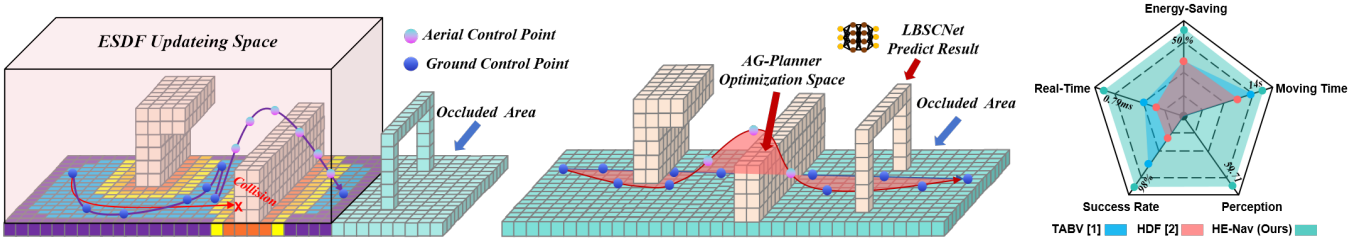
HE-Nav: A High-Performance and Efficient Navigation System for Aerial-Ground Robots

Anonymous Review. Paper-ID [123]



(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 well-suited for search and rescue tasks. Existing AGR navigation systems have made progress in structured indoor scenarios by constructing an Euclidean Signed Distance Field map to search collision-free hybrid paths. However, these ESDF-based systems often show suboptimal performance and efficiency in complex, occluded settings (e.g., forests) due to their inability to perceive unknown areas that are occluded resulting in incomplete local maps coupled with inherent limitations of the path planner.**

In this paper, we present HE-Nav, the first high-performance and efficient navigation system tailored for AGRs. In this novel navigation system, the perception module employs a lightweight semantic scene completion network (LBSCNet), which is guided by a bird's eye view (BEV) and assisted by an exquisitely designed SCB-Fusion module and attention mechanism. This allows it to predict obstacle distribution in occluded areas, generating a complete local map. Subsequently, the AG-Planner, another key component leverages a gradient-based path optimizer and the Kinodynamic A* algorithm to produce safe, energy-saving, and ESDF-free aerial-ground hybrid trajectories.

Extensive simulations and real-world experiments demonstrate that HE-Nav significantly outperforms two recent AGR navigation systems, achieving 34.12% and 19.27% reductions in overall energy consumption while maintaining planning success rates of 98% and 97% in the respective two 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, a crucial component of the autonomous navigation of AGRs is the ***perception module***, which provides a local map for effective navigation. Building upon this, the ***path planner*** search aerial-ground hybrid trajectories that exhibit both *high performance* (i.e., planning success rate) and *efficiency* (i.e., real-time planning and lower energy consumption).

Existing AGR path planner [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 methods 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) due to the narrow field of view in sensor-based mapping. This not only raises collision risks (i.e., *red path* in Fig.1a.) but also prolongs moving time since redundant paths (i.e., *purple path* in Fig.1a). To generate the complete local map for navigation, semantic scene completion (SSC) networks [11] show promise for predicting obstacle distribution and semantics in occluded areas. However, existing networks struggle to strike a balance between completion accuracy and real-time inference. Some use 3D convolution [12, 13] to improve accuracy but unsuitable for resource-limited AGR devices to ensure real-time inference. Others propose lightweight network structures [14] but with significantly reduced accuracy.

Secondly, the existing *path planners* are inefficient. Specifi-

cally, building the ESDF maps generates redundant calculation times that do not meet the real-time requirements of path planning since it takes up about 70% of the time [15], and obstacles only take up 30% of the entire space (in Fig. 1a). Moreover, while the energy costs of flying are typically accounted for in planning, the path-searching algorithms often overlook the energy implications of ground movements, such as steering adjustments. This oversight results in overall energy consumption being inefficient. Notably, the path planner's inefficiency is not only attributed to its inherent flaws but also the limitations of the perception module in providing a complete local map.

In this study, our key insight to tackle these limitations is the co-design of a novel lightweight perception module and efficient AGR path planner. The former achieves accurate completion and real-time reasoning to construct a complete local map, which ensures the collision rate and movement time are reduced. Building upon this, by improving the gradient-based planning method and path search algorithm, our planner can achieve ESDF-free path planning and the planning time and energy consumption are efficient.

Based on our insights into the perception and planning modules, we present ***HE-Nav***, the first *high-performance* and *efficient* navigation system tailored for AGRs, as illustrated in Fig. 2. The system consists of three asynchronous modules: perception, planning, and control. Initially, the perception module employs the lightweight LBSCNet network for real-time 3D scene completion and semantic predictions. Subsequently, these results are seamlessly updated to the local map for path planning. In the planning phase, Our AGR motion planner (AG planner) first generates an initial path (i.e., *blue path* in Fig.1b.), then uses a gradient-based spline optimizer and a post-refinement procedure to optimize the air-ground trajectory, ultimately obtaining an energy-saving, safe, smooth and dynamically feasible trajectory (i.e., *brown path* in Fig.1b.). This trajectory will be sent to the controller for precise tracking.

