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

Junming Wang[1], Zekai Sun[1], Xiuxian Guan[1], Tianxiang Shen[1], Dong Huang[1], Zongyuan Zhang[1], Tianyang Duan[1], Fangming Liu[3] and Heming Cui[1,2,*]

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

In this paper, we present HE-Nav, the first high-performance and efficient navigation system tailored for AGRs. The perception module utilizes a lightweight semantic scene completion network (LBSCNet), guided by a bird's eye view (BEV) feature fusion and enhanced by an exquisitely designed SCB-Fusion module and attention mechanism. This enables real-time and efficient obstacle prediction in occluded areas, generating a complete local map. Building upon this completed map, our novel AG-Planner employs the energy-efficient kinodynamic A* search algorithm to guarantee planning is energy-saving. Subsequent trajectory optimization and post-refinement processes yield safe, smooth, dynamically feasible and ESDF-free aerial-ground hybrid paths. Extensive simulations and real-world experiments demonstrate HE-Nav achieved 7x energy savings in real-world situations while maintaining planning success rates of 98% in simulation scenarios. The code and video can be found on our project page: https://jmwang0117.github.io/HE-Nav/.

***Index Terms*—Motion and Path Planning, Perception and Autonomy, Robotics and Automation in Construction**

I. Introduction

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

Existing AGRs navigation system [2], [1], [4] utilize sensors (e.g., cameras) to perceive surrounding environments and establish Euclidean Signed Distance Field (ESDF) maps

Manuscript received: April 22, 2024; This paper was recommended for publication by Editor Pauline Pounds upon evaluation of the Associate Editor and Reviewers' comments.

[1]The University of Hong Kong, Hong Kong SAR, China. [2]Shanghai AI Laboratory. [3]Huazhong University of Science and Technology, Wuhan, China. *denotes corresponding author.

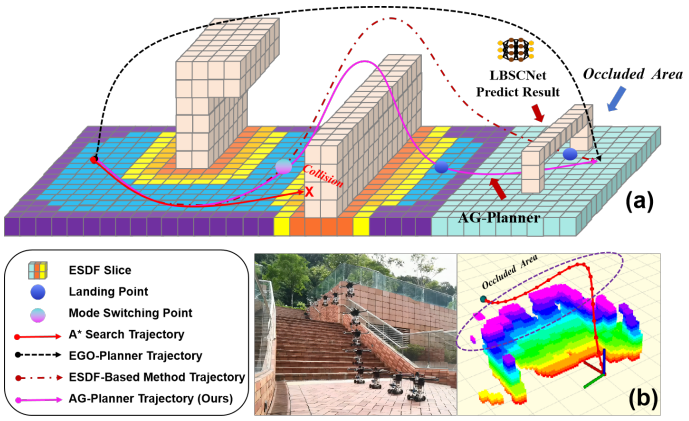

Fig. 1. (a) Current navigation systems underperform in occluded areas due to inaccurate obstacle prediction and the computationally intensive process of creating ESDF maps. (b) Our HE-Nav system can generate energy-saving, collision-free and ESDF-free aerial-ground paths in real-time with the help of the LBSCNet model and AG-Planner.

(in Fig. 1a), subsequently the path planner to search for collision-free trajectories. Unfortunately, while these ESDF-based navigation systems have proven successful in structured indoor scenarios, they face two limitations when navigating in occluded environments (e.g., large buildings or forests).

Firstly, the *perception module* results in incomplete local maps (i.e., containing occlusion-induced unknown areas) since the narrow field of view in sensor-based mapping. This not only generates paths with high collision risk (e.g., *red path* in Fig.1a.) but also prolongs moving time since redundant paths. While emerging semantic scene completion (SSC) networks [8], [9], [10] show promise in predicting occluded obstacles accurately, they face a trade-off between completion accuracy and inference time. Some employ 3D convolution [11] for better accuracy, yet are impractical for resource-constrained robots. Conversely, lightweight designs [12], [3] offer speed but compromise on precision.

Secondly, the existing ESDF-based AGRs *path planners* [2], [3] are inefficient since building the ESDF map generates redundant calculation times that do not meet the real-time requirements. Furthermore, while *Zhou et al.* [13] devised an ESDF-free path planner for quadcopters, it fails to address AGR-specific requirements, particularly energy efficiency and dynamic constraints. Their flight-centric trajectory (e.g., *black path* in Fig.1a.) generation results in elevated energy consumption and the inherent non-holonomic constraints of AGRs make it impossible to naively migrate and use such planners (in Table I). Notably, the inefficiency of AGR path planners not only stems from the above intrinsic flaws but also the

TABLE I
COMPARED WITH THREE BASELINE AGR NAVIGATION SYSTEMS AND EGO-PLANNER DESIGNED SPECIFICALLY FOR MULTICOPTERS.

