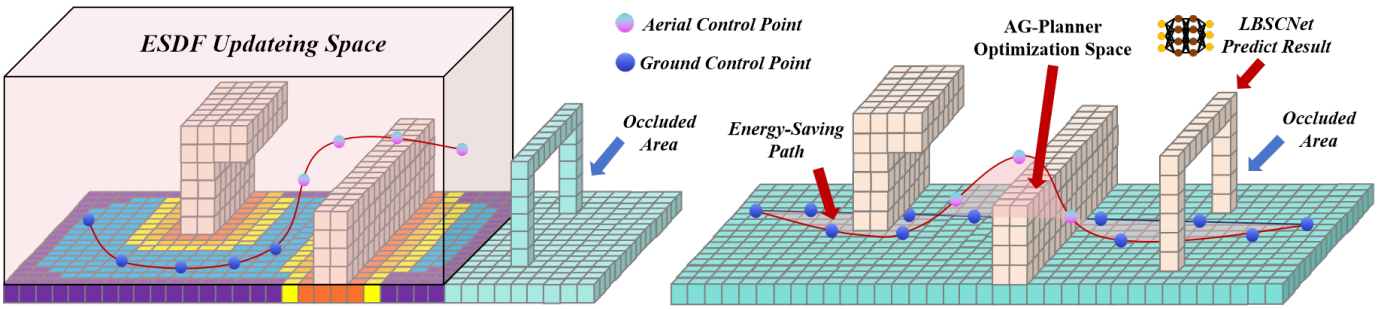
HE-Nav: A **H**igh-performance and **E**nergy-efficient **Nav**igation System for Aerial-Ground Robots

Anonymous Review. Paper-ID [123]



(a) ESDF-based Planning Framework: TABV [1]. **(b) Our HE-Nav for AGR.**

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.

***Abstract*—Aerial-ground robots (AGR) offer unique dual-mode capabilities (i.e., flying and driving), making them ideal for efficient hybrid mobility in uncertain and complex environments, such as disaster zones or wilderness areas. While recent AGR research has made progress in path planning for structured indoor settings, it encounters challenges in uncertain scenarios, including increased collision risks and energy consumption, due to suboptimal trajectory optimization, oversimplified mode-switching strategy, and a lack of occlusion awareness.**

In this paper, we present HE-Nav, the first high-performance and energy-efficient navigation system tailored for AGR. It integrates several innovative features, starting with a lightweight BEV-guided semantic scene completion network (LBSCNet) that predicts obstacle distributions in occluded areas and updates the local map in real-time. Utilizing this updated local map, our aerial-ground motion planner (AG-planner) first generates a collision trajectory that disregards obstacles. Next, it employs an energy-efficient kinodynamic A* algorithm to search the local collision-free guidance trajectory that corresponds to the trajectory segment within obstacles. Then, the planner models the collision cost and generates the estimated gradient between the above two types of trajectories, ultimately wrapping the collision trajectory segment out of the obstacles. This process significantly reduces computation time. Finally, a post-refinement procedure is applied to optimize the aerial-ground trajectory further, ensuring that the dynamic feasibility is maintained.

Our HE-Nav system demonstrates a 50% energy saving and higher performance (i.e., 98% success rate and 59.71 IoU) compared to two recent planning frameworks (e.g., TABV) through extensive simulations and real-world experiments. The code and hardware configuration will be released.

I. INTRODUCTION

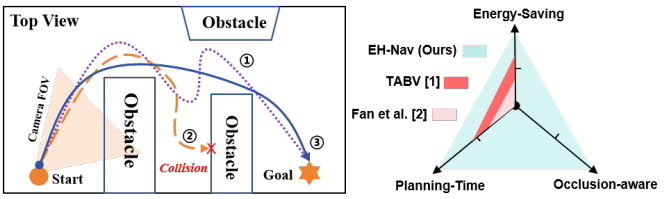
In recent years, aerial-ground robots (AGR) [1, 2, 3, 4] have emerged as a promising solution for a variety of applications, such as search and rescue in uncertain and complex environments like disaster zones or wilderness areas. Their high mobility and long endurance make them an ideal choice for performing hybrid locomotion tasks in these challenging settings. While researchers have made significant strides in developing AGRs with novel mechanical structures [5, 6, 7, 8, 9, 10] tailored to meet the demands of such tasks, it has become increasingly evident that innovative mechanical designs alone cannot fully harness the potential of these robots. The primary obstacle lies in their lack of autonomous navigation capabilities, which prevents them from effectively adapting to and navigating through complex environments.

