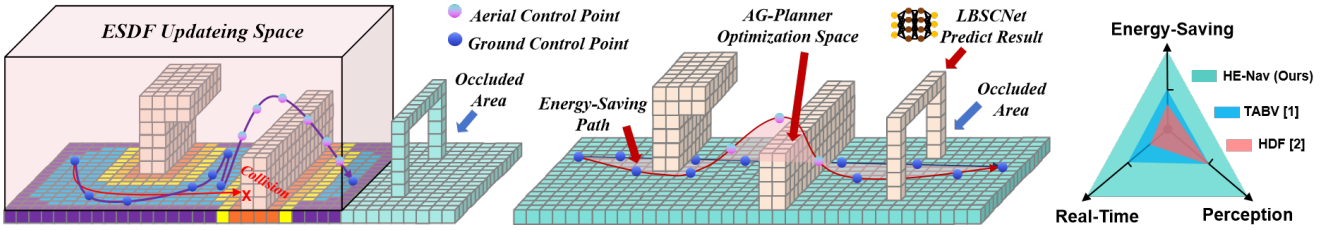
HE-Nav: A **H**igh-performance and **E**nergy-efficient **Nav**igation System for Aerial-Ground Robots

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. While existing Euclidean Signed Distance Field (ESDF)-based navigation systems have made progress in structured indoor scenarios by constructing an ESDF map to search collision-free hybrid paths, these systems often show suboptimal performance and energy consumption 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 energy-efficient navigation system tailored for AGRs. This novel system incorporates a lightweight Bird's Eye View (BEV)-guided semantic scene completion network called LBSCNet within its perception module. Assisted by an exquisitely designed SCB-Fusion module and attention mechanisms that facilitate real-time prediction of obstacle distributions within occluded regions ahead of time. Furthermore, HE-Nav introduces an advanced AGR path planner which integrates both gradient-based path optimiser and Kinodynamic A* algorithms to produce safe and energy-saving trajectories.

Extensive simulations and real-world experiments demonstrate that HE-Nav significantly outperforms two recent AGR navigation systems, achieving up to 50% reduction in overall energy consumption while maintaining superior performance metrics including a remarkable 98% success rate alongside shorter average planning times and moving times. The code and hardware configuration will be released.

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, achieved by seamlessly switching between aerial and ground modes as needed. By mode switching, AGRs can effectively perform hybrid locomotion (i.e., flying and driving) in the above challenging tasks. As a key enabler of AGR's autonomous navigation system, three basic modules (i.e., perception, planning, and control) asynchronous work to provide a local map, search collision-free trajectories and ensure accurate trajectory tracking and smooth mode transition in 3D space, which realize optimal performance (i.e., low collision risk, short path planning time and movement time) and energy efficiency.

Existing ESDF-based navigation systems [1, 4] first employed sensors (e.g., cameras) to perceive surrounding environments and establish local Euclidean Signed Distance Field (ESDF) maps (in Fig. 1a). Next, based on the ESDF map, an aerial-ground hybrid path planner is utilized to search for collision-free trajectories. This planner favours ground paths during the search process and only switches to aerial mode when necessary (e.g., confronting impassable obstacles), thereby promoting energy efficiency.

Unfortunately, While ESDF-based methods have proven successful in structured indoor scenarios, existing works face two limitations when navigating in complex and occluded environments (e.g., disaster zones or wilderness areas). Firstly, sensor-based mapping is restricted by a narrow field of view, leading to incomplete local maps with occlusion-induced unknown areas. This not only significantly increases the risk of collision (e.g., the *red path* in Fig. 1a.) but also boosts movement time by redundancy trajectories (i.e., the *purple path* in Fig. 1a). Drawing inspiration from autonomous driving systems, semantic scene completion (SSC) networks [11] have the potential to predict obstacle distribution and the semantics of occluded areas. However, the use of 3D convolution [12] in these networks prevents them from running on AGR devices

with limited resources.

Secondly, the existing path palnner (e.g., [1, 2]) still has inherent drawbacks which lead to suboptimal performance and energy consumption. Specifically, 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 [13], 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 also results in overall energy efficiency being suboptimal.

The above-mentioned limitations inspire us to rethink the design of the perception module and planning module in the AGRs navigation system. Firstly, for the SSC network that predicts occlusion in the perception module, our key insights are to migrate dense feature fusion from 3D space to the BEV space, supported by 3D sparse convolution [14], ensuring real-time inference on AGR devices. Moreover, To enhance the accuracy of scene completion, especially for occlusion areas, we 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 [15]. 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.

Subsequently, for the path palnner, while *Zhou et al.* [13] have developed an ESDF-free path planner designed for quadcopters, this planner does not align with the specific needs of AGRs, particularly in terms of energy (i.e., flight-centric trajectory generation leads to higher energy usage) and dynamics constraints. Therefore, our key insight is to 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 air to generate energy-efficient collision-free guidance paths. In addition, considering the ground movement of AGRs faced non-holonomic constraints, we enforced a cost on ground control points to limit the curvature of the terrestrial trajectory.