However, the design of the perception module and path planner faces two major challenges. First, striking a balance between completion accuracy and real-time inference presents a significant hurdle. Dense 3D convolution and feature fusion impede real-time performance, as high-latency inference hinders timely local map updates, thereby undermining path planning efficacy. Although employing sparse 3D convolutions can maintain a lightweight network structure, it fails to capture contextual information or refined features, particularly in occluded regions, resulting in a considerable decline in completion accuracy.

Second, while *Zhou et al.* [15] have developed an ESDF-free path planner specifically for quadcopters, does not adequately address the unique requirements of AGRs, particularly in terms of energy efficiency and dynamic constraints. The flight-centric trajectory generation of their planner leads to increased energy consumption, and considering the non-completeness constraints inherent to AGRs, such planners cannot be naively implemented.

To tackle these challenges, our LBSCNet decouple the learning of semantics and geometry into two distinct branches. This separation is crucial, as semantic context and geometric structure are complementary in SSC tasks [16]. By integrating attention mechanisms, not only is the network's ability to learn semantics and geometry accelerated, but it also captures rich and dense contextual information as well as features of occluded areas. For feature fusion, inspired by [17], we migrated feature fusion to BEV space and proposed the innovative component SCB-Fusion to fuse BEV features, semantic and geometric features in BEV space. This fusion method not only reduces the complexity of calculations but also improves the final completion accuracy.

Subsequently, for the path palnner, we design a novel path search algorithm that adds extra energy cost to motion primitives that require sharp turns on the ground or have destinations in the aerial, to ensure optimal overall energy consumption of the trajectory. Moreover, considering that AGRs' ground movement faces non-holonomic constraints, we impose a cost on ground control points to restrict the curvature of the terrestrial trajectory. Ultimately, an energy-efficient, collision-free hybrid path is generated through a gradient-based trajectory optimizer and a post-refinement procedure.

We first evaluated the LBSCNet network on the SemanticKITTI benchmark, comparing its completion accuracy and inference speed to a state-of-the-art SSC network. Next, we tested HE-Nav in simulated and real indoor and outdoor environments, comparing its performance and efficiency against two baselines. Our evaluation shows that:

- **HE-Nav is high-performance.** HE-Nav achieves a 98% success rate in complex and occluded simulation environments, with a reduced average moving time.
- **HE-Nav is highly efficient in planning.** AG-planner demonstrates an 8x improvement in planning time compared to the ESDF-based method.
- **HE-Nav is energy-efficiency.** AG-planner results in overall energy consumption being reduced by 34.12% and 19.27% in the two scenarios respectively.
- **LBSCNet is accurate and real-time.** LBSCNet enables real-time inference (20.08 FPS) and low-latency updates, achieving state-of-the-art performance (IoU = 59.71) on the SemanticKITTI benchmark.

Our main contributions include the development of a cutting-edge lightweight SSC network (i.e., *LBSCNet*) and an energy-efficient AGR path planner (i.e., *AG-Planner*). The former, featuring a lightweight network structure and innovative components (e.g., SCB-Fusion module), enables real-time complete local map generation. Building on this foundation, the latter achieves ESDF-free planning, substantially reducing planning time. Furthermore, considering AGRs' inherent non-integrity constraints in ground motion, a cost is imposed on ground control points to limit curvature, while the energy-saving Kinodynamic A* algorithm ensures overall path energy efficiency. Ultimately, our HE-Nav generates energy-efficient, safe, smooth, and dynamically feasible hybrid trajectories.