To achieve autonomous navigation of AGRs, *zhang et al.* introduced TABV [1], a path planning framework that employs sensors (i.e., depth camera) to construct local Euclidean Signed Distance Field (ESDF) maps for trajectory planning. Meanwhile, TABV prioritizes ground paths in its search path unless it has to fly over extreme terrains, thereby saving energy. While this method has proven successful in structured indoor environments, it faces two challenges in uncertain, complex, and occluded environments (e.g., forests and large buildings), which can result in ***reduced path planning performance*** and ***increased energy consumption***.

The first challenge involves a suboptimal path planner. Existing planner with redundant ESDF map calculations and oversimplified mode-switching strategies. On the one hand, as shown in Fig. 1a. TABV builds an ESDF map for path planning, but obstacles account for only about 30% of the local map, while the remaining 70% of the free area is still involved in the calculation of ESDF. This redundancy results in

path planning time increases significantly. On the other hand, the TABV and *Fan et al.* [2] proposed an aerial-ground mode-switching strategy, which adds an aerial penalty term, and only considers energy consumption in the vertical direction. This strategy neglects the horizontal energy consumption caused by speed and steering changes, which also play a crucial role in overall energy efficiency.

The second challenge involves limited occlusion perception in uncertain environments. These environments typically contain numerous obstacles of varying shapes and sizes, while the AGR's sensors-based (e.g., LiDAR or Camera) perception capabilities are inherently limited to a specific field of view, making it impossible to see through obstructions, as shown in Fig. 2a. This limitation causes the local map constructed by the path planning algorithm to still contain occlusion and unknown areas. The existence of these areas makes path planning suboptimal and leads to a higher risk of collision, as shown in the *orange path* in Fig. 2a. Moreover, the lack of knowledge about the distribution behind obstructions may lead the robot into dead ends with no ground path, forcing it to backtrack or fly, as shown in the *purple path* in Fig. 2a. This results in additional energy consumption due to redundant paths. In conclusion, it is essential to develop a novel navigation system for AGRs that effectively addresses these challenges.



(a) Occluded areas lead to suboptimal paths. (b) Comparison with two baseline.

Fig. 2: Limited occlusion perception increases the risk of collision ② and additional energy consumption ①. HE-Nav's early perception capabilities enable the path planning to strike a balance between performance and energy consumption ③.

Our key observation to resolve these two challenges involves the development of an occlusion-aware module and a gradient-based, energy-efficient path planner to form a complete AGR navigation system. For the perception module, we draw inspiration from the recent rapid development of 3D semantic scene completion(SSC) networks, which are widely used in autonomous driving systems since they only require sparse point cloud input to obtain dense 3D scene completion and semantic prediction. Therefore, By designing the SSC network for the perception module and deploying the pre-trained model on AGR, the obstacle distribution and semantic information of the occluded area can be predicted through sparse point cloud perception. For the path planner, while *Zhou et al.* [11, 12] proposed an ESDF-free path planner tailored for quadcopters, it is not suitable for AGR due to some key limitations. These limitations include energy efficiency (i.e., AGR prefers ground paths to minimize energy consumption), 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). Consequently, we need to design an energy-saving path planner for AGR to ensure efficient and safe motion planning.

In light of our observations, we introduce **HE-Nav**, the first high-performance, energy-efficient autonomous navigation system for AGR with occlusion awareness, real-time computation, exceptional success rates, and minimal energy consumption, as shown in Fig. 2b. Our system is composed of two primary components. The first component is a Bird's Eye View (BEV)-guided lightweight Semantic Scene Completion (SSC) network, referred to as LBSCNet, which is implemented on the AGR for rapid inference. This network accurately predicts obstacle distribution and semantics by decoupling the learning process of semantics and geometry and utilizing semantic context to reconstruct scene geometry. To achieve efficient feature fusion and scene completion, the LBSCNet incorporates a BEV feature fusion module and a memory-efficient self-attention mechanism.

