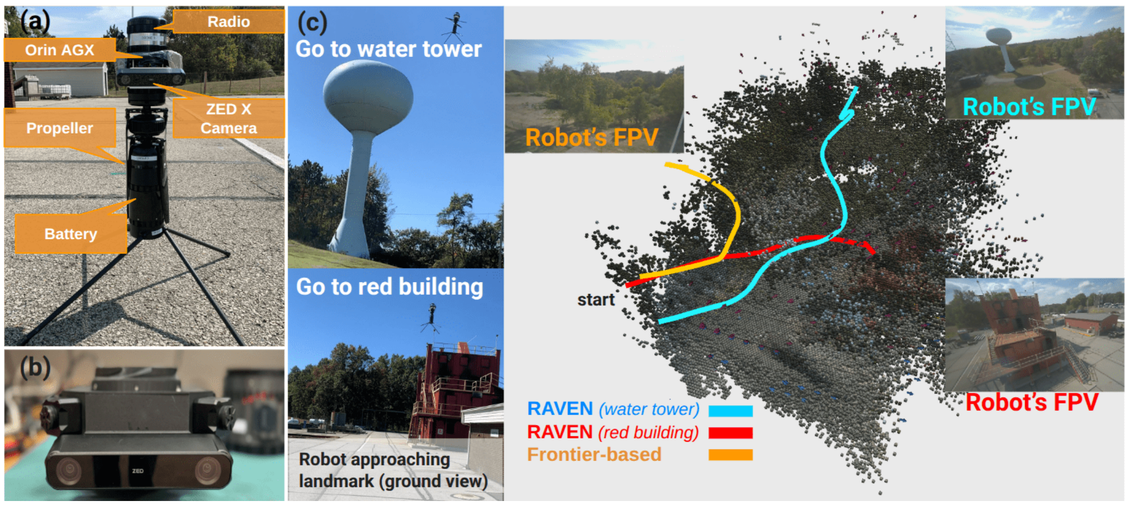

Fig. 7: (a) Robot hardware design (b) Front view of compute and camera payloads (c) **RAVEN**'s real-robot navigation trajectories toward the water tower (light blue), red building (red), and the frontier-based exploration trajectory (yellow), and their first-person, ground view images.

types, followed by the map-based VLFM-3D, then the map-free FPV+LVLM, while the non-semantic frontier-based exploration performs worst, as shown in Table I. In particular, **RAVEN** delivers 68.5% relative improvement in Progress and 102% relative improvement in PPL over VLFM-3D.

Figure 6(a) illustrates single-class navigation to fuel tanks from the same start position with three methods. The map-free FPV+LVLM relies only on recent FPV frames and LVLM prompts; without a consistent map it produces a reactive, meandering path, resulting in low efficiency. VLFM-3D benefits from a value map obtained by projecting similarity scores onto frontiers, which allows eventual goal finding; however, its motion remains myopic, repeatedly chasing the momentary maximum in the value map and yielding an unstable trajectory. In contrast, **RAVEN** activates semantic rays as soon as the fuel tank is briefly captured in an RGB image and travels nearly straight to the goal, producing a highly efficient path.

Figure 6(b) showcases a multi-class task and the role of LVLM. The robot is tasked to find cafe tables and bankomats, neither initially encoded in voxels or rays. Invoking LVLM search suggests additionally seeking sidewalk and building to find the bankomats; incorporating these cues into ray-search quickly guides the robot to the bankomats located on the sidewalk near a building. The robot then reorients and discovers the cafe tables through voxel-ray search.

F. Ablation Studies

We perform ablation studies on each component of **RAVEN**, with results in Table II. For Task Types I and II, the full model consistently outperforms all ablated variants, confirming the benefit of every component. For Task Type III, the semantic voxel+ray configuration achieves slightly higher Progress (within the margin of error), indicating that LVLM is less critical when sequential goal changes mainly test memory from voxels and rays. In contrast, LVLM auxiliary cues clearly aid in Task Types I and II.

Overall, semantic rays contribute more strongly than voxels by providing long-range guidance when objects are far away in large scenes, while voxels alone struggle to reason over long distances. Nevertheless, voxel search remains essential for precise localization near a target. Moreover, false positives in ray search can be particularly detrimental, as they may mislead the robot over long distances. These results indicate that voxels and rays are complementary, and both are necessary.

V. REAL-ROBOT EXPERIMENTS

A. Hardware Design

We use an aerial robot based on the Ascent Aerosystems Spirit platform, equipped with custom payloads. These include an NVIDIA Jetson AGX Orin for onboard computation and a ZED X camera providing RGB and depth for online mapping. Due to unreliable onboard pose estimation, we use GPS odometry. The robot components are shown in Figure 7(a), and the Orin AGX and ZED X payloads is shown in Figure 7(b).

