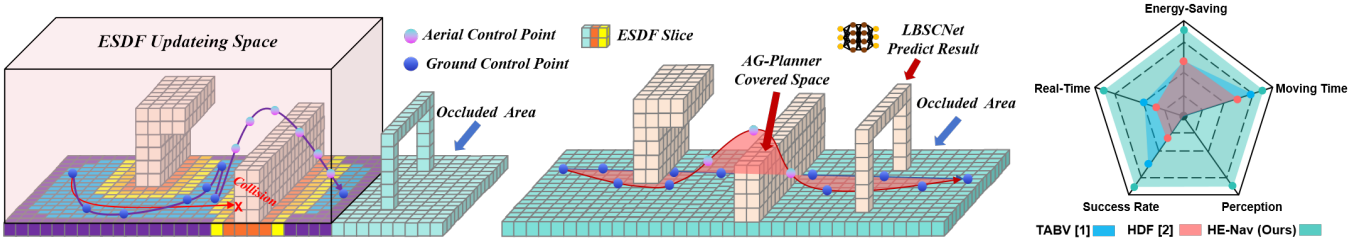
HE-Nav: A High-Performance and Efficient Navigation System for Aerial-Ground Robots

Anonymous Review. Paper-ID [17]



(a) ESDF-based Planning Framework: TABV [1]. (b) Our Navigation System HE-Nav for AGRs. (c) Comparison with Two Baselines.

Fig. 1: **(a)** ESDF-based methods face the dual predicaments of reduced path planning performance and increased energy consumption. **(b)** Our HE-Nav system can generate energy-saving, collision-free aerial-ground paths in real-time with the help of the LBSCNet model and AG-Planner. **(c)** HE-Nav Comparison with two baselines.

***Abstract*—Aerial-ground robots (AGRs) have unique dual-mode capabilities (i.e., flying and driving), making them ideal for search and rescue tasks. Existing AGR navigation systems have advanced in structured indoor scenarios using Euclidean Signed Distance Field (ESDF) maps for collision-free pathfinding. However, these systems are exhibit suboptimal performance and efficient in complex, occluded environments (e.g., forests) due to perception module and path planner limitations.**

In this paper, we present HE-Nav, the first high-performance, efficient and ESDF-free navigation system tailored for AGRs. The perception module utilizes a lightweight semantic scene completion network (LBSCNet), guided by a bird's eye view (BEV) feature fusion and enhanced by an exquisitely designed SCB-Fusion module and attention mechanism. This enables obstacle prediction in occluded areas, generating a complete local map. Building upon this map, our novel AG-Planner employs the energy-efficient Kinodynamic A* search algorithm and a gradient-based estimation method to guarantee planning is ESDF-free and energy-saving. Subsequent trajectory optimization and post-refinement processes yield safe, smooth, and dynamically feasible aerial-ground hybrid paths.

Extensive simulations and real-world experiments demonstrate HE-Nav's superiority over two recent AGR navigation systems, achieving 24.98% and 25.03% reductions in energy consumption while maintaining planning success rates of 98% and 97% in respective simulation scenarios. The code and hardware configuration will be made available.

I. INTRODUCTION

In recent years, aerial-ground robots (AGRs) [1, 2, 3, 4] have emerged as a promising solution for search [5, 6], exploration [7, 8], and rescue tasks [9, 10]. This is attributed to their exceptional mobility and long endurance, which enable them to seamlessly switch between aerial and ground modes, allowing for hybrid locomotion (i.e., flying and driving) in the above challenging tasks. Specifically, the ***perception module*** and the ***path planner*** are two crucial components in AGR navigation system that work synergistically, with the former generating a local map as the foundation for the latter to search for aerial-ground hybrid trajectories, ensuring *high-performance* (i.e., high planning success rate and shorter moving times) and *efficiency* (i.e., real-time planning and lower energy consumption).

Existing AGRs navigation system [1, 2, 4] utilize sensors (e.g., cameras) to perceive surrounding environments and establish Euclidean Signed Distance Field (ESDF) maps (in Fig. 1a), subsequently the path planner to search for collision-free trajectories that favour ground paths and only switch to the aerial mode when necessary (e.g., encountering impassable obstacles), thereby promoting energy efficiency.

Unfortunately, While these ESDF-based navigation systems have proven successful in structured indoor scenarios, they face two limitations when navigating in complex and occluded environments (e.g., disaster zones or wilderness areas).

Firstly, the *perception module* results in incomplete local maps (i.e., containing occlusion-induced unknown areas) since the narrow field of view in sensor-based mapping. This not only generates paths with high collision risk (e.g., *red path* in Fig.1a.) but also prolongs moving time since redundant paths (e.g., *purple path* in Fig.1a). To solve the above problem and generate complete local maps for navigation, the emerging semantic scene completion (SSC) network [11, 12, 13] holds promise, as it accurately predicts obstacle distribution and semantics in occluded areas. However, existing networks face a trade-off between completion accuracy and fast inference. Some use 3D convolution [14] to improve accuracy but unsuitable for resource-limited AGR devices (e.g., Jetson Xvaier NX) to ensure fast inference. Others propose lightweight net-

work structures [15] but with significantly reduced accuracy.

Secondly, the existing AGRs *path planners* are inefficient. Specifically, building the ESDF map generates redundant calculation times that do not meet the real-time requirements (i.e., planning time $<$ 1 ms) of path planning since it takes up about 70% of the time [16], and obstacles only take up 30% of the entire space (in Fig. 1a). Moreover, while the energy cost of flying is considered, the energy implications of ground movement (e.g., steering) are often overlooked, leading to overall energy inefficiency. Notably, the path planner's inefficiency stems from the above intrinsic shortcomings and the perception module's limitations in providing a local map.

To tackle these above limitations, we present ***HE-Nav***, the first *high-performance*, *efficient* and *ESDF-free* navigation system tailored for AGRs, as illustrated in Fig. 2. Initially, we developed the lightweight LBSCNet as a perception module to predict obstacle distribution in occluded areas. By processing sparse point clouds, it produces voxel occupancy and semantics, which are subsequently integrated into the local map for path planning.

During the planning phase, our AG-Planner generates an ignore obstacles initial trajectory (i.e., *blue path* in Fig.1b.) and employs the energy-efficient Kinodynamic A* algorithm to search for corresponding collision-free guide trajectory segments within obstacles. We then estimate the gradient between colliding and collision-free guide segments, wrapping the trajectory around obstacles while avoiding unnecessary ESDF computations. Lastly, a gradient-based spline optimizer and post-refinement process further refine the aerial-ground trajectory, yielding an energy-efficient, safe, smooth, and dynamically feasible path (i.e., *brown path* in Fig.1b.). The optimized trajectory is subsequently sent to the controller for precise tracking.

However, the design of the LBSCNet and AG-Planner confronts multiple challenges. First, balancing completion accuracy and high-speed inference remains a challenge for LBSCNet. Despite our LBSCNet employing sparse 3D convolutions [17] for a lightweight structure, it inadequately captures contextual information in occluded areas, reducing accuracy, while 3D feature fusion raises inference latency, affecting path planning performance.

To address these issues, we separate semantic and geometric learning processes into distinct branches, effectively leveraging their complementarity. Next, we incorporate the Criss-Cross Attention (CCA) [18] mechanism within the completion branch, enabling the sparse 3D convolutions can capture long-range dependencies and contextual information. Finally, we introduce the SCB-Fusion component, which facilitates the merging of BEV, semantic, and geometric features in the BEV space, ultimately reducing computational complexity and enhancing accuracy. (§ III)

Secondly, while *Zhou et al.* [16] devised an ESDF-free path planner for quadcopters, it fails to address AGR-specific requirements, particularly energy efficiency and dynamic constraints. Their flight-centric trajectory generation results in elevated energy consumption and the inherent non-holonomic constraints of AGRs make it impossible to naively migrate and use such planners.