Based on our insights into the perception and planning modules, we present ***HE-Nav***, the first *high-performance* and *energy-efficient* autonomous 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) commences by generating an initial path, and then we develop an energy-saving Kinodynamic A* algorithm to generate local collision-free guidance trajectory segments for trajectory segments within obstacles, taking into account factors such as flight energy consumption and ground steering. Next, the planner models collision, smoothness, and dynamical feasibility costs, effectively wrapping the trajectory around obstacles and reducing computation time, since no need to create an ESDF map. Finally, a post-refinement procedure optimizes the aerial-ground trajectory while maintaining dynamic feasibility. The refined trajectory is sent to the controller for precise tracking and hybrid motion.

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 energy consumption against two baseline approaches (TABV [1] and HDF [2]). Our evaluation shows that:

- **HE-Nav achieves high performance.** HE-Nav achieves a 98% success rate in complex and occluded simulation environments, with a shorter average moving time (≈12s) and an 8x improvement in planning time compared to the ESDF-based method.
- **HE-Nav is energy-efficient.** By novel energy-efficient Kinodynamics A* algorithm resulting in a 50% decrease in energy consumption.
- **HE-Nav is real-time.** LBSCNet enables real-time inference (20.08 FPS) and low-latency updates and achieves state-of-the-art performance (IoU = 59.71) on the SemanticKITTI benchmark.

Our main contributions are LBSCNet, a novel lightweight SSC network and a novel hierarchical path planner (i.e., AG-Planner) tailored for AGRs. The former, with its lightweight network structure and the incorporation of innovative components such as SCB-Fusion and criss-cross attention, can predict obstacle distribution in occluded areas in real-time. The latter uses a gradient-based method to achieve ESDF-free, which significantly reduces the planning time. It is assisted by the energy-saving Kinodynamic A* algorithm to ensure energy saving of the overall path. In addition, considering that AGRs face inherent non-integrity constraints in ground motion, a cost is also added to the ground control points to limit curvature. Ultimately 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 vehicle configurations, such as incorporating passive wheels [1, 16, 3, 8, 4], cylindrical cages [17], or multi-limb [18] onto drones, while others [9, 6, 5, 10, 19] have integrated rotors with wheeled robots to achieve dual-modal (i.e., aerial and terrestrial) 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.

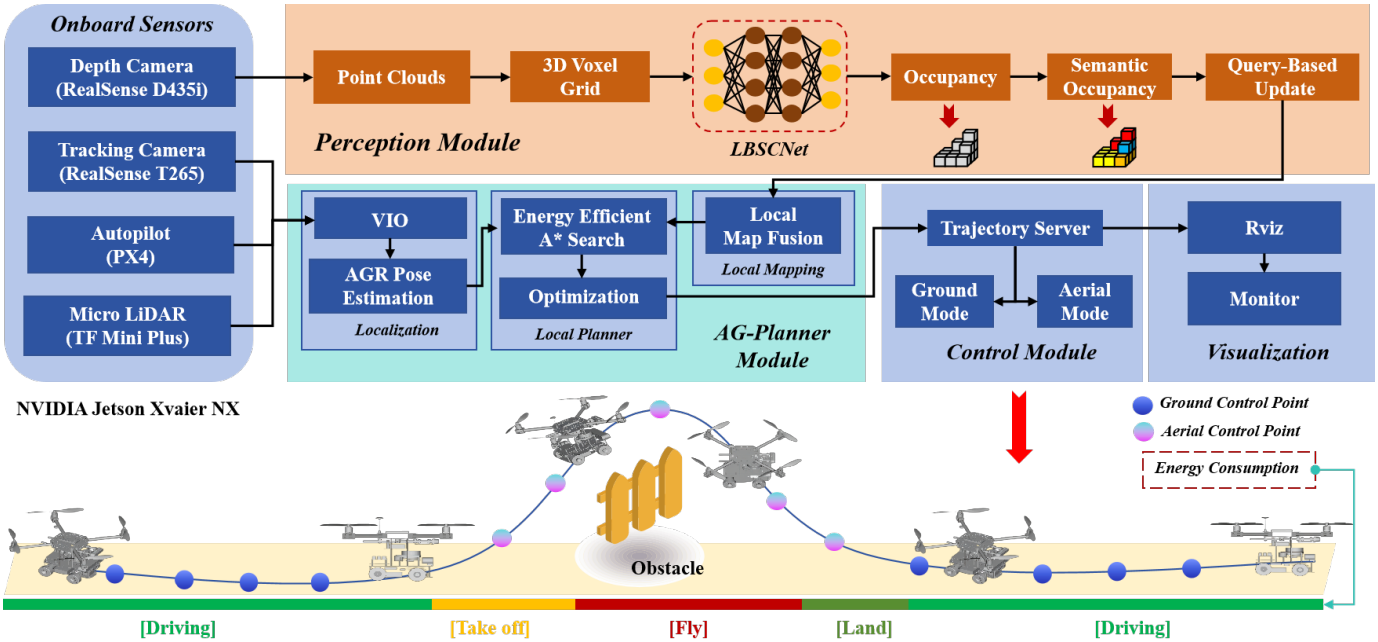


Fig. 2: HE-Nav system architecture. The perception, planning and control modules are deployed on the onboard computer (i.e., NVIDIA Jetson Xavier NX) and run asynchronously.