Based on the updated local map provided by the perception module, our proposed AGR motion planner (AG-planner) first generates an initial path (i.e., collision trajectory), which only considers the starting point and target point while ignoring obstacles. Subsequently, the AG-planner employs the energy-saving Kinodynamic A* algorithm to generate local collision-free guidance trajectories for trajectory segments within obstacles, taking into account multiple factors such as flight energy consumption, ground speed, and steering. Next, the planner models three cost terms (i.e., collision, smoothness, and dynamical feasibility costs) and projects the forces onto the colliding trajectory to estimate the gradient between the local collision-free guidance trajectories and their corresponding collision trajectory segments. This approach effectively wraps the trajectory out of obstacles, significantly reducing computation time by avoiding ESDF calculations while ensuring an energy-efficient trajectory. Ultimately, by applying a post-refinement procedure from [11] to further optimize the aerial-ground trajectory while maintaining dynamic feasibility, our HE-Nav system produces safe and energy-saving AGR trajectories.

Simulations and real-world experiments show that the HE-Nav enable search for safe and energy-saving pathways in occlusion and uncertain environments. The following are the key contributions of this paper:

- **HE-Nav achieves high performance.** HE-Nav achieves a 98% success rate and faster planning time (≈0.81 ms) in uncertain and occluded simulation environments.
- **HE-Nav is energy-efficient.** By novel mode switching strategy and prediction of obstacle distribution in occlusion areas in advance, resulting in a 50% decrease in energy consumption compared to the baseline.
- **HE-Nav is Occlusion-free.** LBSCNet enables real-time (20.08 FPS) and accurate inference and achieves state-of-the-art performance (IoU = 59.71) on the SemanticKITTI benchmark.

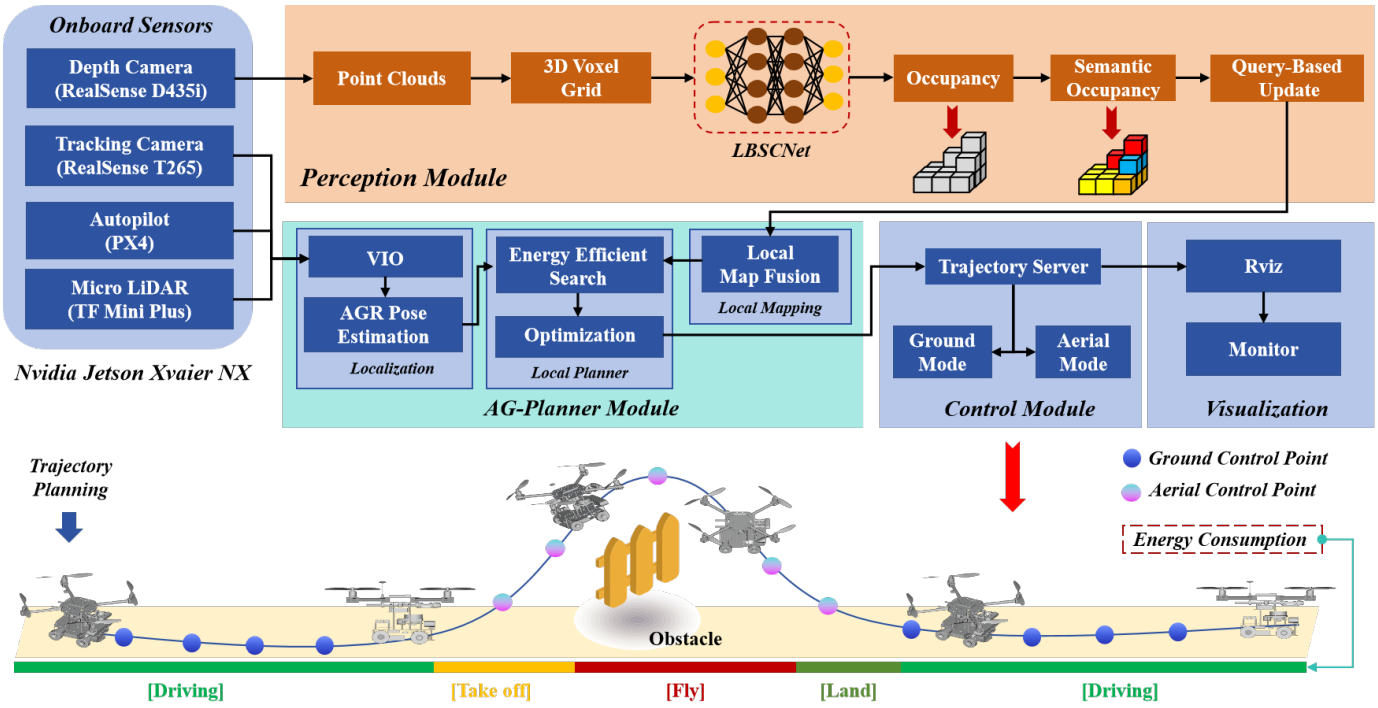


Fig. 3: HE-Nav system architecture. The perception, planning, and control modules run parallelly using onboard sensing and computing resources.