To overcome these challenges, our AG-Planner employs a novel energy-efficient Kinodynamic A* algorithm, which adds additional energy costs to motion primitives involving sharp ground turns or aerial destinations, thereby promoting energy efficiency. Concurrently, we account for AGRs' non-holonomic constraints by limiting ground control point curvature. We then utilize an obstacle distance estimation method from [16] to circumvent obstacles, avoiding ESDF computations. To the best of our knowledge, AG-Planner is the first ESDF-free and energy-efficient planner tailored for AGRs. (§ IV)

We first assessed LBSCNet on the SemanticKITTI benchmark, comparing its accuracy and speed to a leading SSC network. Then, we tested HE-Nav in simulated and real environments, contrasting it with two AGR navigation baselines, showcasing its superior performance and efficiency (Fig.1c). Our evaluation reveals:

- **HE-Nav is high-performance.** HE-Nav achieved success rates of 98% and 97% in the two simulation scenarios, respectively, while having the shortest average movement time. (§ V-C)
- **HE-Nav is real-time planning.** AG-Planner achieves an **8X** reduction in total planning time compared to ESDF-based methods. (§ V-D)
- **HE-Nav is energy efficient.** AG-Planner significantly cuts energy consumption by 24.98% and 25.03% in two simulated settings, and by 10.34% in real-world outdoor situations. (§ V-C and § V-D)
- **LBSCNet is accurate and high-speed inference.** LBSCNet achieves state-of-the-art performance (IoU = 59.71) on the SemanticKITTI benchmark and enables high-speed inference (20.08 FPS). (§ V-B)

Our main contributions encompass the creation of the lightweight LBSCNet and energy-efficient AG-Planner. LBSCNet, with its novel structure and components (e.g., SCB-Fusion module), facilitates high-speed inference and complete local map generation. AG-Planner, based on this foundation, achieves ESDF-free planning and reduced planning time. By imposing costs on ground control points and employing the energy-saving Kinodynamic A* algorithm, our HE-Nav produces energy-efficient, safe, smooth, and dynamically feasible hybrid trajectories.

II. RELATED WORK

A. Motion Planning for AGRs

Numerous researchers have explored various aerial-ground robot configurations, such as incorporating passive wheels [1, 19, 3, 8, 4], cylindrical cages [20], or multi-limb [21] onto drones. In contrast, others [9, 6, 5, 10, 22] have integrated rotors with wheeled robots to achieve dual-mode (i.e., flying and driving) locomotion. These designs facilitate enhanced stability and control in both locomotion modes. Consequently, we also adopted this mechanical structure to customize further

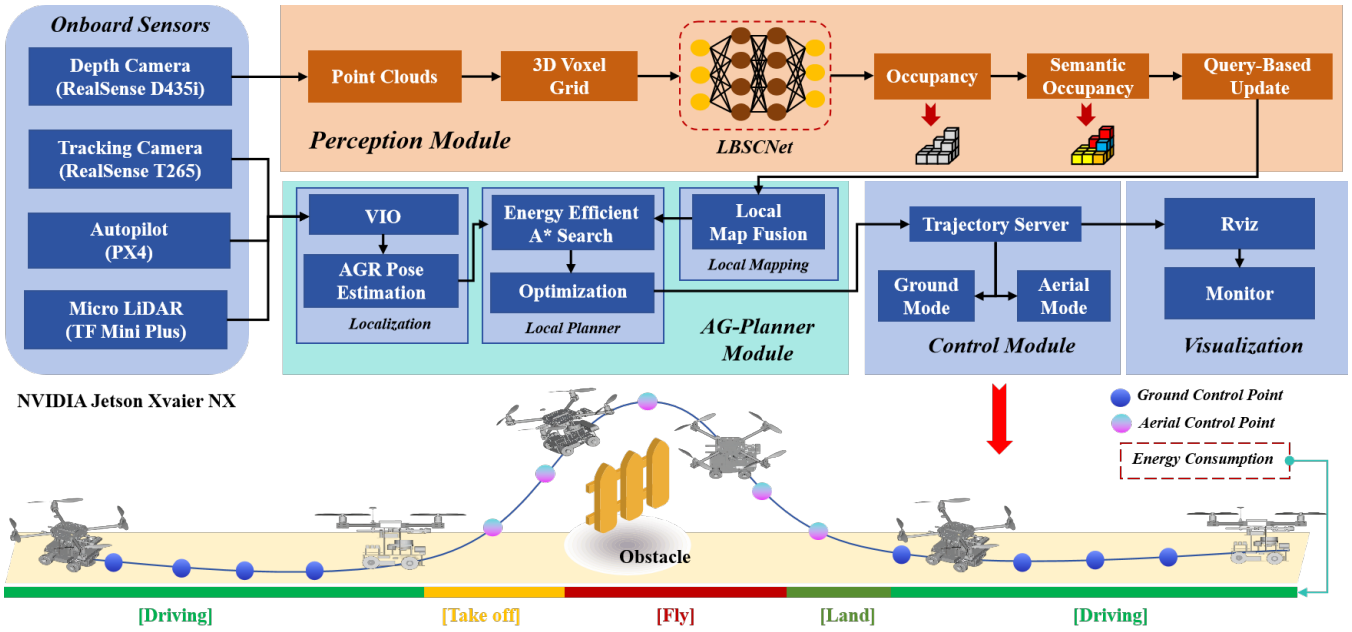


Fig. 2: HE-Nav system architecture. The perception, planning and control modules are deployed on the onboard computer (i.e., NVIDIA Jetson Xavier NX) and run asynchronously.

our AGR, which has four wheels and four rotors. Moreover, Existing research primarily focuses on innovative mechanical structure designs, and the area of AGR autonomous navigation remains underexplored. Recently, *Fan et al.* [2] address ground-aerial motion planning. Their approach initially employs the A* algorithm to search for a geometric path as guidance, favouring ground paths by adding extra energy costs to aerial paths. However, this path-searching method is limited due to its lack of dynamic models. Additionally, the local planner's trajectories lack post-refinement, resulting in potential issues with smoothness and dynamic feasibility. *Zhang et al.* [1] proposed a path planner and controller capable of efficient and adaptive path searching, but it relies on an ESDF map. The intensive computation and limited perception of occluded areas lead to a low success rate in path planning and increased energy consumption.

In this paper, our AG-Planner eliminates the need for building ESDF maps by estimating the gradient of collision trajectory segments, significantly reducing computational effort. It employs an energy-efficient Kinodynamic A* search algorithm to find guidance paths and utilizes a gradient-based spline optimizer to obtain the optimal trajectory while considering collision, smoothing, and dynamic feasibility. A post-refinement process further improves trajectory robustness. Additionally, we address non-holonomic constraints by incorporating curvature limit costs for ground trajectories in the optimization formula.

B. Occlusion-Aware for AGRs

AGR's sensor-based perception method cannot make the local map include the distribution of obstacles in the occluded area, which will cause the planned path to be sub-optimal. In recent years, the field of semantic scene completion [11, 14]has witnessed significant advancements, particularly in addressing the challenges posed by the obstruction of obstacles in complex and unknown environments. These advancements have led to the development of various point-cloud-based and camera-based methods. In the realm of camera-based methods, *Cao et al.* [12] introduced MonoScene, a groundbreaking approach that infers scene structure and semantics from a single monocular RGB image. Their key contribution lies in a novel 2D-3D feature projection bridging, inspired by optics, that enforces spatial semantic consistency. Another notable work by *Li et al.* [13] is VoxFormer, a Transformer-based semantic scene completion framework capable of generating complete 3D volume semantics using only 2D images. On the other hand, point-cloud-based methods have also made significant strides. *Cheng et al.* [23] proposed S3CNet, a method designed to predict semantically complete scenes from a single unified LiDAR point cloud. *Roldao et al.* [15] introduced LMSCNet, a multiscale 3D semantic scene completion approach that uses a 2D UNet backbone with comprehensive multiscale skip connections to enhance feature flow, along with a 3D segmentation head. *Xia et al.* [11] devised a method that employs knowledge distillation from a multi-frame model to improve the performance of single-frame semantic scene completion.

Despite substantial progress in camera and point-cloud-based SSC methods, their high computational demands limit their suitability for resource-constrained AGR platforms. Thus, we propose a lightweight SSC network using BEV feature fusion, serving as the perception module for the HE-Nav system, enabling rapid inference predictions and local map updates for path planning.

C. Energy-Efficient for AGRs

Energy efficiency is vital for aerial-ground robots since it directly impacts their endurance and overall performance. By utilizing energy efficiently, these robots can operate for extended periods, reducing downtime and optimizing flight and ground navigation. Although the path planning frameworks proposed by *Fan et al.* [2] and *Zhang et al.* [1] take into account the energy consumption of AGRs, their approach is not comprehensive enough. They primarily add additional penalties to aerial flight, encouraging the robot to favour ground paths. While this strategy somewhat reduces energy consumption, it overlooks other factors that contribute to energy inefficiency. For instance, when moving on the ground, the robot's turning angle can lead to additional energy consumption. Frequent steering demands extra energy from the motor, resulting in suboptimal energy usage.

Therefore, we propose an energy-efficient Kinodynamic A* path search algorithm that comprehensively considers ground steering energy consumption and aerial flight energy consumption to search for dynamically feasible aerial-ground hybrid trajectories.