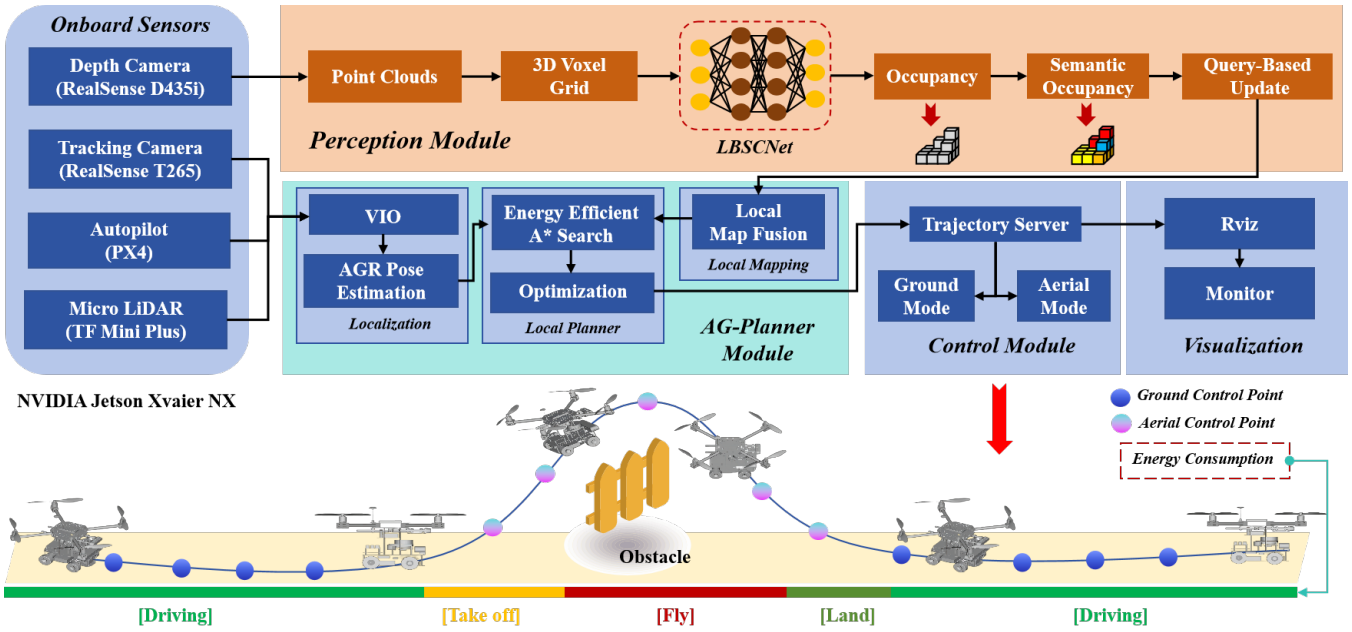


Fig. 2: HE-Nav system architecture. The perception, planning and control modules are deployed on the onboard computer (i.e., NVIDIA Jetson Xavier NX) and run asynchronously.

II. RELATED WORK

A. Motion Planning for AGRs

Numerous researchers have explored various aerial-ground robot configurations, such as incorporating passive wheels [1, 18, 3, 8, 4], cylindrical cages [19], or multi-limb [20] onto drones, while others [9, 6, 5, 10, 21] 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 have adopted a second mechanical structure to further customize our Aerial-Ground Robotic (AGR) system, which has four wheels and four rotors. Although existing research primarily focuses on innovative mechanical structure designs, the area of AGR autonomous navigation remains underexplored. To the best of our knowledge, *Fan et al.* [2] address terrestrial-aerial motion planning. Their approach initially employs the A* algorithm to search for a geometric path as guidance, favouring terrestrial paths by adding extra energy costs to aerial nodes. 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 our navigation system, we propose a new motion planner, namely AG-Palnner, which focuses on estimating the gradient of collision trajectory segments without building ESDF maps, resulting in a significant reduction in computational effort. It also contains an energy-efficient Kinodynamic A* algorithm that searches for energy-efficient guidance paths, and finally models collision, smoothing and dynamic feasibility through a gradient-based spline optimizer to obtain the optimal trajectory, assisted by a post-refinement process to further improve the trajectory of robustness. In addition, for the dynamic constraints of AGR itself, we also added the curvature limit cost of the ground trajectory in the optimization formula to deal with non-holonomic constraints.

B. Occlusion-aware for AGRs

In recent years, the field of semantic scene completion [11, 22]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 for predicting obstacle distribution in occluded areas. 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.* [14] 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. Lastly, *Zuo et al.* [22] proposed PointOcc, which introduces a cylindrical three-perspective view for effective and comprehensive representation of point clouds, along with a PointOcc model for efficient processing.

Despite the remarkable advancements in camera-based and point-cloud-based methods for semantic scene completion, these approaches often demand significant computational resources, rendering them unsuitable for real-time execution on resource-constrained robotic platforms. To address this limitation, we propose a lightweight semantic scene completion network guided by Bird's Eye View (BEV) features, which serves as the perception module for the HE-Nav navigation system.

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 and travelling speed can lead to additional energy consumption. Frequent steering demands extra energy from the motor, resulting in suboptimal energy usage.

III. Perception Module of HE-Nav

In this section, we introduce a lightweight BEV-guided 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 ARG devices, LBSCNet is capable of real-time obstacle distribution prediction in occluded areas. Subsequently, these predictions 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 learning. The specific structures are as follows: ***Semantic Branch*:** Point clouds $P \in \mathbb{R}^{n\times 3}$ undergo processing via a voxelization layer to extract voxel features. Initially, the point cloud is partitioned based on the voxel

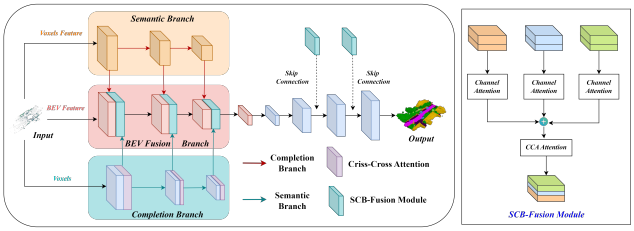


Fig. 3: The overview of the proposed LBSCNet.

resolution s. Points are mapped into voxel space, and their features are aggregated using an aggregation function (e.g., the max function) to obtain a single voxel feature. A multi-layer perceptron (MLP) is then utilized to reduce the dimensionality of this feature vector, yielding the final voxel features V_f with a dense spatial resolution of $L \times W \times H$.