Method	Suitable to AGRs	Category	Occlusion Awareness	Performance Metric		Efficiency Metric	
				Moving Time	Success Rate	Planning Time	Energy Consumption
HDF [1]	✓	Only A*	✗	✗	✗	✗	✗
TABV [2]	✓	ESDF-based	✗	✗	✗	✗	✗
AGRNav[3]	✓	ESDF-based	✓	✗	✗	✗	✗
EGO-Planner [13]	✗	ESDF-free	✗	✓	✓	✓	✗
HE-Nav (Ours)	✓	ESDF-free	✓	✓	✓	✓	✓

perception module's limitations in delivering complete local maps. Incomplete maps lead to either overly conservative or aggressive mode switching (e.g., the landing point location in Fig.1a.), adversely affecting the energy efficiency of AGRs.

Our key insight for addressing the limitations of the *perception module* lies in decoupling the conventional network architecture [12], [3] that jointly learns geometry and semantics into distinct branches. This enables each branch to focus on acquiring domain-specific features, thereby enhancing the overall performance. Concurrently, drawing inspiration from [14], [15], we transition the feature fusion process to the Bird's Eye View (BEV) space, which holds the potential to diminish computational complexity and ensure high-speed inference. Regarding the *path planner* design, our primary objective is to ensure energy-efficient and real-time planning results (i.e., meeting the efficiency metrics in Table I). To achieve these, the path planner must accommodate the non-holonomic constraints inherent to AGRs and remove redundant ESDF calculations. Additionally, incorporating energy costs associated with different modes (e.g., flying and driving) is imperative for facilitating judicious mode switching and promoting energy conservation.

Based on these above insights, we present ***HE-Nav***, the first *high-performance* and *efficient* navigation system tailored for AGRs, as illustrated in Fig. 2. The system comprises two pivotal components, with the first being a lightweight BEV-guided semantic scene completion network (LBSCNet) deployed on the AGR. By processing sparse point cloud inputs, LBSCNet performs fast inference to accurately predict obstacle distribution (i.e., voxel occupancy) and semantics. These predictions are then integrated into local maps for path planning, facilitated by the query-based low-latency map update method presented in [3], ensuring timely updates.

During the planning phase, we develop AG-Planner that searches for aerial-ground hybrid paths. Specifically, an energy-efficient kinodynamic A* path searching front-end utilizes motion primitives instead of straight lines as graph edges, by adding additional energy costs for aerial destinations, the planner not only tends to search ground trajectories but also switches to aerial mode only when AGRs encounter huge obstacles, thereby promoting energy-saving. We then utilize an obstacle distance estimation method from [13] to circumvent obstacles, avoiding ESDF computations. Finally, a gradient-based B-spline optimizer refines paths to generate a safe, smooth, and dynamically feasible trajectory.

We evaluated LBSCNet on the SemanticKITTI benchmark and compared its performance to some leading SSC networks. Then, we tested HE-Nav against two AGR navigation baselines (i.e., TABV [2] and AGRNav [3]) in simulated and real environments, demonstrating its improved performance and efficiency (Table I). Our evaluation reveals:

- **HE-Nav is high-performance.** HE-Nav achieved success rates of 98% in the two simulation scenarios, while having the shortest average movement time. (§ V-C)
- **HE-Nav is efficient.** HE-Nav achieves **7x** energy savings in real-world tests, while reducing planning time by **6x** relative to ESDF-based baselines. (§ V-D)
- **LBSCNet is accurate and high-speed inference.** LBSCNet achieves state-of-the-art performance (IoU = 59.71) on the SemanticKITTI benchmark and enables high-speed inference (i.e., 20.08 FPS). (§ V-B)

Our main contributions are the creation of the lightweight LBSCNet and the energy-efficient AG-Planner. **(1)** LBSCNet featuring a novel BEV fusion branch and SCB-Fusion module for fast inference and complete local map generation. **(2)** Leveraging this map, AG-Planner achieves energy-efficient, ESDF-free path planning.

To the best of our knowledge, HE-Nav is the first AGR-tailored navigation system, combining occlusion awareness network and ESDF-free aerial-ground hybrid path planning, ensuring high-performance and efficient autonomous navigation in occluded environments.