III. Perception Module of HE-Nav

In this section, we introduce a lightweight three-branch SSC network (LBSCNet), depicted in Fig. 3. LBSCNet consists of a semantic branch, a completion branch, and a BEV fusion branch, serving as an alternative to conventional memory-intensive SSC networks that jointly predict geometry and semantics. By employing a pre-trained model offline on AGR devices, LBSCNet can infer and predict the obstacle distribution in occluded areas at high speed. Subsequently, these prediction results are updated into a local map, which is utilized for path planning.

A. LBSCNet Network Structure

LBSCNet decoupling the learning process of semantics and completion (or geometry), allows the network to concentrate on specific features (i.e., semantics and geometry), resulting in more efficient and fast learning. The specific structures are as follows:

***Semantic Branch*:** This branch consists of a voxelization layer and three encoder blocks sharing a similar architecture, each encoder block comprises a residual block [24] with sparse 3D convolutions and a cross-scale global attention (CSGA) module from [25]. The integration of the CSGA module not only aligns multi-scale features with global voxel-encoded attention to capturing the long-range relationship of context but also alleviates the computational burden by reducing feature resolution.

Specifically, in the voxelization layer, point clouds $P \in \mathbb{R}^{N\times 3}$ are partitioned based on the voxel resolution s and mapped into voxel space. Subsequently, an aggregation function (i.e., max function) is applied to the point cloud within each voxel, yielding a single feature vector. A multi-layer perceptron (MLP) reduces the dimensionality of this feature vector, producing the final voxel features V_{f_m} with a spatial

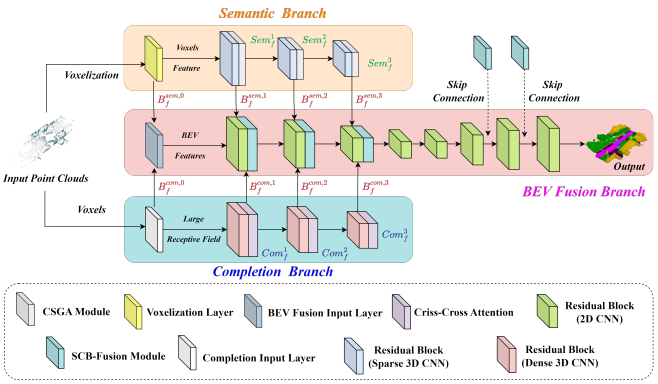


Fig. 3: The overview of the proposed LBSCNet. It consists of semantic, completion and BEV fusion branches.

resolution of $L \times W \times H$, f_m represents the index of the voxel. The voxel features V_{f_m} are then input into three encoder blocks to obtain semantic features $\{Sem_f^1, Sem_f^2, Sem_f^3\}$ (Fig. 3). The semantic branch is optimized using lovasz loss [26] and cross-entropy loss [27]. The semantic loss L_{sem} is the sum of the loss at each stage, expressed as follows:

$$L_{sem} = \sum_{i=1}^{3}(L_{cross,i} + L_{lovasz,i}) \tag{1}$$

***Completion Branch*:** The input to the completion branch is voxels $V \in \mathbb{R}^{1\times L\times W\times H}$ generated by point clouds. The output is the multi-scale dense completion features $\{Com_f^1, Com_f^2, Com_f^3\}$, providing more intricate geometric information.

As depicted in Fig. 3, the completion branch comprises an input layer (kernel size $7 \times 7 \times 7$), three residual blocks and three GPU memory-efficient criss-cross attention (CCA) [18] modules. The residual blocks incorporate dense 3D convolutions with a kernel size of $3 \times 3 \times 3$, capturing local geometric features. Conversely, the criss-cross attention (CCA) [18] module is designed to capture long-range dependencies by gathering contextual information in both horizontal and vertical directions, thereby enriching the completion features with a global context. The training loss L_{com} for this branch is calculated as follows:

$$L_{com} = \sum_{i=1}^{3}(L_{binary\_cross,i} + L_{lovasz,i}) \tag{2}$$

where i denotes the $i-th$ stage of the completion branch and L_{binary_cross} indicates the binary cross-entropy loss. Notably, during training, both the semantic and completion branches undergo deep supervision [28]. Lightweight MLPs are attached as auxiliary heads [25] after each encoder block to obtain semantic and geometric predictions for valid voxels. However, during inference, these auxiliary heads are removed to maintain a lightweight network structure.

***BEV Feature Fusion Branch*:** Previous research on SSC tasks has relied on fusing dense 3D features, resulting in considerable computational overhead and hindering deployment on

resource-constrained AGR devices. We propose a lightweight BEV fusion branch specifically designed for SSC tasks, capitalizing on recent advancements in BEV perception [29, 30, 31]. By projecting learned semantic and geometric features into BEV space and incorporating the innovative SCB-Fusion module, we significantly reduce computational demands while maintaining rapid inference capabilities. Specifically, our BEV fusion network employs a U-Net architecture with 2D convolutions, featuring an input layer and four residual blocks in the encoder (Fig. 3). The process of projecting semantic and geometric features to BEV space is as follows:

***Semantic Feature Projection*:** To project three-dimensional semantic features $\{Sem_f^1, Sem_f^2, Sem_f^3\}$ into the two-dimensional BEV space, we first generate a BEV index based on the voxel index f_m and then the features sharing identical BEV indices are aggregated using an aggregation function (e.g., the max function) to yield sparse BEV features. Utilizing the feature densification function offered by spconv [32], we generate dense BEV features $\{B_f^{sem,0}, B_f^{sem,1}, B_f^{sem,2}, B_f^{sem,3}\}$ based on the BEV index and sparse BEV features.

***Geometric Feature Projection*:** For geometric features $\{Com_f^1, Com_f^2, Com_f^3\}$, we stack dense 3D features along the z-axis and apply 2D convolution to reduce the feature dimension, generating dense BEV features $\{B_f^{com,0}, B_f^{com,1}, B_f^{com,2}, B_f^{com,3}\}$. Subsequently, the projected features are input into the BEV fusion network (Fig. 3). The BEV loss L_{bev} is :

$$L_{bev} = L_{cross} + L_{lovasz} \tag{3}$$

***Feature Fusion after Projection*:** To fuse the projected features, we devise an SCB-Fusion module (Fig. 13a) that fuses current semantic features, geometric features, and BEV features from the previous layer. Specifically, we first compute channel attention for features $B_{pre}/B_{com}/B_{sem}$ to adaptively weight the feature channels. The weighted features are then summed and passed through a 1×1 convolution and CCA attention to obtain the fused features F_{SCB}. The fused features can be expressed as:

$$\begin{aligned} F_{SCB} = \Phi \{ & \lambda [N(B_{pre})] \times B_{pre} \\ & + \lambda [N(B_{com})] \times B_{com} \\ & + \lambda [N(B_{sem})] \times B_{sem} \end{aligned} \tag{4}$$

where λ denotes the sigmod function. Φ is the 1×1 convolution. The B_{pre} represents features from the previous stage.

***LBSCNet Total Loss Function*:** We train the whole network end-to-end. The multi-task loss L_{total} is expressed as :

$$L_{total} = 3 \times L_{bev} + L_{sem} + L_{com} \tag{5}$$

where L_{bev}, L_{sem} and L_{com} respectively represent BEV loss, the semantic loss and completion loss.

IV. Aerial-Ground Motion Planning

In this section, we introduce the novel AG-Planner. It is built on EGO-Planner [16] and consists of **1)** an energy-efficient Kinodynamic A* path searching front-end, **2)** a gradient-based trajectory optimization back-end and **3)** a post-refinement procedure. Our AG-Planner evaluates and projects gradient information directly from obstacles instead of a pre-built ESDF like [1]. To the best of our knowledge, AG-Planner is the first ESDF-free and energy-efficient planner tailored for AGRs.