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 advanced awareness of occluded areas lead to a low success rate in path planning and increased energy consumption.

In the proposed planning method, We use gradient-based methods to search for paths without building an ESDF map. We use kinodynamic path searching instead, and formulate a nonlinear optimization problem to refine the kinodynamic path. Apart from smoothness, collision avoidance, and dynamical feasibility cost, we also add a curvature limit cost for terrestrial trajectories in the optimization formulation to handle the nonholonomic constraint.

B. Occlusion-aware for AGRs

In recent years, the field of semantic scene completion has witnessed significant advancements, particularly in addressing the challenges posed by limited fields of view (FOV) of robot sensors and the obstruction of obstacles in complex and unknown environments. These advancements have led to the development of various LiDAR-based and Camera-based methods for predicting and perceiving occlusion 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.* [11] is VoxFormer, a Transformer-based semantic scene completion framework capable of generating complete 3D volume semantics using only 2D images. Further, *Dong et al.*[20] developed CVSformer, which employs multi-view feature synthesis and cross-view transformers for learning cross-view object relationships, ultimately enhancing the prediction of geometric occupancy and semantic labels of voxels.

On the other hand, LiDAR-based methods have also made significant strides. *Cheng et al.* [21] proposed S3CNet, a method designed to predict semantically complete scenes from a single unified LiDAR point cloud. *Roldao et al.* [22] 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.* [23] 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.* [24] 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 LiDAR-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 EH-Nav navigation system.

This module efficiently predicts the distribution of obstacles in occluded areas, ensuring seamless navigation in complex environments while maintaining low computational overhead, making it an ideal solution for resource-limited robotic devices.

C. Energy-Efficient for AGRs

Energy efficiency is vital for aerial-ground robots since it directly impacts their operational capabilities, 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. Moreover, their frameworks lack adequate perception of occluded areas, causing the robot to face corner cases such as entering a dead-end with no ground path. In such scenarios, the robot is forced to either take off or retreat, leading to redundant paths and suboptimal energy consumption. To address these limitations, our novel HE-Nav system incorporates an advanced perception module and planning module designed for energy efficiency.

III. Perception Module of HE-Nav

In this section, we propose a lightweight dual-branch SSC network, comprising a semantic branch and a completion branch, as an alternative to existing memory-intensive SSC networks that jointly predict geometry and semantics. By decoupling the learning process of semantics and completion (or geometry), our approach allows the network to concentrate on specific feature types, resulting in more efficient parameter utilization. Furthermore, our focus is on predicting and completing occluded areas. We gained two key insights: (1) Occluded areas exhibit discontinuous features due to the absence of visual perception. To achieve higher precision completion, the network must be capable of learning long-distance dependencies and capturing more detailed features. (2) Dense feature fusion in 3D space hampers the network's real-time performance. Drawing inspiration from [1], we can shift feature fusion to the BEV space, significantly reducing computational demands.

As a result, we introduce criss-cross attention to the completion branch, enabling the capture of more refined features, enhancing the quality of completion, improving long-distance dependency learning, maintaining performance, and reducing memory consumption. We also propose the BEV fusion branch, which includes an essential component (i.e., the SCB-Fusion module) designed to fuse three types of features (i.e., BEV features, semantic features, and geometric features). This ultimately leads to dense 3D scene completion and semantic prediction.

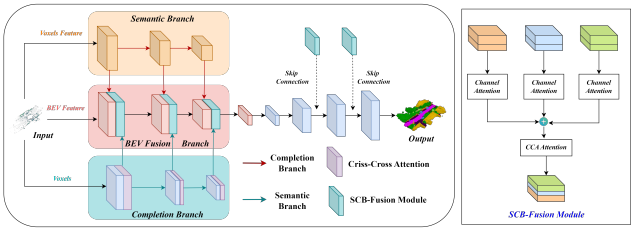


Fig. 3: Four methods were used to plan paths in a simulated square room. AGRNav demonstrates the ability to predict the distribution of obstacles in occluded areas.

A. LBSCNet Network Structure

As discussed earlier, we need to design a lightweight SSC network to serve as the perception module for our navigation system, HE-Nav. Therefore, we propose the Lightweight BEV-Guided 3D Scene Completion Network (LBSCNet), as shown in Fig. 3. By deploying its pre-trained model on robot devices, LBSCNet can predict the distribution of obstacles in occluded areas in real-time. The prediction results are then integrated into a local map, which is used for path planning.The specific encoder and decoder structures are as follows:

Semantic Branch. Point clouds $P \in \mathbb{R}^{n\times 3}$ are processed by a voxelization layer to extract voxel features, which are then fed into the semantic branch. Specifically, the point cloud is first partitioned according to the voxel resolution s. Points are mapped into the voxel space, and their features are subsequently aggregated using an aggregation function (e.g., the max function) to obtain a single voxel feature. Finally, a multi-layer perceptron (MLP) is employed to reduce the dimensionality of this feature vector, resulting in the final voxel features V_f with a dense spatial resolution of $L \times W \times H$. After completing voxelization and entering the semantic branch, the 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 consists of a residual block [25] with sparse convolutions and an SGFE module developed in [26]. The addition of the SGFE module not only enhances the features of voxels, thanks to the multi-scale sparse projection and attention mechanisms that capture more local and global features but also reduces the computational burden by reducing feature resolution. We use lovasz loss [39] and cross-entropy loss to optimize the semantic branch. The semantic loss Ls is the summation of the loss of each stage, which can be expressed as:

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

Completion Branch. The completion branch takes the occupancy voxels $V \in \mathbb{R}^{1\times L\times W\times H}$ generated by the depth camera point cloud, which indicates whether the voxels are occupied or not. This branch outputs multi-scale dense completion features $\{Com_f^1, Com_f^2, Com_f^3\}$ to provide more detailed geometric information. As shown in Fig. 3, the completion branch is composed of three residual blocks and a GPU memory-efficient criss-cross attention module. The residual blocks consist of dense 3D convolutions with a kernel size of $3 \times 3 \times 3$, which are responsible for capturing local geometric details. In contrast, the criss-cross attention module is designed to capture long-range dependencies by gathering contextual information in both horizontal and vertical directions, thus enhancing

the completion features with global context. Similar to the semantic branch, the training loss L_c for this branch is computed by:

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

BEV Feature Fusion Branch. Previous research has employed dense 3D convolutions to fuse dense 3D features to achieve semantic scene completion in 3D environments. This approach, however, is memory-intensive and often necessitates substantial GPU resources. Consequently, it is impractical to deploy and utilize such networks on robotic devices with limited resources. In light of recent advancements in BEV perception, we propose a lightweight BEV feature fusion module for the Semantic Scene Completion (SSC) task. By projecting the learned semantic and geometric features into the BEV space for fusion, the computational overhead is significantly reduced. This not only enhances scene completion performance but also ensures real-time inference capabilities. Specifically, we need to project the features learned in the three-dimensional space into the two-dimensional BEV space. For the semantic features $\{Sem_f^1, Sem_f^2, Sem_f^3\}$, we generate the BEV index based on the voxel index. Subsequently, features sharing the same BEV index are aggregated using an aggregation function (i.e., the max function) to obtain sparse BEV features. Finally, with the assistance of the feature densification function provided by spconv [27], dense BEV features $\left\{B_f^{sem,0}, B_f^{sem,1}, B_f^{sem,2}, B_f^{sem,3}\right\}$ are generated based on the BEV index and sparse BEV features. Regarding geometric features $\{Com_f^1, Com_f^2, Com_f^3\}$, we stack dense 3D features along the $z-axis$. Then, 2D convolution is employed to reduce the feature dimension and generate dense BEV features $\left\{B_f^{com,0}, B_f^{com,1}, B_f^{com,2}, B_f^{com,3}\right\}$. Lastly, semantic BEV features and geometric BEV features have the same dimensions.our BEV feature fusion network is U-Net architecture with 2D convolutions. The encoder consists of an input layer and four residual blocks. In order to make full use of geometric and semantic features at different scales, we also designed a BSC-FR module to fuse the current semantic features, geometric features and BEV features of the previous layer.The fused features can be expressed as:

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

where λ denotes the sigmod function. Φ is the $1X1$ convolution. **Total Loss Function.** We train the whole network end-to-end. The multi-task loss L_all is expressed as :

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

IV. Gradient-based Aerial-Ground Motion Planning

In this section, we introduce our innovative gradient-based energy-efficient AG-Planner. The first part of our planner creates an initial trajectory that overlooks obstacles by randomly adding coordinate points and applying the min-snap method, considering the positions of both the starting point and the target point. Following that, the back end of our planner employs an energy-efficient kinodynamic path search to establish a safe aerial-ground hybrid guidance path. We also use a gradient-based spline optimizer and an additional refinement process to refine the path further. This approach leads to the generation of the final hybrid aerial-ground path. The problem formulation in this paper is based on the current state-of-the-art aerial-ground planning framework TABV[1].

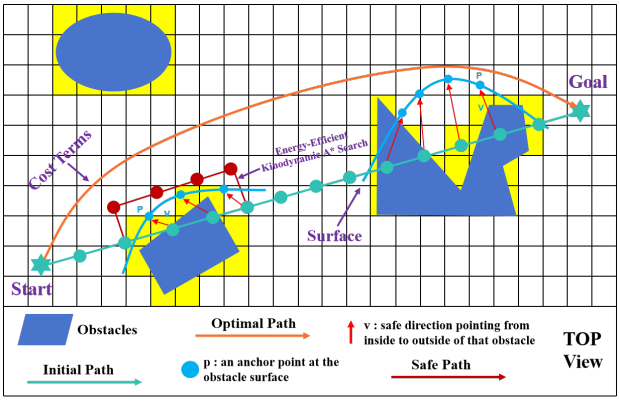


Fig. 4: Illustration of AG-Planner and topological trajectory generation.