II. RELATED WORK

A. Motion Planning for AGRs

Numerous researchers have explored various aerial-ground robot configurations, such as incorporating passive wheels [2], [16], [17], [18], [4], [3], or multi-limb [19] onto drones. In contrast, others [7], [6], [5] have integrated rotors with wheeled robots to achieve dual-mode (i.e., flying and driving) locomotion. These designs facilitate enhanced stability and control in both locomotion modes. Consequently, we also adopted this mechanical structure to customize further our AGR, which has four wheels and four rotors. Moreover, Existing research primarily focuses on innovative mechanical structure designs, and the area of AGR autonomous navigation remains underexplored. Recently, [1] tackled ground-aerial motion planning, utilizing the A* algorithm for geometric path guidance and favouring ground paths by adding extra

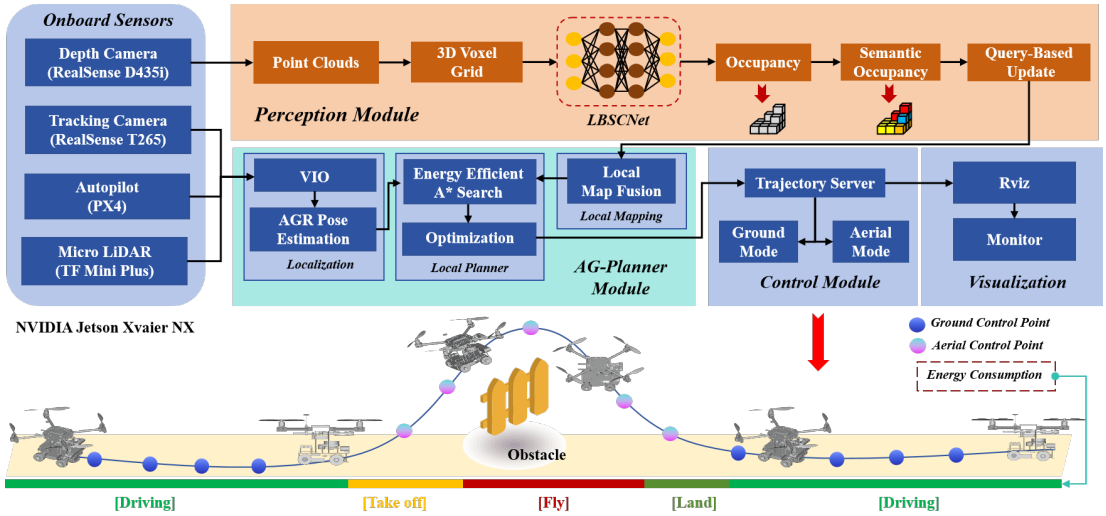

Fig. 2. HE-Nav system architecture. The perception module and path planner run asynchronously on the onboard computer, connected through a query-based map update method [3] to ensure real-time local map updates with prediction results.

energy costs to aerial paths. However, this approach is limited by its lack of dynamic models and post-refinement in local planner trajectories, potentially compromising smoothness and dynamic feasibility. [2] introduced an efficient and adaptive path planner and controller, but its reliance on an ESDF map results in intensive computation and limited perception of occluded areas, consequently leading to a low success rate in path planning and increased energy consumption.

B. Occlusion-Aware for AGRs

AGR's sensor-based perception method cannot make the local map include the distribution of obstacles in the occluded area, which will cause the planned path to be sub-optimal. In recent years, the field of semantic scene completion [8], [11]has witnessed significant advancements, particularly in addressing the challenges posed by the obstruction of obstacles in complex and unknown environments. These advancements have led to the development of various point-cloud-based and camera-based methods. In the realm of camera-based methods, *Cao et al.* [9] introduced MonoScene, a groundbreaking approach that infers scene structure and semantics from a single monocular RGB image. On the other hand, point-cloud-based methods have also made significant strides.[12] 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. Despite substantial progress in camera and point-cloud-based SSC methods, their high computational demands limit their suitability for resource-constrained AGR platforms.