II. RELATED WORK

A. Motion Planning for AGRs

Numerous researchers have explored various aerial-ground vehicle configurations, such as incorporating passive wheels [1, 13, 3, 6, 4], cylindrical cages [14], or multi-limb [15] onto drones, while others [9, 7, 8, 10, 16] 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. 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.* [17] 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.* [18] is VoxFormer, a Transformer-based semantic scene completion framework capable of generating complete 3D volume semantics using only 2D images. Further, *Dong et al.*[19] 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.* [20] proposed S3CNet, a method designed to predict semantically complete scenes from a single unified LiDAR point cloud. *Roldao et al.* [21] 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.* [22] 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.* [23] 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 of paramount importance for aerial-ground robots, as it directly impacts their operational capabilities, endurance, and overall performance. Aerial-ground robots often perform complex tasks in diverse environments, which require them to navigate through various terrains and obstacles. Efficient energy utilization allows aerial-ground robots to operate for extended periods, reducing the need for frequent recharging and enabling them to complete tasks with minimal downtime. Moreover, energy efficiency enhances the overall performance of aerial-ground robots by enabling them to optimize their flight and ground navigation, adapt to changing environmental conditions, and respond effectively to unforeseen challenges. This adaptability is essential for various applications, such as search and rescue, surveillance, environmental monitoring, and infrastructure inspection, where prolonged operation and swift response times are critical.

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. Lightweight BEV-guided Semantic Scene Completion Network

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. 4. 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 $\left\{Sem_f^1, Sem_f^2, Sem_f^3\right\}$. Each sparse encoder block consists of a residual block [24] with sparse convolutions and an SGFE module developed in [25]. 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 $\left\{Com_f^1, Com_f^2, Com_f^3\right\}$ to provide more detailed geometric information. As shown in Fig. 4, 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

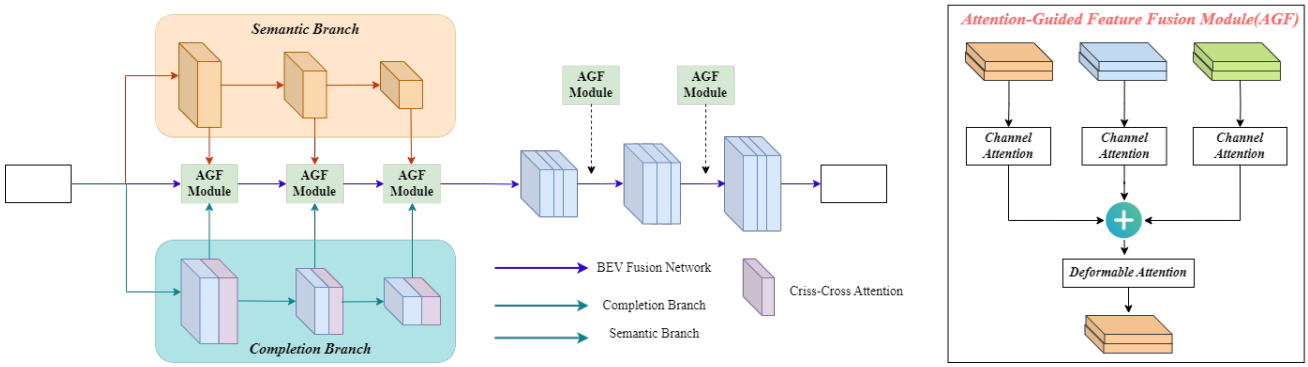


Fig. 4: Four methods were used to plan paths in a simulated square room. AGRNav demonstrates the ability to predict the distribution of obstacles in occluded areas.

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 $\left\{Sem_f^1, Sem_f^2, Sem_f^3\right\}$, 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 [26], 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 $\left\{Com_f^1, Com_f^2, Com_f^3\right\}$, 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].