B. Experiments and Results

We conducted real-world experiments using onboard computing with semantic voxel-ray online mapping and **RAVEN**, while leaving onboard LVLM integration for future work. Field tests took place at a firefighter training site. **RAVEN** was evaluated on two tasks—finding a water tower (beyond depth perception range) and finding a red building—and, for reference, a frontier-based exploration run was also conducted without a specific target. Since no oracle path exists in the real environment, the PPL metric is not applicable; instead, we report qualitative outcomes, trajectory lengths, average flight speed, and real-time mapping rates. The frontier-based method achieved a total flight distance of 40.46m with an average speed of 1.29m/s, but wandered without purposeful heading. Using **RAVEN**, the robot incrementally built a semantic voxel-ray memory and navigated toward the water tower and red building, as shown in Figure 7(c). The trajectory lengths for the two tasks were 74.97m and 42.72m, respectively, with an average flight speed of 1.21m/s. The onboard mapping module operated in real time at an average rate of 4.8Hz.

VI. CONCLUSION

We present **RAVEN**, a resilient behavior tree framework for aerial semantic navigation in outdoor environments. It uses an open-set semantic voxel-ray memory to enable strategic, long-horizon planning. **RAVEN** combines semantic voxel search for precise nearby localization, semantic ray search for long-range reasoning, and LVLM-guided search to mitigate sparse targets. We validate **RAVEN** through extensive simulation experiments and demonstrate its real-world applicability with initial aerial robot deployments.

REFERENCES

[1] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.

[2] A. Bircher *et al.*, "Receding horizon" next-best-view" planner for 3d exploration," in *ICRA*, 2016.

[3] D. Chaplot *et al.*, "Object goal navigation using goal-oriented semantic exploration," *NeurIPS*, 2020.

[4] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *CVPR*, 2022.

[5] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, *et al.*, "Goat: Go to any thing," in *Robotics: Science and Systems*, 2024.

[6] N. Yokoyama *et al.*, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *ICRA*, 2024.

[7] V. S. Dorbala, G. Sigurdsson, R. Piramuthu, J. Thomason, and G. S. Sukhatme, "Clip-nav: Using clip for zero-shot vision-and-language navigation," *arXiv preprint arXiv:2211.16649*, 2022.

[8] 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 *CVPR*, 2023.

[9] S. Liu, H. Zhang, Y. Qi, P. Wang, Y. Zhang, and Q. Wu, "Aerialvln: Vision-and-language navigation for uavs," in *ICCV*, 2023.

[10] Y. Gao, C. Li, Z. You, J. Liu, Z. Li, P. Chen, Q. Chen, Z. Tang, L. Wang, P. Yang, *et al.*, "Openfly: A comprehensive platform for aerial vision-language navigation," *arXiv preprint arXiv:2502.18041*, 2025.

[11] J. Xiao, Y. Sun, Y. Shao, B. Gan, R. Liu, Y. Wu, W. Gua, and X. Deng, "Uav-on: A benchmark for open-world object goal navigation with aerial agents," *arXiv preprint arXiv:2508.00288*, 2025.

[12] Q. Xie, S. Y. Min, P. Ji, Y. Yang, 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.

[13] J. Strader *et al.*, "Language-grounded hierarchical planning and execution with multi-robot 3d scene graphs," *arXiv preprint arXiv:2506.07454*, 2025.

[14] 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.

[15] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[16] J. Zhu *et al.*, "Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models," *arXiv preprint arXiv:2504.10479*, 2025.

[17] G. Zhou, Y. Hong, and Q. Wu, "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," in *AAAI Conference on Artificial Intelligence*, 2024.

[18] A. Z. Ren, J. Clark, A. Dixit, M. Itkina, A. Majumdar, and D. Sadigh, "Explore until confident: Efficient exploration for embodied question answering," *arXiv preprint arXiv:2403.15941*, 2024.

[19] P. Anderson *et al.*, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018.

[20] A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang, "Navila: Legged robot vision-language-action model for navigation," *arXiv preprint arXiv:2412.04453*, 2024.

[21] S. Wani *et al.*, "Multion: Benchmarking semantic map memory using multi-object navigation," *NeurIPS*, 2020.

[22] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*, 1998, pp. 47–53.

[23] 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.

[24] K. Weerakoon *et al.*, "Behav: Behavioral rule guided autonomy using vlms for robot navigation in outdoor scenes," in *ICRA*, 2025.

[25] B. Chen *et al.*, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," in *CVPR*, 2024.

[26] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *Robotics: Science and Systems (RSS)*, 2023.

[27] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *ICRA*, 2024.

[28] A. Werby *et al.*, "Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation," in *RSS*, 2024.

[29] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," in *Conference on Robot Learning*, 2023.

[30] Y. Deng, J. Wang, J. Zhao, X. Tian, G. Chen, Y. Yang, and Y. Yue, "Opengraph: Open-vocabulary hierarchical 3d graph representation in large-scale outdoor environments," *IEEE RA-L*, 2024.

[31] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *CVPR*, 2023.

[32] G. Dubey *et al.*, "Droan—disparity-space representation for obstacle avoidance," in *IROS*, 2017.