Algorithm 1: Energy-Efficient Kinodynamic A* Search

Input: Start State x_s and Target State x_g
Output: Energy-Efficient Valid Path between x_s and x_g
Data: O=$\emptyset$ and C= $\emptyset$; $f(x_s) = g(x_s) + h(x_s)$; $O.push(x_s)$

while *!O.empty()* **do**
  $x \longleftarrow O.popMin()$
  if $x == x_g$ **then**
    return *path*
  end
  else
    $C.push(x)$
    foreach $n \in neig(x)$ **do**
      $g_n \longleftarrow (um.squaredNorm() + w_{time}) * \tau + g(x)$
      // next node flying
      if $z \geq$ *ground_judge* **then**
        $g_n -= x.fly\_penalty\_g$
        // add fly penalty cost
        $g_n += fly\_cost * z + f\_cost\_base$
        $fly\_penalty\_g = fly\_cost * z + f\_cost\_base$
        $steer\_penalty\_g = 0$
        $next\_motion\_state = true$
      end
      // next node driving
      else
        $g_n -= x.steer\_penalty\_g$
        // add steer penalty cost
        $steer\_cost = steer\_cost\_ * pow(\omega_z, 2)$
        $g_n += steer\_cost + ground\_cost\_base$
        $steer\_penalty\_g = steer\_cost + g\_cost\_base$
        $fly\_penalty\_g = 0$
        $next\_motion\_state = false$
      end
      $f_n = g_n + \lambda * estimateHeuristic(n, x_g)$
      if $n \notin O \cup C$ **then**
        $n.updateCost(g_n, fly\_penalty, steer\_penalty, f_n)$
        $O.push(n)$
      end
    end
  end
end
return *null* // Cannot find a valid path

A. Energy-Efficient Kinodynamic Hybrid A* Path Searching

Our AG-Planner first creates a naive "*initial trajectory*" ι (in Fig. 4a) that overlooks obstacles by randomly adding coordinate points, considering the positions of both the starting and target points. Following that, for the "*collision trajectory segment*" (i.e., the trajectory inside the obstacle), the back end of our planner based on [33] to propose an energy-efficient kinodynamic A* path search algorithm (in Alg. 1) to establish a safe "*guidance trajectory segment*" τ, which uses motion primitives instead of straight lines as graph edges in the searching loop. In this algorithm, we add extra flying and

ground-steering energy consumption for the motion primitives (in Fig. 4a). Consequently, the path searching not only tends to plan ground trajectories and avoid large turns but also switches to aerial mode and flies over them only when AGRs encounter huge obstacles, thereby promoting energy-saving.

B. Gradient-Based B-spline Trajectory Optimization

***B-spline Trajectory Formulation*:** In trajectory optimization (in Fig. 4b), the trajectory is parameterized by a uniform B-spline curve Θ, which is uniquely determined by its degree p_b, N_c control points $\{Q_1, Q_2, Q_3, ..., Q_{N_c}\}$, and a knot vector $\{t_1, t_2, t_3, ..., t_{M-1}, t_M\}$, where $Q_i \in \mathbb{R}^3, t_m \in \mathbb{R}, M = N + p_b$. Following the matrix representation of the [34] the value of a B-spline can be evaluated as:

$$\Theta(u) = [1, u, ..., u^p] \cdot M_{p_b+1} \cdot [Q_{i-p_b}, Q_{i-p_b+1}, ..., Q_i]^T \quad (6)$$

where M_{p_b+1} is a constant matrix depends only on p_b. And $u = (t - t_i)/(t_{i+1} - t_i)$, for $t \in [t_i, t_{i+1})$.

In particular, in ground mode, we assume that AGR is driving on flat ground so that the vertical motion can be omitted and we only need to consider the control points in the two-dimensional horizontal plane, denoted as $Q_{ground} = \{Q_{t0}, Q_{t1}, ..., Q_{tM}\}$, where $Q_{ti} = (x_{ti}, y_{ti}), i \in [0, M]$. In aerial mode, the control points are denoted as Q_{aerial}. According to the properties of B-spline: the k^{th} derivative of a B-spline is still a B-spline with order $p_{b,k} = p_b - k$, since $\triangle t$ is identical alone Θ, the control points of the velocity V_i, acceleration A_i and jerk J_i curves are obtained by:

$$V_i = \frac{Q_{i+1} - Q_i}{\triangle t}, A_i = \frac{V_{i+1} - V_i}{\triangle t}, J_i = \frac{A_{i+1} - A_i}{\triangle t} \quad (7)$$

***Collision Avoidance Force Estimation*:** Inspired by [16], for each control point on the collision trajectory segment, vector v (i.e., a safe direction pointing from inside to outside of that obstacle) is generated from ι to τ and p is defined at the obstacle surface (in Fig. 4a). With generated $\{p, v\}$ pairs, the planner maximizes D_{ij} and returns an optimized trajectory. The obstacle distance D_{ij} if i^{th} control point Q_i to j^{th} obstacle is defined as:

$$D_{ij} = (Q_i - p_{ij}) \times v_{ij} \quad (8)$$

Because the guide path τ is energy-saving, the generated path is also energy efficient (in Fig. 4a).

***B-spline Trajectory Optimization and Post-refinement Procedure*:** The basic requirements of the B-spline paths are threefold: *smoothness*, *safety*, and *dynamical feasibility*. Based on the special properties of AGR bimodal, we first adopt the following cost terms designed by *Zhou et al.* [16]:

$$\min J_1 = \lambda_s J_s + \lambda_c J_c + \lambda_f (J_v + J_a + J_j) \quad (9)$$

where J_s is the smoothness penalty, J_c is for collision, and J_v, J_a, J_j are dynamical feasibility costs that limit velocity, acceleration and jerk. $\lambda_s, \lambda_c, \lambda_f$ are weights for each cost terms. Detailed explanations can be found in [16]. Subsequently, based on our observations, AGR faces non-holonomic constraints when driving on the ground, which means that the ground velocity vector of AGR must be aligned with its yaw angle. Additionally, AGR needs to deal with curvature limitations that arise due to minimizing tracking errors during sharp turns. Therefore, a cost for curvature needs to be added, and J_n can be formulated as:

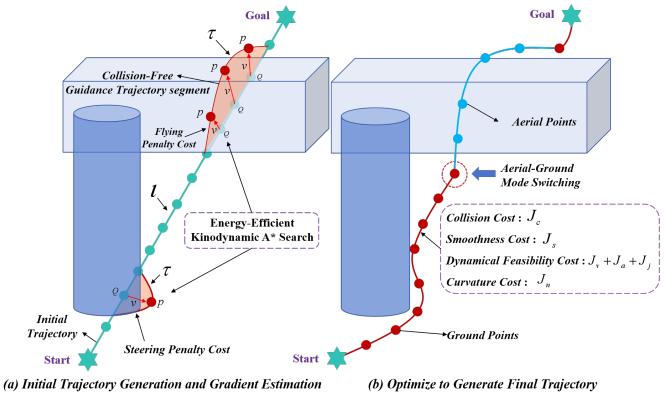


Fig. 4: Illustration of AG-Planner and topological trajectory generation.

$$J_n = \sum_{i=1}^{M-1} F_n(Q_{ti}) \quad (10)$$

where $F_n(Q_{ti})$ is a differentiable cost function with C_{max} specifying the curvature threshold:

$$F_n(Q_{ti}) = \begin{cases} (C_i - C_{max})^2, C_i > C_{max}, \\ 0, C_i \leq C_{max} \end{cases} \quad (11)$$

where $C_i = \frac{\triangle \beta_i}{\triangle Q_{ti}}$ is the curvature at Q_{ti}, and the $\triangle \beta_i = \left| \tan^{-1} \frac{\triangle y_{ti+1}}{\triangle x_{ti+1}} - \tan^{-1} \frac{\triangle y_{ti}}{\triangle x_{ti}} \right|$. In general, the overall objective function is formulated as follows:

$$\min J_{all} = \lambda_s J_s + \lambda_c J_c + \lambda_f (J_v + J_a + J_j) + \lambda_n J_n$$
$$s.t. \begin{cases} J_s = \sum_{i=1}^{N_c-1} \|A_i\|_2^2 + \sum_{i=1}^{N_c-2} \|J_i\|_2^2 \\ J_c = \sum_{i=1}^{N_c} j_c(Q_i) \\ J_v = \sum_{i=1}^{N_c} \omega_v F(V_i) \\ J_a = \sum_{i=1}^{N_c-1} \omega_a F(A_i) \\ J_j = \sum_{i=1}^{N_c-2} \omega_j F(J_i) \\ J_n = \sum_{i=1}^{M-1} F_n(Q_{ti}) \end{cases} \quad (12)$$

The optimization problem is solved using the non-linear optimization solver NLopt [35], with post-refinement from [16] for constraint violations. After path planning, a setpoint from the trajectory is selected and sent to the controller. Aerial setpoints include yaw angle and 3D position, velocity, and acceleration, while ground ones include yaw angle and 2D position and velocity. In addition, when the z-axis coordinate of the next control point is greater than the ground threshold, that is, when mode switching is required, an additional trigger signal will be sent to the controller (i.e., PX4 Autopilot). The controller will automatically switch to the flight state.