A. Collision Cost Estimation and Energy-Efficient Path Search

In this paper, 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 $Q_i \in \mathbb{R}^3, t_m \in \mathbb{R}, M = N_c + p_b$. 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 denoted as Q_{aerial}.

Our AG-planner first generates a "*collision trajectory*" that ignores obstacles based on the starting point and target point and finds the path segments where the collision occurs. These segments are composed of collision points. We then propose the energy-efficient Kinodynamic A* path search algorithm, which adds an extra energy consumption cost (i.e., fly, ground speed and yaw) to the motion primitives, as shown in Algorithm 1. The algorithm will search for a collision-free aerial-ground hybrid path τ, which also energy-saving for ground mode and fly mode.

Inspired by [13], For each control point on the collision trajectory segment, vector v 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. Then the obstacle distance D_{ij} if i^{th} control point Q_i to j^{th} obstacle is defined as:

$$D_{ij} = (Q_i - p_{ij}) \times v_{ij} \tag{5}$$

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

B. Post-trajectory refinement procedure

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:

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

Based on the special properties of AGR bimodal, we let the objective J make out of four terms, and the problem becomes:

$$\min J = \lambda_s J_s + \lambda_c J_c + \lambda_f (J_v + J_a) + \lambda_n J_n \tag{7}$$

where J_s is the smoothness penalty, J_c is for collision, and J_v, J_a are dynamical feasibility costs that limit velocity and acceleration.

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**
        $g_n -= x.penalty\_g$
        $g_n += fly\_cost * z + fly\_cost\_base$
        // calculate fly penalty cost
        $penalty\_g = fly\_cost * z + fly\_cost\_base$
        $next\_motion\_state = true$
      end
      else
        // next node driving
        $g_n -= x.penalty\_g$
        $penalty\_g = 0$
        $next\_motion\_state = false$
      end
      $f_n = g_n + \lambda * estimateHeuristic(n, x_g)$
      if $n \notin O \cup C$ **then**
        $n.updateCost(g_n, penalty\_g, f_n)$
        $O.push(n)$
      end
    end
  end
end
return *null* // Cannot find a valid path

$\lambda_s, \lambda_c, \lambda_f, \lambda_n$ are weights for each cost terms. Based on our observations, AGR faces the non-holonomic constraints (i.e., AGR's ground velocity vector must be aligned with its yaw angle), and curvature limitations (i.e., minimizing tracking errors during sharp turns) when driving on the ground. Therefore, a cost for curvature needs to be added, that is J_n can be formulated as

$$J_n = \sum_{i=1}^{M-1} F_n(Q_{ti}) \tag{8}$$

$$F_n(Q_{ti}) = \begin{cases} (C_i - C_{max})^2, C_i > C_{max}, \\ 0, C_i \leq C_{max} \end{cases} \tag{9}$$

The optimization problem is solved by a non-linear optimization solver NLopt. 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 metrics such as collision rate, completion 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 dataset used is the large-scale outdoor scenarios dataset SemanticKITTI [28]. The model is trained for 80 epochs on a single NVIDIA 3090 with a batch size of 12. During the training procedure, the input point cloud is augmented by randomly flipping along the x-y axis. We employ the Adam optimizer [29] with an initial learning rate of 0.001 to train the LBSCNet end-to-end. Finally, the pre-trained model with the best completion accuracy is deployed offline to predict the occlusion area.

Simulation. Simulation experiments were conducted on a Lenovo 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 20×20 room and a 3×30 corridor with random obstacles, creating occluded spaces and unknown areas. The AGR's task was to navigate from a starting point to a designated destination.

Indoor and Outdoor. We deployed HE-Nav on a custom AGR platform (in Fig. 5) for real-world indoor and outdoor environment experiments. This platform utilizes the Prometheus software [30] 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.

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. Specifically, we also focus on LBSCNet's inference speed to ensure it meets the real-time requirements for autonomous navigation. Regarding navigation, we pay attention to performance metrics such as planning success rate (%) and completion time (t), as well as energy consumption (W) for ground, aerial, and overall operations.

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[11], (2) LiDAR-based SSC methods including LMSCNet [22], and SSCNet [31] and SCPNet[23]. To evaluate

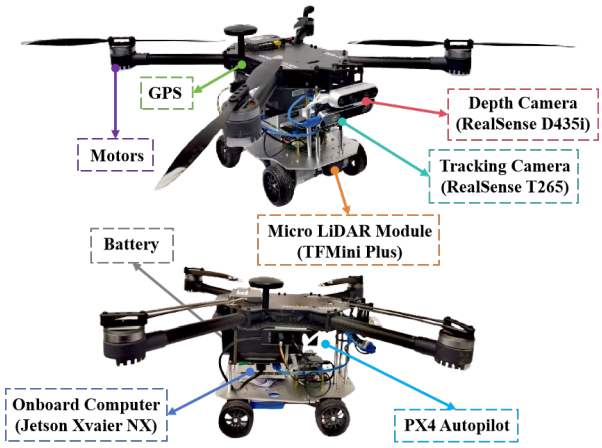


Fig. 5: The detailed composition of the robot platform.