III. Perception Module of HE-Nav

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

A. LBSCNet Network Structure

LBSCNet decoupling the learning process of semantics and completion, 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*:** This branch consists of a voxelization layer and three encoder blocks sharing a similar architecture, each encoder block comprises a residual block [20] with sparse 3D convolutions and a cross-scale global attention (CSGA) module from [21]. The integration of the CSGA module not only aligns multi-scale features with global voxel-encoded attention to capturing the long-range relationship of context but also alleviates the computational burden by reducing feature resolution. Specifically, in the voxelization layer, point clouds $P \in \mathbb{R}^{N\times 3}$ are partitioned based on the voxel resolution s and mapped into voxel space. Subsequently, an aggregation function (i.e., max function) is applied to the point cloud within each voxel, yielding a single feature vector. A multi-layer perceptron (MLP) reduces the dimensionality of this feature vector, producing the final voxel features V_{f_m} with a spatial resolution of $L\times W\times H$, f_m represents the index of the voxel. The voxel features V_{f_m} are then input into three encoder blocks to obtain semantic features $\{Sem_f^1, Sem_f^2, Sem_f^3\}$ (Fig. 3). The semantic branch is optimized using lovasz loss [22] and cross-entropy loss [23]. The semantic loss L_{sem} is the sum of the loss at each stage, expressed as follows:

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

***Completion Branch*:** The input to the completion branch is voxels $V \in \mathbb{R}^{1\times L\times W\times H}$ generated by point clouds.

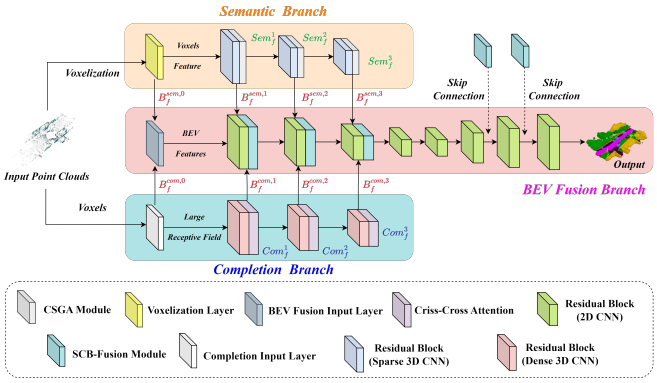

Fig. 3. The overview of the proposed LBSCNet. It consists of semantic, completion and BEV fusion branches.

The output is the multi-scale dense completion features $\{Com_f^1, Com_f^2, Com_f^3\}$, providing more intricate geometric information. As depicted in Fig. 3, the completion branch comprises an input layer (kernel size $7\times7\times7$), three residual blocks and three GPU memory-efficient criss-cross attention (CCA) [24] modules. The residual blocks incorporate dense 3D convolutions with a kernel size of $3\times3\times3$, capturing local geometric features. Conversely, the criss-cross attention (CCA) [24] module is designed to capture long-range dependencies by gathering contextual information in both horizontal and vertical directions, thereby enriching the completion features with a global context. The training loss L_{com} for this branch is calculated as follows:

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

where i denotes the $i-th$ stage of the completion branch and L_{binary_cross} indicates the binary cross-entropy loss. Notably, during training, both the semantic and completion branches undergo deep supervision [25]. Lightweight MLPs are attached as auxiliary heads [21] after each encoder block to obtain semantic and geometric predictions for valid voxels. However, during inference, these auxiliary heads are removed to maintain a lightweight network structure.

***BEV Feature Fusion Branch*:** Previous research on SSC tasks has relied on fusing dense 3D features, resulting in considerable computational overhead and hindering deployment on resource-constrained AGR devices. We propose a lightweight BEV fusion branch specifically designed for SSC tasks, capitalizing on recent advancements in BEV perception [14], [15], [26]. By projecting learned semantic and geometric features into BEV space and incorporating the innovative SCB-Fusion module, we significantly reduce computational demands while maintaining rapid inference capabilities. Specifically, our BEV fusion network employs a U-Net architecture with 2D convolutions, featuring an input layer and four residual blocks in the encoder (Fig. 3). The process of projecting semantic and geometric features to BEV space is as follows:

***Semantic Feature Projection*:** To project three-dimensional semantic features $\{Sem_f^1, Sem_f^2, Sem_f^3\}$ into the two-dimensional BEV space, we first generate a BEV index based on the voxel index f_m and then the features sharing identical BEV indices are aggregated using an aggregation function (e.g., the max function) to yield sparse BEV features. Utilizing the feature densification function offered by spconv [27], we generate dense BEV features $\{B_f^{sem,0}, B_f^{sem,1}, B_f^{sem,2}, B_f^{sem,3}\}$ based on the BEV index and sparse BEV features.

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

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

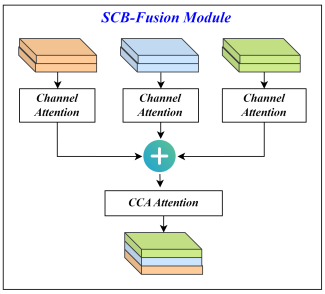

Fig. 4. SCB-Fusion Module and Qualitative results in the simulation environment.