V. Evaluation

In this section, we first assess the LBSCNet-based perception module on the SemanticKITTI benchmark, examining its accuracy and rapid inference capabilities in SSC tasks. Subsequently, we integrate this module with the AG-Planner by deploying a pre-trained model offline, forming a comprehensive HE-Nav system. We then evaluate the AGR's autonomous navigation capability using HE-Nav in both simulated and real-world settings, focusing on ***performance*** metrics (i.e., planning success rate, average movement time) and ***efficiency*** aspects (i.e., average planning time, energy consumption).

A. Evaluation setup

Perception Module: For training and testing of LBSCNet, we utilized a server with 4 NVIDIA RTX 3090 GPUs and 128GB memory, employing the outdoor SemanticKITTI dataset [36]. We trained the model for 80 epochs on a single NVIDIA 3090 GPU with a batch size of 12, using the Adam optimizer [37] at an initial learning rate of 0.001, and augmenting the input point cloud by random flipping along the $x-y$ axis. Ultimately, we deployed the pre-trained model offline with the best completion accuracy to complete the local map.

Simulation Experiment: Experiments were executed on a laptop equipped with Ubuntu 20.04, an i9-13900HX CPU, and an NVIDIA RTX 4060 GPU to simulate aerial-ground robotic navigation within complex environments. The test scenarios comprised a $20m \times 20m \times 5m$ square room and a $3m \times 30m \times 5m$ corridor with numerous random obstacles, creating occluded spaces and unknown areas (Fig. 8A). The AGR's task was to navigate from a starting point to a designated destination without collision.

Indoor and Outdoor Real-world Experiment: We employed HE-Nav on a custom AGR platform (Fig. 5) for indoor and outdoor experiments, using Prometheus software [38] with a RealSense D435i depth camera, a T265 tracking camera, and a Jetson Xavier NX computer. Hardware details are in the supplementary materials. We assessed the average energy consumption per second for AGR during driving and flying (Table IV) to establish a basis for evaluating energy usage in real and simulated tests.

Metrics: For the LBSCNet, we use intersection over union (IoU) to evaluate scene completion quality and the mean IoU (mIoU) of 19 semantic classes to assess semantic segmentation performance. moreover, we also focus on LBSCNet's inference speed to ensure it meets the real-time requirements for autonomous navigation. Regarding planning, we pay attention to performance metrics such as planning success rate (%), total moving time (s), and efficient metrics planning time (ms) as well as energy consumption (J).

***Baseline methods*:** For the perception module, we compare LBSCNet against the state-of-the-art SSC methods: (1) Camera-based SSC method MonoScene [12] and VoxFormer[13], (2) Point-Cloud-based SSC methods including LMSCNet [15], and SSCNet [39] and SCPNet[11]. To evaluate the performance and efficiency of HE-Nav, we compared HE-Nav with two state-of-the-art AGRs navigation systems: TABV [1], HDF [2].

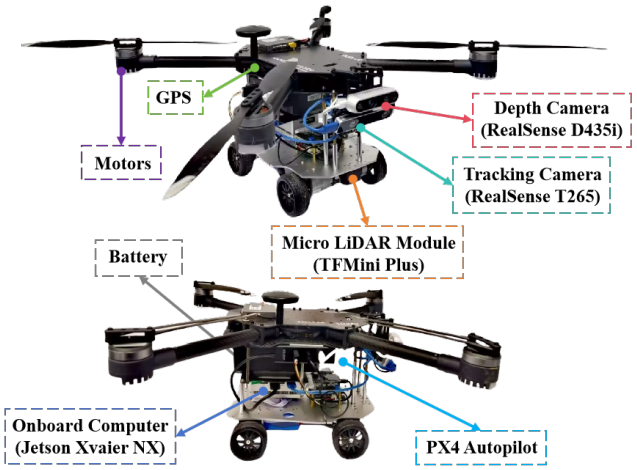


Fig. 5: The detailed composition of the robot platform.

Method	*IoU*	*mIoU*	*Prec.*	*Recall*	*FPS*
SSCNet [39]	53.20	14.55	59.13	**84.15**	12.00
LMSCNet [15]	55.32	17.01	77.11	66.19	13.50
LMSCNet-SS [15]	56.72	17.62	**81.55**	65.07	13.50
S3CNet [23]	45.60	29.50	48.79	77.13	1.20
Monoscene [12]	38.55	12.22	51.96	59.91	< 1
VoxFromer-T [13]	57.69	18.42	69.95	76.70	< 1
VoxFromer-S [13]	57.54	16.48	70.85	75.39	< 1
SCPNet [11]	56.10	**36.70**	72.43	78.61	< 1
LBSCNet (Ours)	**59.71**	23.58	77.60	71.29	**20.08**

TABLE I: Quantitative comparison against the state-of-the-art SSC methods on the official SemanticKITTI test benchmark.

B. LBSCNet Comparison against the state-of-the-art.

Quantitative Results: We evaluated our proposed LBSCNet against state-of-the-art SSC methods on the SemanticKITTI test datasets by submitting results to the official test server. Table I demonstrates that LBSCNet not only achieves the highest completion metric IoU (59.71%) but also ranks third in the semantic segmentation metric mIoU (23.58%). Although SCPNet's semantic segmentation accuracy surpasses ours, its dense network design renders it incapable of real-time operation (i.e., FPS < 1). In contrast, LBSCNet outperforms SCPNet by 6.43% in IoU and runs approximately ***20 times*** faster in a single RTX 3090 GPU.

The remarkable accuracy and rapid inference performance of our LBSCNet are primarily due to the innovative semantic and completion decoupling network structure, which exploits contextual semantic information to bolster scene understanding and completion. The integration of the novel SCB-fusion and CCA modules enables the network to remain lightweight while significantly enhancing completion accuracy by capturing

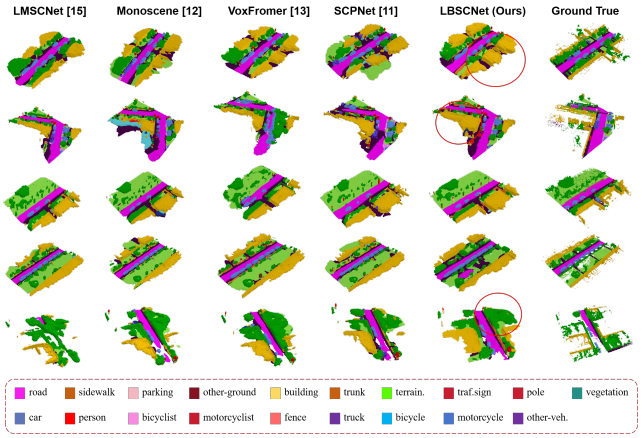


Fig. 6: Qualitative results of our method and others. LBSCNet better captures the scene layout in large-scale scenarios.

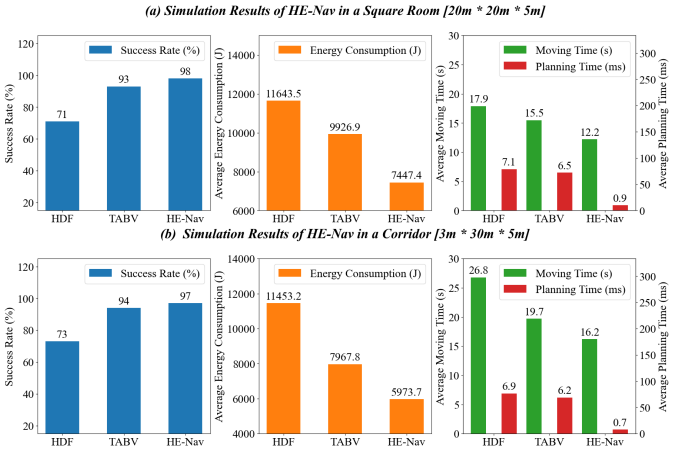


Fig. 7: Quantitative results of HE-Nav in two simulation scenarios.

contextual features and learning long-distance dependencies. Additionally, the employment of sparse 3D convolutions and lightweight BEV feature fusion ensures low latency and high-speed inference (20.08 FPS), making LBSCNet ideal for real-time perception in AGR navigation systems. Further details can be found in Table V and Fig 11.

Qualitative Results: We provide visualizations results on the SemanticKITTI validation set and include results from LMSCNet [15], Monoscene[12], VoxFormer[13], and SCPNet [11]. As illustrated in Fig. 6, our LBSCNet demonstrates superior SSC predictions, particularly for "wall" classes and larger objects like cars, aligning with the results in Table I. Importantly, the occlusion areas we target, such as vegetation and trees behind walls, are accurately completed, proving vital for subsequent path-planning applications. More qualitative and quantitative results are provided in the supplementary material, i.e., in Section VII-A.