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

Method	*IoU*	*mIoU*	*Prec.*	*Recall*	*FPS*
SSCNet [31]	53.20	14.55	59.13	**84.15**	12.00
LMSCNet [22]	55.32	17.01	77.11	66.19	13.50
LMSCNet-SS [22]	56.72	17.62	**81.55**	65.07	13.50
S3CNet [21]	45.60	29.50	48.79	77.13	1.20
Monoscene [12]	38.55	12.22	51.96	59.91	< 1
VoxFromer-T [11]	57.69	18.42	69.95	76.70	< 1
VoxFromer-S [11]	57.54	16.48	70.85	75.39	< 1
SCPNet [23]	56.10	**36.70**	72.43	78.61	< 1
LBSCNet (Ours)	**59.71**	23.58	77.60	71.29	**20.08**

the performance and energy-efficient of HE-Nav, we compared HE-Nav with TABV [1], *Fan et al.* [2] and EGO-Planner* [13].

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

Quantitative Results. We compare our proposed LBSCNet with the state-of-the-art SemanticKITTI test set by submitting the results to the official test server. As shown in Table I, our LBSCNet achieves the best performance in the completion metric IoU (59.71%) and ranks third in terms of the semantic segmentation metric mIoU (23.58%). This can be attributed to our novel semantic and completion decoupling network structure, which utilizes contextual semantic information to help the network better understand the scene structure and promote completion. Moreover, our LBSCNet has low latency and runs in real-time (20.08 FPS). It is worth noting that LBSCNet outperforms SCPNet, another point cloud-based method, by 6.45% in IoU and runs approximately 20 times faster. This improvement stems from the use of sparse 3D convolutions and lightweight BEV feature fusion within our network. As a result, LBSCNet can meet the real-time requirements for subsequent path planning.

Qualitative Results. We present visualizations on the SemanticKITTI validation set, as shown in Fig. 5. Additionally,

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.

we visualize the results of LMSCNet [22], Monoscene[12], VoxFormer[11], and SCPNet [23] for comparison purposes. From Fig. 6, it is evident that our LBSCNet predicts more accurate SSC results, particularly for "wall" classes and large objects such as cars, which is consistent with the findings in Table I. Crucially, the occlusion areas we focus on, such as vegetation and trees behind the wall, are also accurately completed, which is essential for subsequent path planning.

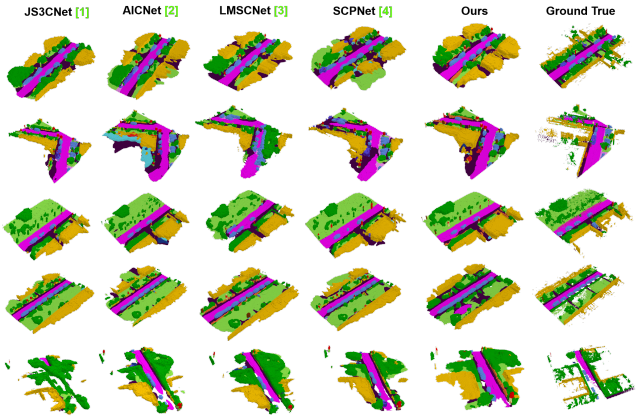


Fig. 6: Qualitative results of our method and others. LBSCNet better captures the scene layout in large-scale scenarios.

Ablation Study. Ablation studies on the SemanticKITTI validation set (Table 4) highlight the significance of two key components in our network: self-attention mechanisms and 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.

C. Simulated Air-Ground Robot Navigation

We conducted a comparative analysis of our HE-Nav navigation system against TABV [1], [2] and EGO-Planner* [13] in a square room and corridor scenario. 100 trials with varying obstacle placements, we recorded the travel time, length, energy consumption, and success rate (i.e., no collisions). EGO-Planner* indicates that the AGR is regarded as a drone and remains in flight during the navigation process.

Quantitative Results. As illustrated in Fig. 7, our state-of-the-art HE-Nav system achieves an outstanding planning success rate of 98% in both square rooms and corridor environments.

This superior performance can be ascribed to our innovative LBSCNet's capability to predict obstacle distributions in occluded areas, effectively allowing the planner to avoid these locations and substantially reducing the probability of collisions. Furthermore, our path planner seamlessly collaborates with the KinoDynamic A* algorithm to attain the lowest overall energy consumption. In comparison to the flight-centric EGO-planner*, our HE-Nav system diminishes energy consumption by an impressive 8X. Additionally, when juxtaposed with the cutting-edge TABV planning approach, HE-Nav realizes a 4X reduction in energy consumption. This advantage is especially evident in corridor scenarios, where our AGR strategy relies exclusively on ground movement, resulting in a remarkable 10X energy savings compared to the EGO-planner*. Regarding real-time performance, our HE-Nav system excels in two aspects: the perception module's reasoning is not only real-time but also benefits from a 7X acceleration in computation time, due to the elimination of redundant ESDF calculations. Although the EGO-planner* demonstrates the shortest overall movement time because of its reliance on flight, HE-Nav outshines its competitors in the AGR field, boasting a travel time of a mere 15 seconds.