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

Algorithm 1: Energy-Efficient Kinodynamic A* path search algorithm

Input: start position $start_{pt}$, velocity $start_v$, acceleration $start_a$ and time $start_t$; target position end_{pt} and velocity end_v;

Output: energy-efficient kinodynamic trajectory

Data: $time\_to\_goal$, tmp_g_score, tmp_f_score, $penalty\_g\_score$, $next\_motion\_state$, $next\_gnd\_penalty\_state$, $proximity\_penalty\_score$

```
while !open_set.empty() do
    if cur_node is near end or reach horizon then
        terminate_node = cur_node
        retrievePath(terminate_node)
        return trajectory
    end
    foreach inputs do
        foreach durations do
            tmp_g_score = (um.squaredNorm() + w_time) * tau + cur_node.g_score
            // next node flying
            if pro_state_z ≥ ground_judge then
                tmp_g_score− = cur_node.penalty_g_score
                tmp_g_score+ = flying_cost * pro_state_z/barrier_max + flying_cost_base
                // calculate penalty cost for flying
                penalty_g_score = flying_cost * pro_state_z/barrier_max + flying_cost_base
                next_motion_state = true
            end
            // next node driving
            else
                tmp_g_score− = cur_node.penalty_g_score
                penalty_g_score = 0
                next_motion_state = false
            end
            tmp_f_score = tmp_g_score + lambda_heu * estimateHeuristic(pro_state, end_state, time_to_goal)
        end
    end
end
```

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 [11], 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. $\lambda_s, \lambda_c, \lambda_f, \lambda_n$ 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.

TABLE I: Comparison of published methods on the official SemanticKITTI benchmark.

Method	*IoU*	*mIoU*	*Prec.*	*Recall*	*FPS*
SSCNet	29.83	9.53	31.71	**83.40**	12.00
SG-NN	31.26	9.90	31.60	54.50	12.00
J3S3Net	51.10	23.80	40.23	61.09	1.70
LMSCNet	55.32	17.01	77.11	66.19	13.50
LMSCNet-SS	56.72	17.62	81.55	65.07	13.50
S3CNet	45.60	29.50	48.79	77.13	1.20
SSA-SC	58.80	23.5	48.79	77.13	1.20
SCPNet	56.10	**36.70**	72.43	78.61	13.00
SCONet (our)	**59.71**	23.58	**77.60**	71.29	**20.08**

V. Experiments

We first evaluated the performance of HE-Nav's perception module (i.e., LBSCNet) on the SemanticKITTI benckmark, and then combined HE-Nav and two state of the art AGR planning frameworks in two simulation environments by deploying the pre-trained model: TABV and Fan et al. performed a performance comparison. Finally, HE-Nav was deployed on our customized AGR platform to demonstrate its performance and energy consumption in a real unknown, complex, and obstructed environment.

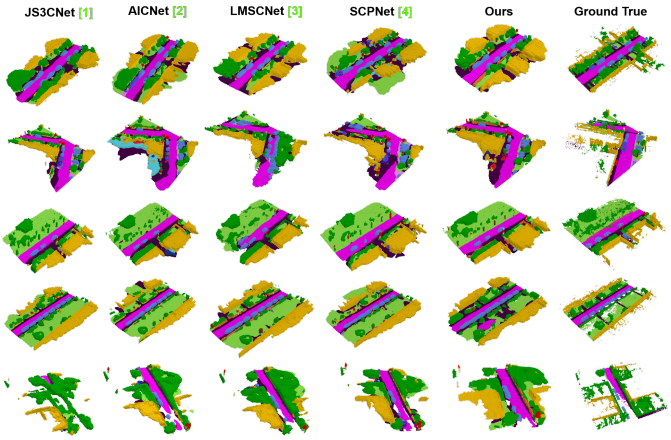


Fig. 5: Four methods were used to plan paths in a simulated square room. AGRNav demonstrates the ability to predict the distribution of obstacles in occluded areas.