Method	IoU ↑	mIoU ↑
LBSCNet (ours)	58.34	22.74
w/o SCB-Fusion Module	57.05	21.26
w/o Criss-Cross Attention	57.20	22.17

TABLE II: Ablation study of our model design choices on the SemanticKITTI validation set.

Ablation Study: Ablation studies conducted on the SemanticKITTI validation set (Table II) emphasize the significance of two crucial components in our network: CCA attention mechanisms and the SCB-Fusion Module. The CCA attention mechanism greatly influences completion accuracy by effectively aggregating context across rows and columns. The absence of CCA results in a 1.95% decrease in completion accuracy. On the other hand, the SCB-Fusion module captures local scene features, including occluded areas, with minimal computational overhead. Removing the SCB-Fusion module leads to a 2.21% reduction in IoU.

C. Simulated Air-Ground Robot Navigation

In a square room and corridor scenario (Fig 8B), we conducted a comparative analysis of our HE-Nav system, TABV [1], and HDF [2]. Through 100 trials with varied obstacle placements, we evaluated the average moving time, planning time (including updating the ESDF map and path planning for TABV), and success rate (i.e., collision-free) of each system (Fig. 7). Furthermore, we obtained average energy consumption results for the 100 simulated trials by combining recorded flight and driving times with real-world energy consumption data from our custom AGR (Table IV).

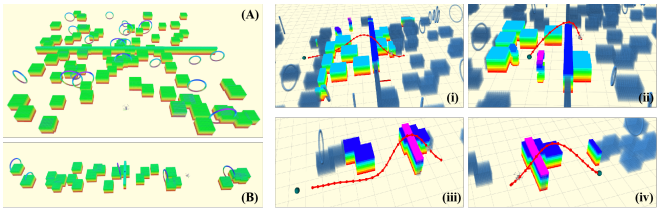
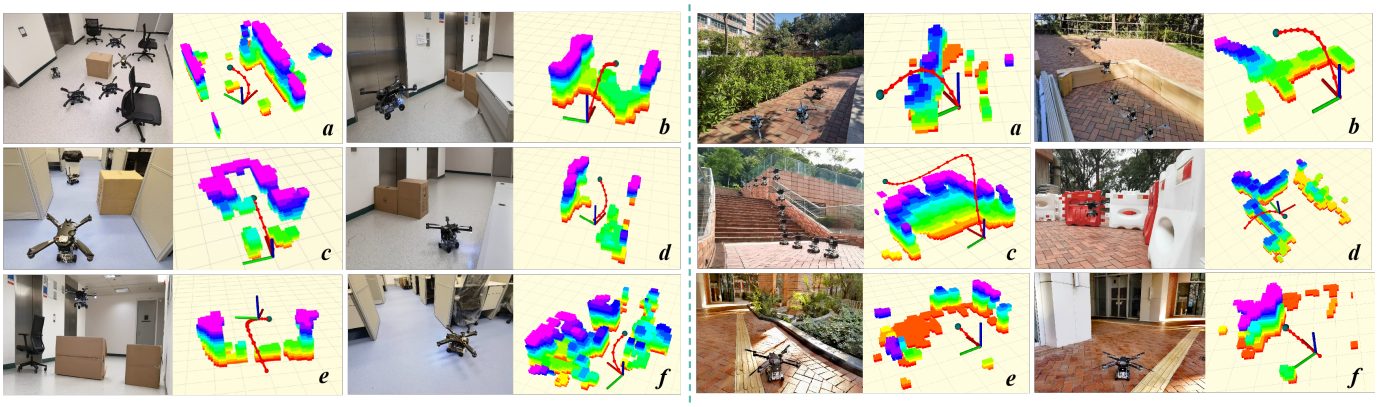


Fig. 8: Qualitative results of path planning and occlusion prediction in simulation environment.

The results shown in Fig. 7 highlight the exceptional performance of our HE-Nav system. It achieves high success rates of 98% and 97% in square rooms and corridors, respectively, with average movement times of 12.2s and 16.2s. The average planning time is significantly accelerated, being 6 times faster than TABV [1] and HDF [2], thanks to the elimination of redundant ESDF calculations. Additionally, our path planner seamlessly integrates with the *energy-efficient Kinodynamic A* algorithm*, resulting in the lowest average energy consumption of 7447.4 J and 5973.7 J. Comparatively, HE-Nav achieves a 24.98% energy reduction in square scenarios and a 25.03% reduction in corridors compared to TABV [1].

In contrast to TABV [1], which primarily focuses on flight energy consumption and lacks the ability to sense obstacle distribution in occluded areas beforehand, our HE-Nav system addresses this limitation effectively. By perceiving and



(a) ***Indoor Real-World Experiments.*** **(b)** ***Outdoor Real-World Experiments.***

Fig. 9: HE-Nav's visual results showcase its autonomous navigation capabilities in 6 indoor and 6 outdoor scenes. The system effectively predicts obstacle distribution in occluded areas and plans collision-free hybrid trajectories.

Methods	**Indoor Scenario**						**Outdoor Scenario**					
	a	b	c	d	e	f	a	b	c	d	e	f
TABV [1]	3217.4	10207.3	3971.8	5783.5	12362.9	3105.6	10323.6	10569.4	13117.2	12649.3	6480.5	6682.8
HE-Nav (Ours)	2891.7	9652.5	3614.1	5279.3	11874.5	2765.9	9662.9	9764.8	11283.2	11858.7	5579.3	5754.2
HE-Nav Energy Savings	10.12%	5.43%	9.01%	8.71%	3.95%	10.93%	6.41%	7.61%	13.95%	6.25%	13.89%	13.90%
Average Energy Savings	8.02%						10.34%					

TABLE III: Quantitative results of AGR energy consumption (J) in complex indoor and outdoor scenes.

predicting occlusions, our ESDF-free AG-Planner can bypass these regions and significantly reduce collision risks. This not only results in more optimal overall energy consumption but also greatly mitigates the risk of collision for the planned path.

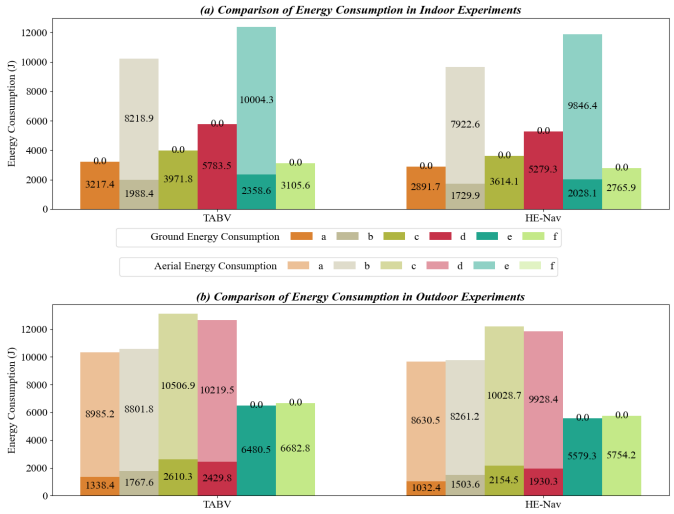


Fig. 10: Quantitative results of indoor and outdoor real environmental energy consumption.

D. Real-world Air-Ground Robot Navigation

We assess HE-Nav's performance and energy efficiency across 6 indoor and 6 outdoor scenarios (Fig. 9). In indoor settings, as illustrated in Tab. III and Fig. 10a, HE-Nav consistently demonstrates lower average energy consumption than TABV [1]. For example, in scenarios a, c, and f, our system achieves ground energy consumption reductions of 10.12%, 9.01%, and 10.93% compared to TABV, primarily attributable to the incorporation of additional turning penalty terms in the ground segment. This approach effectively minimizes energy usage in ground mode by reducing high-angle turning paths. Concurrently, LBSCNet swiftly predicts obstacle distribution in occluded areas, constructing a more complete local map (e.g., a, c, f visualization results) to serve as the foundation for AG-Planner's search path.