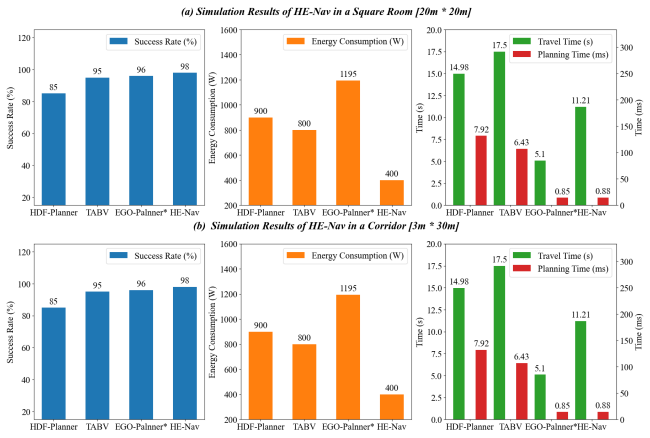


Fig. 7: Quantitative results of HE-Nav in two simulation scenarios.

Qualitative Results. As depicted in Figure 7, the path generated by the HDF-Planner fails to effectively consider both smoothness and dynamic feasibility. Additionally, the TABV path primarily focuses on the energy consumption associated with flight, which results 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. For a more comprehensive understanding of the qualitative results, please refer to the supplementary material provided.

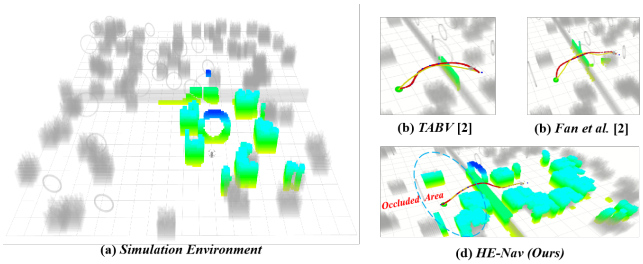


Fig. 8: Qualitative results of path planning and occlusion prediction in simulation environment.

D. Real-world Air-Ground Robot Navigation

On the customized AGR platform, we deployed HE-Nav on the NVIDIA Jetson Xavier NX airborne computer. The depth camera captured sparse point cloud data, which served as the input for LBSCNet. By utilizing TensorRT, we optimized the pre-trained model to ensure its compatibility with the airborne computer, maintaining real-time reasoning capabilities even on such resource-limited devices.

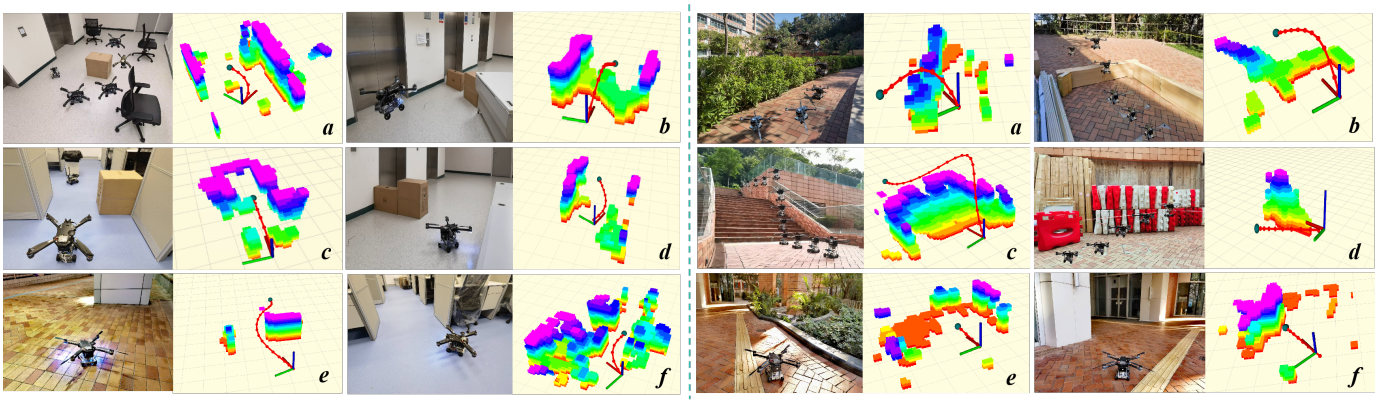
Figure 8 demonstrates the average energy consumption of our AGR system in three distinct states: driving, flying, and hovering. By monitoring the working time of the two modes in real indoor and outdoor environments, we can accurately assess more fine-grained energy consumption metrics. We conducted a comparison of energy consumption in real-world settings with TABV to further validate our system's performance.

We tested the performance and energy efficiency of HE-Nav in 6 indoor scenes and 6 outdoor scenes. The inference speed of LBSCNet and AG-Planner on different devices is shown in the figure. On the onboard computer, LBSCNet can still guarantee low latency, and after AG-Planner removes ESDF, the speed is increased by 5 times. As shown in Figure 11, whether it is an indoor environment or an outdoor environment, the planner can always generate smooth, safe, and dynamically feasible paths.