Upon completing voxelization and entering the semantic branch, voxel features V_f are fed into three sparse encoder blocks to obtain sparse semantic features $\{Sem_f^1, Sem_f^2, Sem_f^3\}$. Each sparse encoder block comprises a residual block [24] with sparse convolutions and an SGFE module developed in [25]. The integration of the SGFE module not only enriches voxel features through multi-scale sparse projection and attention mechanisms, capturing both local and global features but also alleviates the computational burden by reducing feature resolution. The semantic branch is optimized using lovasz loss [26] and cross-entropy loss [27]. The semantic loss L_s is the sum of the loss at each stage, expressed as follows:

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

***Completion Branch*:** The completion branch takes occupancy voxels $V \in \mathbb{R}^{1\times L\times W\times H}$ generated by the depth camera point cloud, indicating whether the voxels are occupied. This branch outputs 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 three residual blocks and three GPU memory-efficient criss-cross attention modules. The residual blocks incorporate dense 3D convolutions with a kernel size of $3 \times 3 \times 3$, capturing local geometric nuances. Conversely, the criss-cross attention 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_c for this branch is calculated as follows:

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

***BEV Feature Fusion Branch*:** Prior research has utilized dense 3D convolutions to achieve semantic scene completion by fusing dense 3D features. However, this approach is memory-intensive and requires significant GPU resources, rendering it impractical for deployment on resource-constrained

robotic devices. Leveraging recent advancements in BEV perception, we propose a lightweight BEV feature fusion branch for SSC tasks. By projecting learned semantic and geometric features into the BEV space for fusion, the computational overhead is substantially reduced, enhancing scene completion performance and ensuring real-time inference capabilities.

To project the three-dimensional semantic features $\{Sem_f^1, Sem_f^2, Sem_f^3\}$ into the two-dimensional BEV space, we generate the BEV index based on the voxel index. Features sharing the same BEV index are aggregated using an aggregation function (i.e., the max function) to obtain sparse BEV features. Employing the feature densification function provided by spconv [28], dense BEV features $\{B_f^{sem,0}, B_f^{sem,1}, B_f^{sem,2}, B_f^{sem,3}\}$ are generated based on the BEV index and sparse BEV features.

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 $\{B_f^{com,0}, B_f^{com,1}, B_f^{com,2}, B_f^{com,3}\}$. With semantic and geometric BEV features sharing the same dimensions, our BEV feature fusion network adopts a U-Net architecture with 2D convolutions. The encoder comprises an input layer and four residual blocks. To fully utilize geometric and semantic features at different scales, we designed an SCB-Fusion module to fuse current semantic features, geometric features, and BEV features from the previous layer. The fused features can be expressed as:

$$\begin{aligned} F_{SCB} = \Phi\{&\lambda[N(F_{bev})] \times F_{bev} \\ &+\lambda[N(F_{com})] \times F_{com} \\ &+\lambda[N(F_{sem})] \times F_{sem} \end{aligned} \tag{3}$$

where λ denotes the sigmod function. Φ is the 1×1 convolution.

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

$$L_{total} = 3 \times L_{bev} + L_{sem} + L_{com} \tag{4}$$

IV. Aerial-Ground Motion Planning

In this section, we introduce the novel AG-Planner. It is built on EGO-Planner [15] 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].

A. Energy-Efficient Kinodynamic Hybrid Path Searching

Our AG-Planner first creates a naive "*initial trajectory*" (in Fig.4) 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 employs an energy-efficient kinodynamic A* path search (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 study, we add extra flying and ground-steering energy consumption to the motion primitives (in Fig. 4). 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.

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

B. Gradient-Based B-spline Trajectory Optimization

***B-spline Trajectory Formulation*:** In trajectory optimization, 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 [29] the value of a B-spline can be evaluated as:

$$\Theta(u) = [1, u, ..., u^p] \cdot M_{p_b+1} \cdot [Q_{i-p_b}, Q_{i-p_b+1}, ..., Q_i]^T \tag{5}$$

where M_{p+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:

$$V_i = \frac{Q_{i+1} - Q_i}{\triangle t}, A_i = \frac{V_{i+1} - V_i}{\triangle t}, J_i = \frac{A_{i+1} - A_i}{\triangle t} \tag{6}$$

***Collision Avoidance Force Estimation*:** Inspired by [15], 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. 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:

$$D_{ij} = (Q_i - p_{ij}) \times v_{ij} \tag{7}$$

Because the guide path ι is energy-saving, the generated path is also energy efficient.

***B-spline Trajectory Optimization*:** The basic requirements of the re-planed B-spline are three-folds: smoothness, safety, and dynamical feasibility. Based on the special properties of AGR bimodal, we firstly adopt the following cost terms designed by *Zhou et al.* [15]:

$$\min J_1 = \lambda_s J_s + \lambda_c J_c + \lambda_f (J_v + J_a + J_j) \tag{8}$$

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 [15]. 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:

$$J_n = \sum_{i=1}^{M-1} F_n(Q_{ti}) \tag{9}$$

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

$$F_n(Q_{ti}) = \begin{cases} (C_i - C_{max})^2, C_i > C_{max}, \\ 0, C_i \leq C_{max} \end{cases} \tag{10}$$

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:

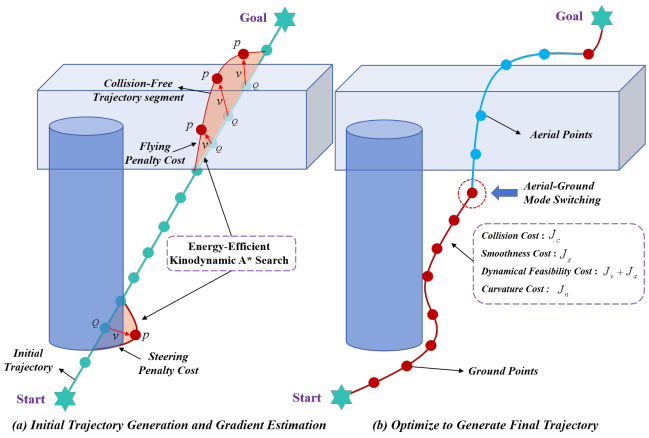


Fig. 4: Illustration of AG-Planner and topological trajectory generation.