***Feature Fusion after Projection*:** To fuse the projected features, we devise an SCB-Fusion module (Fig. 4) that fuses current semantic features, geometric features, and BEV features from the previous layer. Specifically, we first compute channel attention for features $B_{pre}/B_{com}/B_{sem}$ to adaptively weight the feature channels. The weighted features are then summed and passed through a 1×1 convolution and CCA attention to obtain the fused features F_{SCB}. The fused features can be expressed as:

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

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

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

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

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

IV. Aerial-Ground Motion Planner of HE-Nav

In this section, we introduce the novel AG-Planner. It is built on EGO-Planner [13] 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.

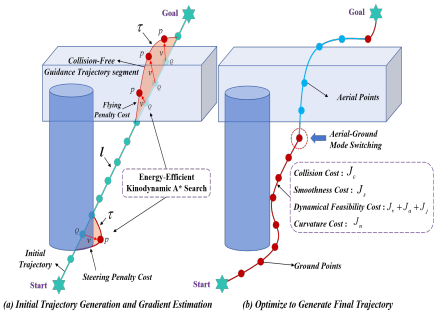

Fig. 5. Illustration of AG-Planner and topological trajectory generation.

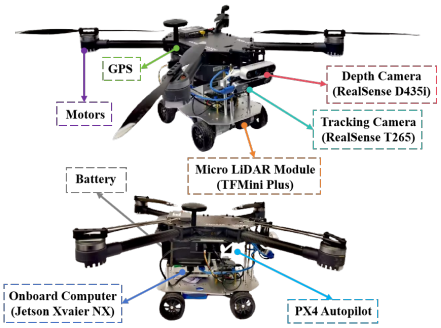

Fig. 6. The detailed composition of our robot platform.

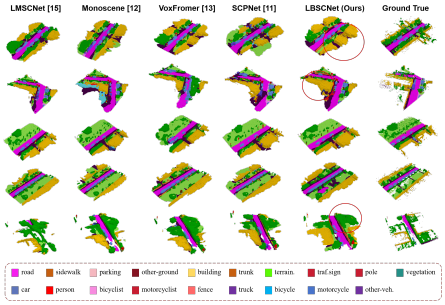

Fig. 7. Qualitative results of our LBSCNet and others.

A. Energy-Efficient Kinodynamic Hybrid A* Path Searching

Our AG-Planner first creates a naive "*initial trajectory*" ι (in Fig. 5a) that overlooks obstacles by randomly adding coordinate points, considering the positions of both the starting and target points. Following that, for the "*collision trajectory segment*" (i.e., the trajectory inside the obstacle), the back end of our planner based on [28] to propose an energy-efficient kinodynamic A* path search algorithm to establish a safe "*guidance trajectory segment*" τ, which uses motion primitives instead of straight lines as graph edges in the searching loop. In this algorithm, we add extra flying and ground-steering energy consumption for the motion primitives. Consequently, the path searching not only tends to plan ground trajectories but also switches to aerial mode and flies over them only when AGRs encounter huge obstacles, thereby promoting energy-saving.

B. Gradient-Based B-spline Trajectory Optimization

***B-spline Trajectory Formulation*:** In trajectory optimization (in Fig. 5b), 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:

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

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

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

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

***Collision Avoidance Force Estimation*:** For each control point on the collision trajectory segment, vector v (i.e., a safe direction pointing from inside to outside of that obstacle) is generated from ι to τ and p is defined at the obstacle surface (in Fig. 5a). With generated $\{p, v\}$ pairs, the planner maximizes D_{ij} and returns an optimized trajectory. The obstacle distance D_{ij} if i^{th} control point Q_i to j^{th} obstacle is defined as:

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

Because the guide path τ is energy-saving, the generated path is also energy efficient (in Fig. 5a).

***B-spline Trajectory Optimization and Post-refinement Procedure*:** The basic requirements of the B-spline paths are threefold: *smoothness*, *safety*, and *dynamical feasibility*. Based on the special properties of AGR bimodal, we first adopt the following cost terms designed by *Zhou et al.* [30]:

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

where J_s is the smoothness penalty, J_c is for collision, and J_v, J_a, J_j are dynamical feasibility costs that limit velocity, acceleration and jerk. $\lambda_s, \lambda_c, \lambda_f$ are weights for each cost terms. Detailed explanations can be found in [13]. 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:

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

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

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

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

The optimization problem is solved using the non-linear optimization solver NLopt [31], with post-refinement from [13] for constraint violations. After path planning, a setpoint

from the trajectory is selected and sent to the controller. Aerial setpoints include yaw angle and 3D position, velocity, and acceleration, while ground ones include yaw angle and 2D position and velocity. In addition, when the z-axis coordinate of the next control point is greater than the ground threshold, that is, when mode switching is required, an additional trigger signal will be sent to the controller (i.e., PX4 Autopilot). The controller will automatically switch to the flight state.