Taking (a) 6 and (b) 6 as examples, we record the planning time and energy consumption changes of HE-Nav and TABV during the navigation process. As shown in Figure 12, the planning time of HE-Nav is significantly reduced due to the removal of ESDF calculation, and the energy consumption is reduced by 4 times.More qualitative results can be found in the supplementary material.

VI. Conclusion

In conclusion, we have presented HE-Nav, a high-performance and energy-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



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

Fig. 9: 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.

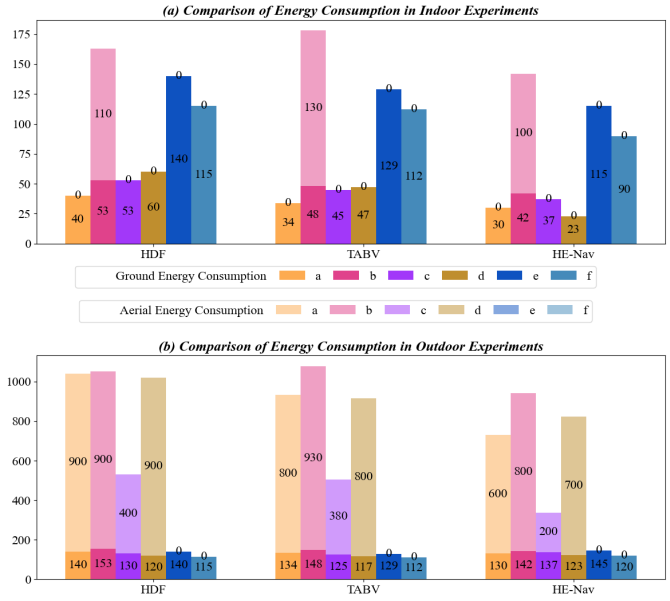


Fig. 10: Qualitative results of path planning and occlusion prediction in simulation environment.

and real experiments, HE-Nav has been demonstrated to significantly outperform recent planning frameworks, such as TABV, achieving 50% energy savings, a 98% success rate, and a 59.71 IoU. Our work lays a solid foundation for future research on AGR navigation systems, and we believe the release of our code and hardware configuration will contribute to further advancements in this field.

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] 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.

[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] 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.

[14] 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 recognition*, pages 9224–9232, 2018.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] Haotian Dong, Enhui Ma, Lubo Wang, Miaohui Wang, Wuyuan Xie, Qing Guo, Ping Li, Lingyu Liang, Kairui Yang, and Di Lin. Cvsformer: Cross-view synthesis transformer for semantic scene completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8874–8883, 2023.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

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

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

[31] 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.

VIII. APPENDIX

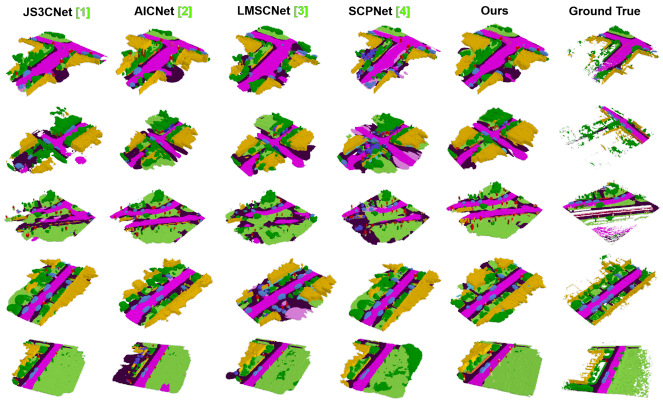


Fig. 11: Four methods were used to plan paths in a simulated square room. AGRNav demonstrates the ability to predict the distribution of obstacles in occluded areas.

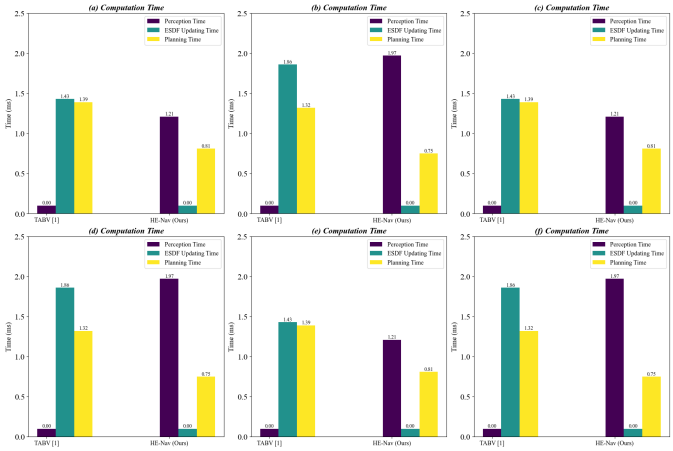


Fig. 12: Qualitative results of path planning and occlusion prediction in simulation environment.