$$\min J_{all} = \lambda_s J_s + \lambda_c J_c + \lambda_f (J_v + J_a + J_j) + \lambda_n J_n$$
$$s.t. \quad \begin{cases} J_s = \sum_{i=1}^{N_c-1} \|A_i\|_2^2 + \sum_{i=1}^{N_c-2} \|J_i\|_2^2 \\ J_c = \sum_{i=1}^{N_c} j_c(Q_i) \\ J_v = \sum_{i=1}^{N_c} \omega_v F(V_i) \\ J_a = \sum_{i=1}^{N_c-1} \omega_a F(A_i) \\ J_j = \sum_{i=1}^{N_c-2} \omega_j F(J_i) \\ J_n = \sum_{i=1}^{M-1} F_n(Q_{ti}) \end{cases} \tag{11}$$

The optimization problem is solved by a non-linear optimization solver NLopt [30]. After path planning is completed, a setpoint on the generated trajectory is selected according to the current timestamp and then sent to the controller. An aerial setpoint includes the yaw angle and 3D position, velocity, and acceleration. A terrestrial one includes the 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 *Offboard Mode* to enter the flight state.

V. Evaluation

In this section, we first evaluate the perception module (i.e., LBSCNet) on the SemanticKITTI benchmark for its accuracy in SSC tasks, as well as its real-time inference and update capabilities. We then integrate the perception module and the planning module by deploying a pre-trained model offline, forming a complete HE-Nav system. Subsequently, we conduct experiments in both simulated and real-world environments to assess the performance of the aerial-ground robot (AGR) when using HE-Nav for autonomous navigation, focusing on ***performance*** metrics (i.e. planning success rate, total movement time) and ***efficiency*** (planning time and energy consumption).

A. Evaluation setup

Perception Module: For the training and testing of LBSCNet, we carried out the process on a server equipped with 4 NVIDIA RTX 3090 GPUs, 128GB of memory, and an Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz. The large-scale outdoor scenarios dataset SemanticKITTI [31] is the dataset used. We trained the LBSCNet model for 80 epochs on a single NVIDIA 3090 with a batch size of 12, employing the Adam optimizer [32] with an initial learning rate of 0.001, and augmenting the input point cloud by randomly flipping along the x-y axis during the training process. Ultimately, we deployed the pre-trained model offline with the best completion accuracy to predict occlusion areas.

Navigation Simulation Experiment: Simulation experiments were conducted on a laptop with Ubuntu 20.04, i9-13900HX CPU, and NVIDIA RTX 4060 GPU. We simulated aerial-ground robot navigation in complex scenarios, consisting of a $20m \times 20m$ room and a $3m \times 30m$ corridor with numerous random obstacles, creating occluded spaces and unknown areas. The AGR's task was to navigate from a starting point to a designated destination without collision. We also record the total movement and path planning time to compare with the baseline.

Indoor and Outdoor Real-world Experiment: We deployed HE-Nav on a custom AGR platform (in Fig. 5) for real-world indoor and outdoor environment experiments, and also tested the average energy consumption per second of AGR under driving, flying and hovering, as shown in Table III. This platform utilizes the Prometheus software [33] and is equipped with a RealSense D435i depth camera and a T265 camera. Additionally, it features a Jetson Xavier NX onboard computer to run the HE-Nav. More detailed hardware specifications are provided in the supplementary materials.

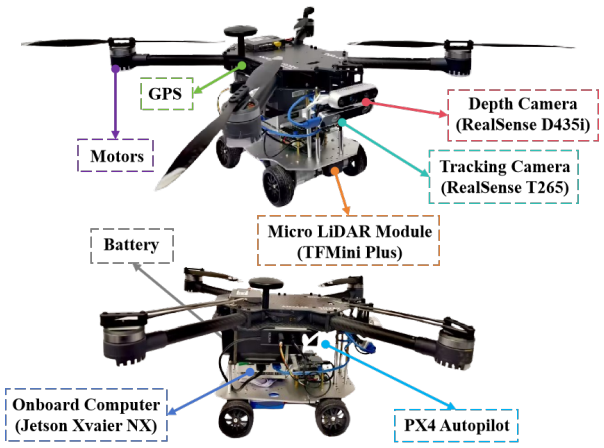


Fig. 5: The detailed composition of the robot platform.

Metrics: For the perception module, we use intersection over union (IoU) to evaluate scene completion quality and the mean IoU (mIoU) of 19 semantic classes to assess the performance of semantic segmentation. 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 (t), and efficient metrics planning time (ms) as well as energy consumption (W).

***Baseline methods*:** For the perception module, we compare LBSCNet against the state-of-the-art SSC methods with public resources: (1) a camera-based SSC method MonoScene [12] and VoxFormer[13], (2) point-cloud-based SSC methods including LMSCNet [14], and SSCNet [34] 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].