V. Evaluation

In this section, we first assess the LBSCNet on the SemanticKITTI benchmark. Subsequently, we selected the model with the best completion accuracy on SemanticKITTI and integrating this model with AG-Planner, a comprehensive HE-Nav system is formed. We then evaluate the AGR's autonomous navigation capability using HE-Nav in both simulated and real-world settings, focusing on ***performance*** metrics (i.e., planning success rate, average movement time) and ***efficiency*** metrics (i.e., average planning time, energy consumption).

TABLE II
Battery and Energy Consumption Parameters

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

A. Evaluation setup

Perception Module: We trained and tested LBSCNet using the outdoor SemanticKITTI dataset [32] on a single NVIDIA 3090 GPU. The model was trained for 80 epochs with a batch size of 12, using the Adam optimizer [33] at an initial learning rate of 0.001, and input point cloud augmentation by random flipping along the $x-y$ axis. Finally, we deployed the pre-trained model offline with the best completion accuracy to complete the local map.

Simulation Experiment: The simulation scenarios comprised a $20m \times 20m \times 5m$ square room and a $3m \times 30m \times 5m$ corridor with numerous random obstacles, creating occluded spaces (Fig. 9). The AGR's task was to navigate from a starting point to a designated destination without collision.

Real-world Experiment: We employed HE-Nav on a custom AGR platform (Fig. 6) for indoor and outdoor experiments, using Prometheus software [34] with a RealSense D435i depth camera, a T265 tracking camera, and a Jetson Xavier NX computer. We assessed the average energy consumption per second for AGR during driving and flying (Table II) to establish a basis for evaluating energy usage in real and simulated tests.

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 III 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. In contrast, LBSCNet outperforms SCPNet by 6.43% in IoU and runs approximately ***20 times*** faster.

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

Method	*IoU*	*mIoU*	*Prec.*	*Recall*	*FPS*
SSCNet [35]	53.20	14.55	59.13	**84.15**	12.00
LMSCNet [12]	55.32	17.01	77.11	66.19	13.50
LMSCNet-SS [12]	56.72	17.62	81.55	65.07	13.50
SCONet [3]	56.12	17.61	**85.02**	63.47	18.50
S3CNet [36]	45.60	29.50	48.79	77.13	1.20
Monoscene [9]	38.55	12.22	51.96	59.91	< 1
VoxFromer-T [10]	57.69	18.42	69.95	76.70	< 1
VoxFromer-S [10]	57.54	16.48	70.85	75.39	< 1
SCPNet [8]	56.10	**36.70**	72.43	78.61	< 1
LBSCNet (Ours)	**59.71**	23.58	77.60	71.29	**20.08**

Qualitative Results: We provide visualization results on the SemanticKITTI validation set. As illustrated in Fig. 7, our LBSCNet demonstrates superior SSC predictions, particularly for "wall" classes and larger objects like cars, aligning with the results in Table III. Importantly, the occlusion areas we target, such as vegetation and trees behind walls, are accurately completed, proving vital for subsequent path-planning.

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

TABLE IV
Ablation study on the SemanticKITTI validation set.

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

C. Simulated Air-Ground Robot Navigation

In a square room and corridor scenario (Fig 9), through 100 trials with varied obstacle placements, we evaluated the

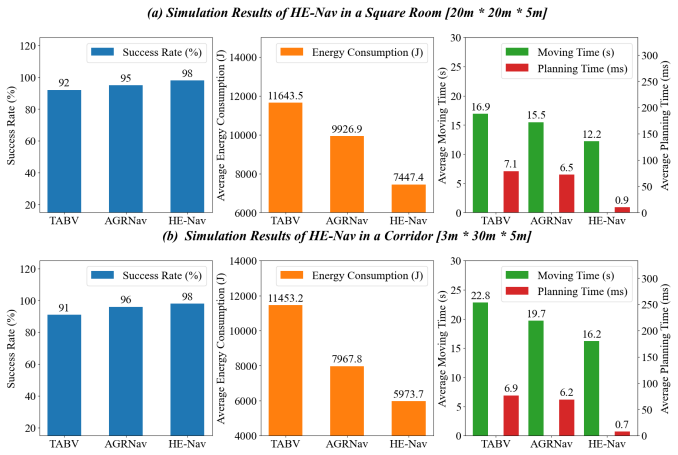

Fig. 8. Quantitative results of HE-Nav in two simulation scenarios.

average moving time, planning time, and success rate (i.e., collision-free) of each method (Fig. 8). To calculate the energy consumption in the simulation environment, we use the data in Table II and record the flight and driving time in the simulation to complete the calculation.