Transitioning to outdoor scenarios, HE-Nav surpasses TABV [1] with a 10.34% reduction in average energy consumption (Table III), mainly due to the optimization of smooth aerial paths, which minimizes flight energy consumption. Our system adeptly predicts obstacle distribution in cluttered and occluded environments (i.e., a, b, c), while fully accommodating AGR's non-holonomic constraints and energy costs. In ground mode, the planned path exhibits smoothness and dynamic feasibility (e.g., b, e, f visualization results). Crucially, scenes e and f display substantial reductions in ground energy consumption of 13.89% and 13.90%, respectively, owing to challenging terrain (e.g., the yellow blind road in e, f). In contrast to the TABV method, which neglects ground steering energy consumption, HE-Nav integrates ground steering constraints into the optimization process, thereby enhancing energy efficiency on difficult terrains.

Furthermore, the removal of ESDF significantly reduced the overall path planning time, achieving an approximate 8x improvement compared to the ESDF-based TABV (Fig. 15). The average planning time, including ESDF updating, was reduced to 0.95ms and 1.12ms on the Jetson Xavier NX platform. For additional qualitative and quantitative results, please refer to the supplementary material and video, specifically Section VII-C.

VI. Conclusion

we have presented HE-Nav, the first high-performance, efficient and ESDF-free navigation system specifically designed for aerial-ground robots (AGRs). By integrating innovative features such as the lightweight BEV-guided semantic scene completion network (LBSCNet) and the aerial-ground motion planner (AG-planner), our system is capable of predicting obstacle distributions in occluded areas and generating low-collision risk, energy-efficient aerial-ground hybrid trajectories in real-time ($\approx$ 1 ms). Through extensive simulations and real experiments, HE-Nav has been shown to significantly outperform recent planning frameworks in performance (i.e., planning success rate and total movement time) and efficiency (i.e., planning time and energy consumption).

VII. Acknowledgments

We thank all anonymous reviewers for their helpful comments.

References

[1] Ruibin Zhang, Yuze Wu, Lixian Zhang, Chao Xu, and Fei Gao. Autonomous and adaptive navigation for terrestrial-aerial bimodal vehicles. *IEEE Robotics and Automation Letters*, 7(2):3008–3015, 2022.

[2] David D Fan, Rohan Thakker, Tara Bartlett, Meriem Ben Miled, Leon Kim, Evangelos Theodorou, and Ali-akbar Agha-mohammadi. Autonomous hybrid ground/aerial mobility in unknown environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3070–3077. IEEE, 2019.

[3] Neng Pan, Jinqi Jiang, Ruibin Zhang, Chao Xu, and Fei Gao. Skywalker: A compact and agile air-ground omnidirectional vehicle. *IEEE Robotics and Automation Letters*, 8(5):2534–2541, 2023.

[4] Ruibin Zhang, Junxiao Lin, Yuze Wu, Yuman Gao, Chi Wang, Chao Xu, Yanjun Cao, and Fei Gao. Model-based planning and control for terrestrial-aerial bimodal vehicles with passive wheels. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1070–1077, 2023. doi: 10.1109/IROS55552.2023.10342188.

[5] Xinyu Zhang, Yuanhao Huang, Kangyao Huang, Xiaoyu Wang, Dafeng Jin, Huaping Liu, and Jun Li. A multi-modal deformable land-air robot for complex environments. *arXiv preprint arXiv:2210.16875*, 2022.

[6] Xinyu Zhang, Yuanhao Huang, Kangyao Huang, Ziqi Zhao, Jingwei Li, Huaping Liu, and Jun Li. Coupled modeling and fusion control for a multi-modal deformable land-air robot. *arXiv preprint arXiv:2211.04185*, 2022.

[7] Eric Sihite, Arash Kalantari, Reza Nemovi, Alireza Ramezani, and Morteza Gharib. Multi-modal mobility morphobot (m4) with appendage repurposing for locomotion plasticity enhancement. *Nature communications*, 14(1):3323, 2023.

[8] Youming Qin, Yihang Li, Xu Wei, and Fu Zhang. Hybrid aerial-ground locomotion with a single passive wheel. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1371–1376. IEEE, 2020.

[9] Qifan Tan, Xinyu Zhang, Huaping Liu, Shuyuan Jiao, Mo Zhou, and Jun Li. Multimodal dynamics analysis and control for amphibious fly-drive vehicle. *IEEE/ASME Transactions on Mechatronics*, 26(2):621–632, 2021.

[10] Xiaoyu Wang, Kangyao Huang, Xinyu Zhang, Honglin Sun, Wenzhuo Liu, Huaping Liu, Jun Li, and Pingping Lu. Path planning for air-ground robot considering modal switching point optimization. In *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 87–94. IEEE, 2023.

[11] Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, and Yu Qiao. Scpnet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17642–17651, 2023.

[12] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3991–4001, 2022.

[13] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9087–9098, 2023.

[14] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*, 2023.

[15] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, pages 111–119. IEEE, 2020.

[16] Xin Zhou, Zhepei Wang, Hongkai Ye, Chao Xu, and Fei Gao. Ego-planner: An esdf-free gradient-based local planner for quadrotors. *IEEE Robotics and Automation Letters*, 6(2):478–485, 2020.

[17] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recog-*

nition, pages 9224–9232, 2018.

[18] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 603–612, 2019.

[19] Tong Wu, Yimin Zhu, Lixian Zhang, Jianan Yang, and Yihang Ding. Unified terrestrial/aerial motion planning for hytaqs via nmpc. *IEEE Robotics and Automation Letters*, 8(2):1085–1092, 2023.

[20] Arash Kalantari and Matthew Spenko. Design and experimental validation of hytaq, a hybrid terrestrial and aerial quadrotor. In *2013 IEEE International Conference on Robotics and Automation*, pages 4445–4450. IEEE, 2013.

[21] Mikhail Martynov, Zhanibek Darush, Aleksey Fedoseev, and Dzmitry Tsetserukou. Morphogear: An uav with multi-limb morphogenetic gear for rough-terrain locomotion. In *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 11–16. IEEE, 2023.

[22] Muqing Cao, Xinhang Xu, Shenghai Yuan, Kun Cao, Kangcheng Liu, and Lihua Xie. Doublebee: A hybrid aerial-ground robot with two active wheels. *arXiv preprint arXiv:2303.05075*, 2023.

[23] Ran Cheng, Christopher Agia, Yuan Ren, Xinhai Li, and Liu Bingbing. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *Conference on Robot Learning*, pages 2148–2161. PMLR, 2021.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Shuangjie Xu, Rui Wan, Maosheng Ye, Xiaoyi Zou, and Tongyi Cao. Sparse cross-scale attention network for efficient lidar panoptic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2920–2928, 2022.

[26] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4413–4421, 2018.

[27] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[28] Yu Liu and Michael S Lew. Learning relaxed deep supervision for better edge detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 231–240, 2016.

[29] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2774–2781. IEEE, 2023.

[30] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023.

[31] Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18580–18590, 2023.

[32] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022.

[33] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Practical search techniques in path planning for autonomous driving. *Ann Arbor*, 1001 (48105):18–80, 2008.

[34] C de Boor. Subroutine package for calculating with b-splines, 1971.

[35] Steven G Johnson et al. The nlopt nonlinear-optimization package, 2014.

[36] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] Amovlab. Prometheus UAV open source project. https://github.com/amov-lab/Prometheus.

[39] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017.

SUPPLEMENTARY MATERIAL

In the supplementary material, we discuss additional implementation details and provide more qualitative and quantitative results about *LBSCNet*, *Simulation Experiments* and *Real-World Experiments*. We encourage the reader to browse these results and videos.

A. LBSCNet

In Table V, we present an extensive array of quantitative results, encompassing completion accuracy and semantic segmentation accuracy. Moreover, the visualization of outcomes in the SemantiKITTI dataset validation set is depicted in Fig. 11. It is evident that LBSCNet excels in comparison to other methods with respect to completion and semantic representation of roads, vehicles, buildings, and vegetation, which is in alignment with the findings displayed in Table 3. Despite our semantic segmentation results ranking third among all approaches, we possess superior completion accuracy and real-time performance. This is of paramount importance for Autonomous Ground Robots (AGRs) to accurately and promptly predict the distribution of obstacles in occluded areas during navigation.

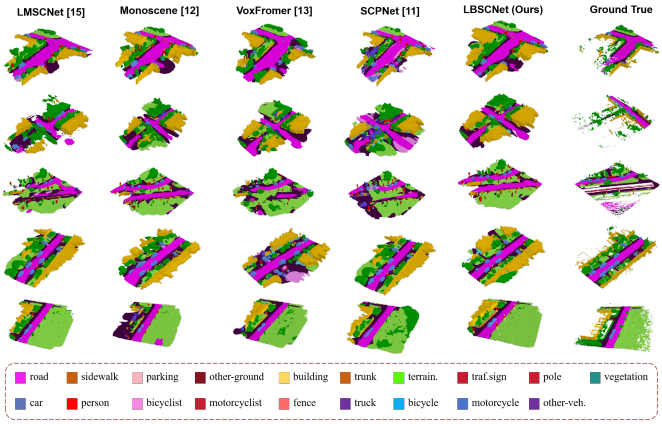


Fig. 11: Qualitative results of our method and others. LBSCNet better captures the scene layout in large-scale scenarios.