Method	*IoU*	*mIoU*	*Prec.*	*Recall*	*FPS*
SSCNet [34]	53.20	14.55	59.13	**84.15**	12.00
LMSCNet [14]	55.32	17.01	77.11	66.19	13.50
LMSCNet-SS [14]	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 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 6.45% in IoU and runs approximately 20 times faster.

The exceptional accuracy and real-time inference performance of our LBSCNet can be attributed to the innovative semantic and completion decoupling network structure, which leverages contextual semantic information to enhance scene understanding and completion. Additionally, the incorporation of the novel SCB-fusion module and CCA module allows the network to remain lightweight while significantly improving completion accuracy by capturing contextual features and learning long-distance dependencies. Furthermore, our LBSCNet exhibits low latency and real-time operation (20.08 FPS) due to the utilization of sparse 3D convolutions and lightweight BEV feature fusion within the network. Consequently, LBSCNet is well-suited for real-time perception in ARG navigation systems.

Qualitative Results: We provide visualizations on the SemanticKITTI validation set, as depicted in Fig. 5, and include results from LMSCNet [14], Monoscene[12], VoxFormer[13], and SCPNet [11] for a comprehensive comparison. 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.

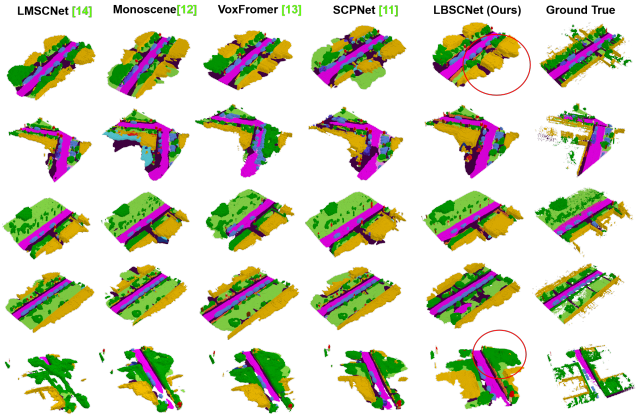


Fig. 6: Qualitative results of our method and others. LBSCNet better captures the scene layout in large-scale scenarios.

Ablation Study: Ablation studies on the SemanticKITTI validation set (in Table II) highlight the significance of two key components in our network: self-attention mechanisms and the SCB-Fusion Module. The CCA mechanism substantially impacts completion and semantic prediction by effectively aggregating context across rows and columns. *Without CCA* causes a 3.86% and 7.48% drop for completion and semantic completion, respectively. Meanwhile, SCB-Fusion captures local scene features, such as occluded areas, with low computational overhead. *Without SCB-Fusion* leads to a 2.47% decline in IoU.

Method	IoU ↑	mIoU ↑
LBSCNet (ours)	54.92	17.69
w/o SCB-Fusion Module	54.15	17.26
w/o Criss-Cross Attention	52.80	16.37

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

C. Simulated Air-Ground Robot Navigation

We conducted a comparative analysis of our HE-Nav navigation system against TABV [1] and HDF [2] in a square room and corridor scenario. 100 trials with varying obstacle placements, we recorded the moving time, length, planning time and success rate (i.e., no collisions). In addition, based on the recorded flight time and driving time, combined with the real-world energy consumption of our customized AGR (in Table III), the energy consumption results of navigation in the simulation environment were obtained.

Parameter	Value
Battery Capacity	10000 mAh
Battery Weight	1008 g
Rated Power	231 Wh
Operating Voltage	23.05 V
Driving Energy Consumption	≈ 252.00 W
Hovering Energy Consumption	≈ 533.08 W
Flying Energy Consumption	≈ 990.00 W

TABLE III: Battery and Energy Consumption Parameters

As illustrated in Fig. 7, our HE-Nav system exhibits outstanding planning success rates of 98% and 97%, along with average movement times of 11.2s and 14.2s in square rooms and corridors, respectively. Owing to the elimination of redundant ESDF calculations, the planning time for our HE-Nav system is accelerated 6x compared to TABV in a square room. This exceptional performance is attributed to our advanced LBSCNet's ability to predict obstacle distribution in occluded areas, allowing planners to effectively bypass these regions and significantly reduce collision risks. Moreover, our path planner seamlessly integrates with the Kinodynamic A* algorithm, achieving the lowest average energy consumption (i.e., 6186.2 W and 3478.4 W). In square scenarios, HE-Nav attains a 34.12% reduction in energy consumption compared to state-of-the-art TABV planning methods, while in corridors, our HE-Nav system realizes a 19.27% reduction in energy consumption compared to its counterparts.