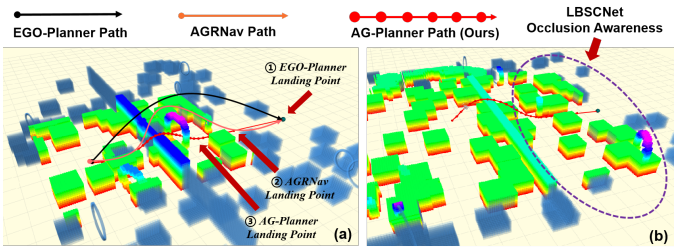

Fig. 9. Qualitative results of path planning and occlusion prediction in simulation environment.

Fig. 8 showcases the exceptional performance of our HE-Nav system, achieving 98% success rates in square rooms and corridors, with average movement times of 12.2s and 16.2s. Our system significantly accelerates planning time, being 6 times faster than TABV [2] and AGRNav [3], due to eliminating redundant ESDF calculations. Addressing the limitations of TABV, which lacks obstacle sensing in occluded areas, and AGRNav, with lower obstacle prediction accuracy, HE-Nav effectively employs the ESDF-free AG-Planner. Leveraging LBSCNet's precise obstacle prediction (Fig. 9b) and integrating the energy-efficient kinodynamic A* algorithm, our system achieves the lowest average energy consumption of 7447.4 J and 5973.7 J. AG-Planner also achieves the mode-switching balance between radical and conservative (e.g., optimal landing position in Fig. 9a), further enhancing energy savings.

D. Real-world Air-Ground Robot Navigation

We assess HE-Nav's performance and energy efficiency across 4 indoor and 4 outdoor scenarios (Fig. 10). In indoor settings, our HE-Nav consistently demonstrates lower average energy consumption than AGRNav and EGO-Planner. For instance, in scenarios a and b, our system achieves energy

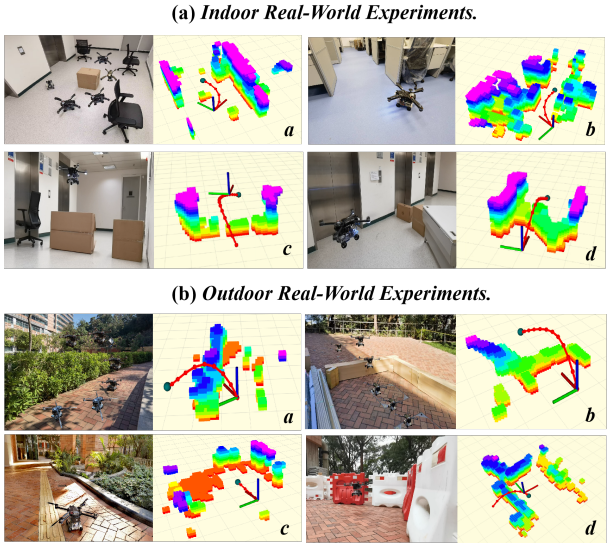

Fig. 10. HE-Nav effectively predicts obstacle distribution in occluded areas and plans collision-free hybrid trajectories.

consumption reductions of 69.3% and 76.8% relative to EGO-Planner, primarily due to the inclusion of additional penalty terms in the aerial segment, prompting our AG-Planner to favour energy-saving ground paths. Simultaneously, LBSCNet rapidly predicts obstacle distribution in occluded areas, generating a more comprehensive local map (e.g., b visualization results) to serve as the basis for AG-Planner's path search. This high-precision completion aids the planner in identifying optimal landing points, further contributing to energy conservation.

Fig. 11. Quantitative results of indoor and outdoor real environmental energy consumption.

Transitioning to outdoor scenarios, HE-Nav surpasses AGRNav with a 13.29% reduction in average energy consumption in scenario d, This can be attributed not only to the optimization of smooth aerial paths, which minimizes flight energy consumption, but also to LBSCNet's ability to accurately predict obstacle distribution in occluded environments (i.e., a, c visualization results). This proficiency aids in reducing redundant paths and identifying optimal mode-switching points (e.g., the

landing point in scenario d), as early landing and transition to ground driving mode promote energy conservation.