Parameter	Value
Battery Capacity	10000 mAh
Battery Weight	1008 g
Rated Power	231 Wh
Operating Voltage	23.05 V
Driving Energy Consumption	≈ 251.45 J/s
Flying Energy Consumption	≈ 988.33 J/s

TABLE IV: Battery and Energy Consumption Parameters

In addition, as illustrated in Fig. 12a, the inference speed comparison of LBSCNet highlights its performance advantages. Owing to their dense 3D convolution design, existing point-cloud-based SSC methods are unable to achieve real-time inference. Concurrently, Fig. 12b demonstrates the inference speed of LBSCNet on various devices. It achieves 20.08 FPS on an RTX 3090 GPU and 17.15 FPS on an RTX 4060 GPU (i.e., in a simulated experiment). Furthermore, when optimized by TensorRT on a Jetson Xavier NX, LBSCNet attains a real-time performance of 10.02 FPS (i.e., in a real-world experiment).

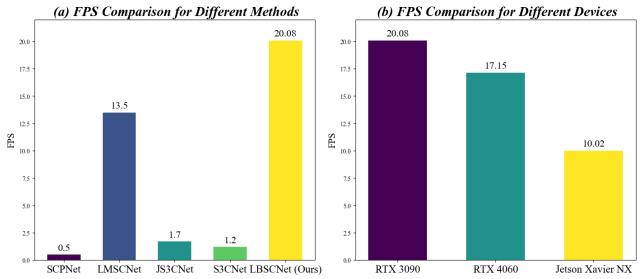
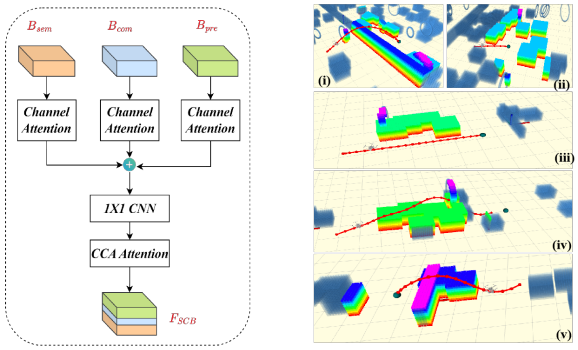


Fig. 12: Qualitative results of our method and others. LBSCNet better captures the scene layout in large-scale scenarios.



(a) Our SCB-Fusion module structure. **(b) Qualitative results in the simulation environment.**

Fig. 13: SCB-Fusion Module and Qualitative results in the simulation environment.

B. Simulation Experiment

More qualitative simulation results are shown in Fig. 13b. The prediction results are updated to the local map with low latency, which allows AG-Planner to avoid these areas in advance when searching for paths.

C. Real-World Experiment

For autonomous navigation, we equip the AGR with the following onboard devices:

- **RealSense D430 depth camera :** This camera provides the depth point clouds for local map fusion. Point clouds are also input to the LBSCNet network.
- **RealSense T265 tracking camera :** This camera provides robust Visual Inertial Odometry (VIO) for UAV state estimation.
- **PX4 Autopilot :** It provides onboard IMU measurements and serves as the inner-loop controller.

Method	LBSCNet (Ours)	SCPNet [11]	VoxFormer [13]	MonoScene [12]	LMSCNet [15]
IoU (%)	**59.71**	56.10	57.69	38.55	54.89
Precision (%)	78.60	68.13	69.95	51.96	**82.21**
Recall (%)	71.29	74.92	**76.70**	59.91	62.29
mIoU	23.58	**36.70**	18.42	12.22	14.13
car (3.92%)	35.80	**46.40**	37.46	24.64	35.41
bicycle (0.03%)	8.00	**33.20**	2.87	0.23	0.00
motorcycle (0.03%)	4.10	**34.90**	1.24	0.20	0.00
truck (0.16%)	4.90	13.80	**10.38**	13.84	3.49
other-veh. (0.20%)	8.10	**29.10**	10.61	2.13	0.00
person (0.07%)	3.40	**28.20**	3.50	1.37	0.00
bicyclist (0.07%)	2.70	**24.70**	3.92	1.00	0.00
motorcyclist (0.05%)	1.80	1.80	0.00	0.00	0.00
road (15.30%)	**71.30**	68.50	66.15	57.11	67.56
parking (1.12%)	39.40	**51.30**	23.96	18.60	13.22
sidewalk (11.13%)	42.90	**49.80**	34.53	27.58	34.20
other-grnd (0.56%)	16.70	**30.70**	0.76	2.00	0.00
building (14.10%)	**43.40**	38.80	29.45	15.97	27.83
fence (3.90%)	31.50	**44.70**	11.15	7.37	4.42
vegetation (39.3%)	45.10	**46.40**	38.07	19.68	33.32
trunk (0.51%)	26.20	**40.10**	12.75	2.57	3.01
terrain (9.17%)	40.90	**48.70**	39.61	31.59	41.51
pole (0.29%)	15.00	**40.40**	15.56	3.79	4.43
traf.-sign (0.08%)	6.80	**25.10**	8.09	2.54	0.00

TABLE V: Quantitative comparison against the state-of-the-art SSC methods.

- **TF Mini Plus :** It provides robust height information for the robot because T265 will drift outdoors, especially in the Z-axis direction.
- **Jetson Xavier NX :** It is an onboard computer with a 6-core NVIDIA CPU and 8 GB RAM. The entire HE-Nav, including map fusion, state estimation, motion planning and control modules, runs on it.

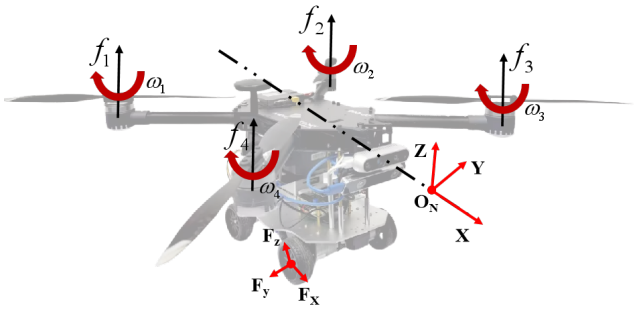


Fig. 14: An equivalent mathematical model for the AGR.

The battery information used by our customized AGR is shown in Table IV. We tested the energy consumption in the real environment to obtain the corresponding energy consumption based on the driving time and flight time recorded in the simulation experiment and the real environment experiment. The relative pose relationship of the AGRs platform was represented by a body coordinate system $(B-xyz)$, and a global coordinate system $(O-XYZ)$, Fig. 14. These were used

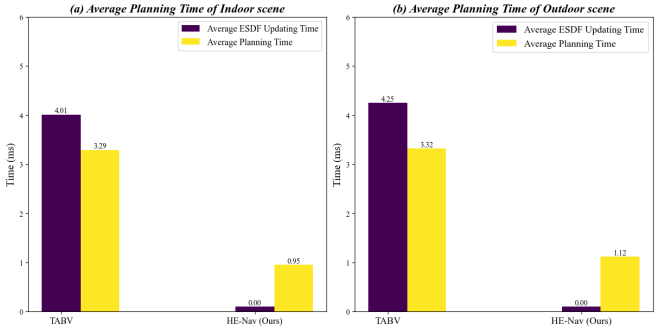


Fig. 15: Total planning time of HE-Nav on Jetson Xavier NX (i.e. ESDF updating time + planning time).

to establish additional coordinate systems, the body's angular velocity in the body coordinate system $\omega_n = [\phi, \theta, \psi]^T$, and angular velocity in global coordinate system $\omega_b = [p, q, r]^T$. where ϕ, θ, ψ represent the roll, pitch, and yaw angles for the body relative to the global coordinate system. The terms p, q, r denote the roll, pitch, and yaw angular velocities for the body relative to the body coordinates, respectively. Furthermore, our customized AGR can continuously fly for 14 minutes or drive on the ground for 55 minutes using a 10000mAh power source. When combined with our HE-Nav navigation system, its energy efficiency is maximized, making it well-suited for continuous operation in complex and occluded wild environments. We encourage reviewers to view our supplementary video for additional information.