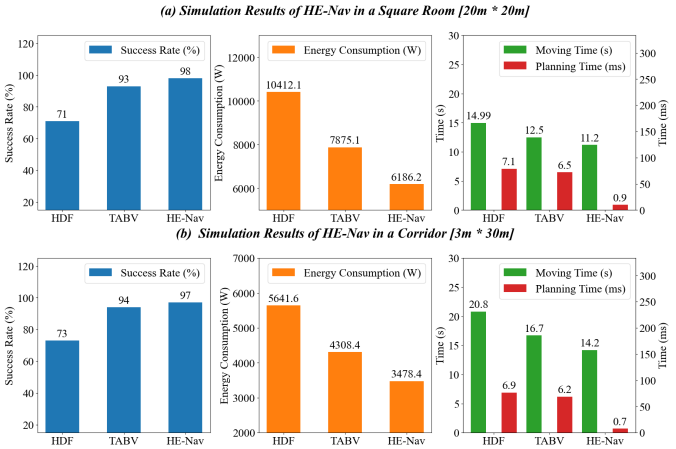
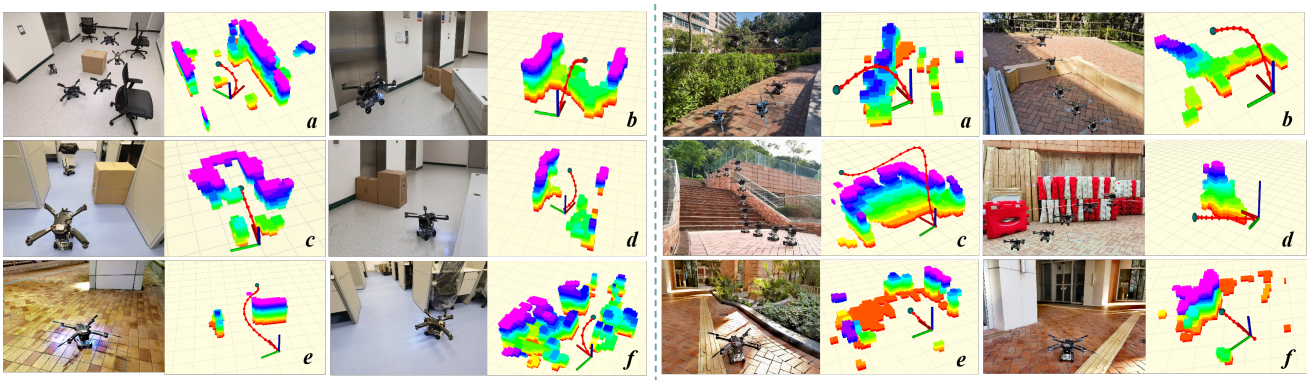


Fig. 7: Quantitative results of HE-Nav in two simulation scenarios.

As depicted in Fig. 9, the path generated by the HDF fails to effectively consider both smoothness and dynamic feasibility. Additionally, the TABV path primarily focuses on the energy consumption associated with flight, which results



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

Fig. 8: Four methods were used to plan paths in a simulated square room. AGRNav demonstrates the ability to predict the distribution of obstacles in occluded areas.

in premature flight actions and consequently leads to increased energy consumption, rendering the overall energy consumption suboptimal. This lack of perception causes TABV to encounter difficulties in pathfinding, further exacerbating energy consumption. In contrast, our HE-Nav system adeptly addresses this shortcoming through its ability to perceive and predict occlusions, thereby optimizing both path planning and energy consumption.

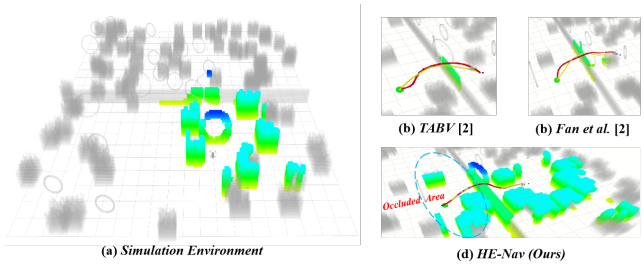


Fig. 9: Qualitative results of path planning and occlusion prediction in simulation environment.

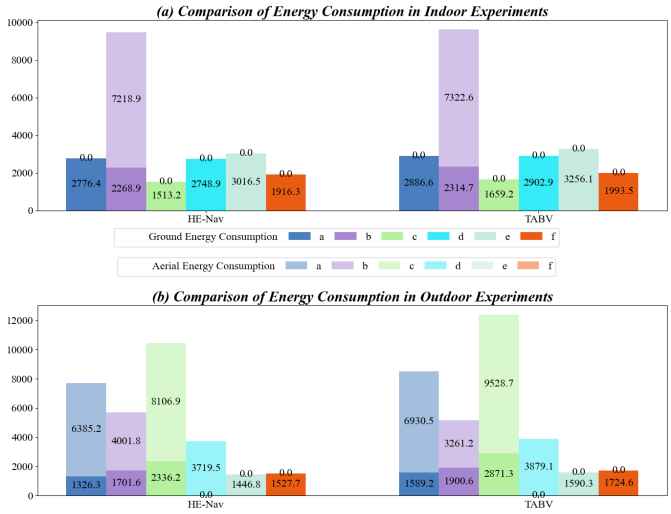


Fig. 10: Quantitative results of indoor and outdoor real environmental energy consumption.

D. Real-world Air-Ground Robot Navigation

We assess HE-Nav's performance and energy efficiency across 6 indoor and 6 outdoor scenarios (Figure 8). As depicted in Fig. 10, HE-Nav consistently exhibits lower indoor average energy consumption than TABV [1], for example, in scenarios d, e, and f, the ground energy consumption of our HE-Nav is reduced by 5.3%, 7.3% and 3.8% compared with TABV. This is primarily attributed to the incorporation of extra turning penalty terms in the ground segment, which effectively curtails energy usage in ground mode by minimizing high-angle turning paths.