VI. Conclusion

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

VII. Acknowledgments

We thank all anonymous reviewers for their helpful comments.

References

[1] D. D. Fan, R. Thakker, T. Bartlett, M. B. Miled, L. Kim, E. Theodorou, and A.-a. Agha-mohammadi, "Autonomous hybrid ground/aerial mobility in unknown environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3070–3077.

[2] R. Zhang, Y. Wu, L. Zhang, C. Xu, and F. Gao, "Autonomous and adaptive navigation for terrestrial-aerial bimodal vehicles," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3008–3015, 2022.

[3] J. Wang, Z. Sun, X. Guan, T. Shen, Z. Zhang, T. Duan, D. Huang, S. Zhao, and H. Cui, "Agrnav: Efficient and energy-saving autonomous navigation for air-ground robots in occlusion-prone environments," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[4] R. Zhang, J. Lin, Y. Wu, Y. Gao, C. Wang, C. Xu, Y. Cao, and F. 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)*, 2023, pp. 1070–1077.

[5] X. Zhang, Y. Huang, K. Huang, X. Wang, D. Jin, H. Liu, and J. Li, "A multi-modal deformable land-air robot for complex environments," *arXiv preprint arXiv:2210.16875*, 2022.

[6] X. Zhang, Y. Huang, K. Huang, Z. Zhao, J. Li, H. Liu, and J. Li, "Coupled modeling and fusion control for a multi-modal deformable land-air robot," *arXiv preprint arXiv:2211.04185*, 2022.

[7] Q. Tan, X. Zhang, H. Liu, S. Jiao, M. Zhou, and J. Li, "Multi-modal dynamics analysis and control for amphibious fly-drive vehicle," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 621–632, 2021.

[8] Z. Xia, Y. Liu, X. Li, X. Zhu, Y. Ma, Y. Li, Y. Hou, and Y. Qiao, "Scpnet: Semantic scene completion on point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 642–17 651.

[9] A.-Q. Cao and R. de Charette, "Monoscene: Monocular 3d semantic scene completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3991–4001.

[10] Y. Li, Z. Yu, C. Choy, C. Xiao, J. M. Alvarez, S. Fidler, C. Feng, and A. 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*, 2023, pp. 9087–9098.

[11] S. Zuo, W. Zheng, Y. Huang, J. Zhou, and J. Lu, "Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction," *arXiv preprint arXiv:2308.16896*, 2023.

[12] L. Roldao, R. de Charette, and A. Verroust-Blondet, "Lmscnet: Lightweight multiscale 3d semantic completion," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 111–119.

[13] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.

[14] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[15] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu *et al.*, "Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 830–17 839.

[16] T. Wu, Y. Zhu, L. Zhang, J. Yang, and Y. Ding, "Unified terrestrial/aerial motion planning for hytaqs via nmpc," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1085–1092, 2023.

[17] N. Pan, J. Jiang, R. Zhang, C. Xu, and F. Gao, "Skywalker: A compact and agile air-ground omnidirectional vehicle," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2534–2541, 2023.

[18] Y. Qin, Y. Li, X. Wei, and F. Zhang, "Hybrid aerial-ground locomotion with a single passive wheel," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1371–1376.

[19] M. Martynov, Z. Darush, A. Fedoseev, and D. Tsetserukou, "Morphogear: An uav with multi-limb morphogenetic gear for rough-terrain locomotion," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2023, pp. 11–16.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] S. Xu, R. Wan, M. Ye, X. Zou, and T. Cao, "Sparse cross-scale attention network for efficient lidar panoptic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2920–2928.

[22] M. Berman, A. R. Triki, and M. 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*, 2018, pp. 4413–4421.

[23] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[24] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 603–612.

[25] Y. Liu and M. S. Lew, "Learning relaxed deep supervision for better edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 231–240.

[26] H. Liu, Y. Teng, T. Lu, H. Wang, and L. Wang, "Sparsebev: High-performance sparse 3d object detection from multi-camera videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 580–18 590.

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

[28] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

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

[30] X. Zhou, J. Zhu, H. Zhou, C. Xu, and F. Gao, "Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 4101–4107.

[31] S. G. Johnson *et al.*, "The nlopt nonlinear-optimization package," 2014.

[32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[35] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1746–1754.

[36] R. Cheng, C. Agia, Y. Ren, X. Li, and L. Bingbing, "S3cnet: A sparse semantic scene completion network for lidar point clouds," in *Conference on Robot Learning*. PMLR, 2021, pp. 2148–2161.