For outdoor scenarios like scene c, a 15.78% reduction in energy consumption compared to TABV [1] is achieved, owing to the smooth aerial path that minimizes flight energy consumption. Furthermore, the removal of ESDF considerably streamlines path planning time. On Jetson Xavier NX, planning time is reduced to 1.2 milliseconds, a fivefold improvement over ESDF-based TABV. Additional qualitative and quantitative results can be found in the supplementary material, i.e., in Section VII-C.

VI. Conclusion

we have presented HE-Nav, a high-performance and efficient 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. 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] 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.

[15] 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.

[16] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3101–3109, 2021.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

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

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

[31] 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.

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

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

[34] 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 IV, 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.

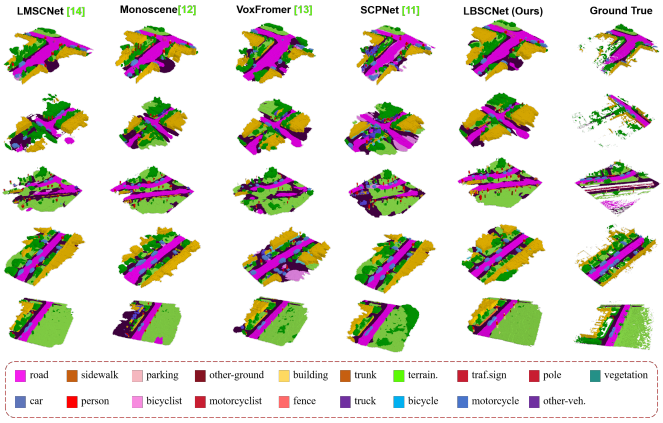


Fig. 11: Qualitative results of our method and others. LBSCNet better captures the scene layout in large-scale scenarios.

In addition, as illustrated in Fig. 13a, 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. 13b demonstrates the inference speed of LBSCNet on various devices. It achieves 20.08 FPS on an RTX 3090 GPU and 19.85 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 11.55 FPS (i.e., in a real-world experiment).

B. Simulation Experiment

C. Real-World Experiment

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

Method	LBSCNet (Ours)	SCPNet [11]	VoxFormer [13]	MonoScene [12]	LMSCNet [14]
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 IV: Quantitative comparison against the state-of-the-art SSC methods.

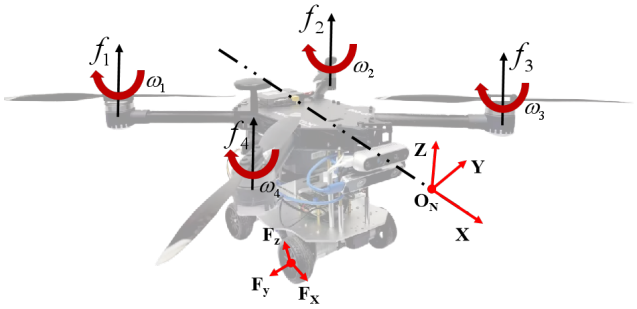


Fig. 12: Qualitative results of path planning and occlusion prediction in simulation environment.

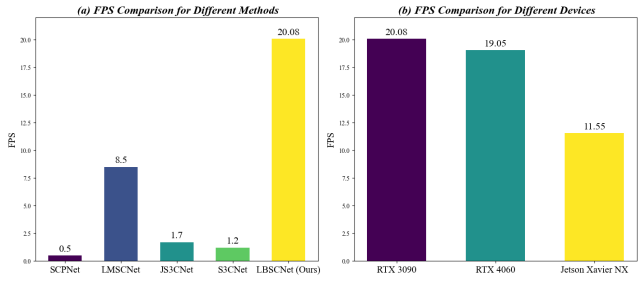


Fig. 13: Qualitative results of our method and others. LBSCNet better captures the scene layout in large-scale scenarios.

- **RealSense D430 depth camera :** This camera provides the point clouds for local map fusion.
- **RealSense T261 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.
- **TF Mini Plus :** It provides height information.
- **Jetson Xavier NX :** It is an onboard computer with 6-core NVIDIA Carmel CPU and 8 GB RAM. The entire HE-Nav, including map fusion, state estimation, motion planning and control modules, runs on it.

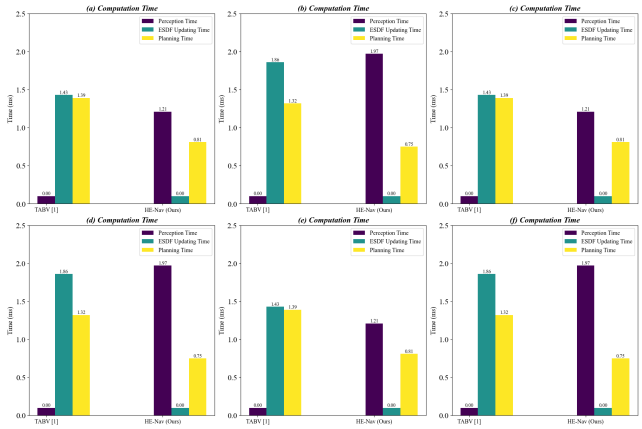


Fig. 14: Qualitative results of path planning and occlusion prediction in simulation environment.