A. BEV-guided lightweight semantic scene completion network

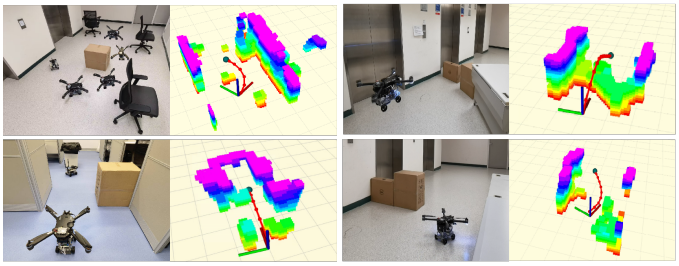
Environment	Published Venues	Energy Efficiency	Computing Time	Occlusion Prediction	Success Rate	Robot
Square Room 20m x 20m	Fan et al. [13] (IROS 2019)	300	3.66 ms	None	74%	
	Zhang et al. [1] (ICRA 2022)	200	3.36 ms	None	94%	
	EH-Nav (Ours)	100	0.81 ms	59.7	98%	
Corridor 3m x 30m	Fan et al. [13] (IROS 2019)	300	3.66 ms	None	74%	
	Zhang et al. [1] (ICRA 2022)	200	3.36 ms	None	94%	
	EH-Nav (Ours)	100	0.81 ms	59.7	98%	

Fig. 6: 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.

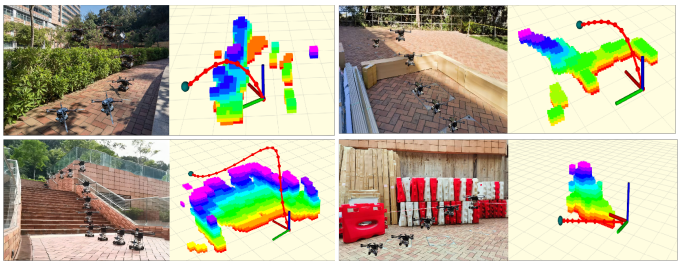
B. Simulated Air-Ground Robot Navigation

We simulated an air-ground robot navigation scenario within a complex environment. The experimental setting consists of a $20m \times 20m$ square room and a $3m \times 30m$ corridor, which are filled with random obstacles, leading to numerous occlusion spaces and unknown regions throughout the scene. The air-ground robot is required to navigate from its starting point to its designated destination. The initial ground and flight speeds are set to 1 m/s and 3 m/s, respectively, with ground speed adjusted to 1.5 m/s during speed compensation in passable areas.

Quantitative Results. We conducted a comparative analysis of our AGRNav navigation framework against two mapping-based and one learning-based navigation method in a square room and corridor scenario. 100 trials with varying obstacle placements, we recorded the average travel time, length, energy consumption, and success rate (i.e., no collisions) for all 4 methods.



(a) Indoor Real-World Experiments.



(b) Outdoor real-world Experiments.

Fig. 7: 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.

Table 1 shows that our AGRNav outperforms the other three approaches, achieving the highest success rate (98%), since our network (SCONet) predicts a broader range of occlusion areas (in Fig. 4), and generates the path with the lowest collision rate. Furthermore, our framework substantially reduces redundant paths and cuts energy consumption by half (434.55 W) in a square room. This efficiency stems from SCONet's accurate predictions, which minimize high-energy-consuming aerial paths in favour of low-energy ground paths. Simultaneously, the predicted semantics are converted into speed compensation, contributing to the reduction of travel lengths and times. In the corridor scene, while the average travel time of [1] is shorter (16.97 s), its average energy consumption is higher due to the inability to predict occlusion areas and a greater reliance on aerial paths.

C. Real-world Air-Ground Robot Navigation

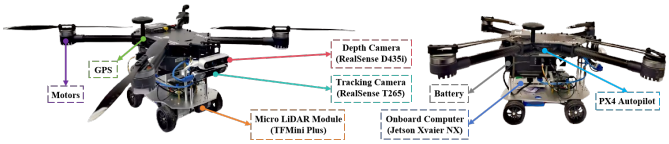


Fig. 8: Four methods were used to plan paths in a simulated square room. AGRNav demonstrates the ability to predict the distribution of obstacles in occluded areas.

Our custom AGR platform, in Fig. 5, is composed of a quadrotor with a 600mm diagonal wheelbase. This platform employs the Prometheus [**?**] software system and is equipped with a RealSense D435i depth camera and a T265 camera. It also features a Jetson Xavier NX onboard computer for the deployed AGRNav framework. Mobility is sustained by a 10,000 mAh energy source, which enables up to 26 minutes of hovering. Table 2 shows energy usage data in different modes. More detailed configurations can be found in the supplementary material.

We evaluated the AGRNav's performance in 3 complex real-world environments where the robot's vision was obstructed by walls and bushes. In contrast to mapping-based methods that could result in potential collisions or suboptimal trajectories (Fig. 6a and Fig. 6b), our AGRNav demonstrates superior performance. In Fig. 6a, AGRNav accurately anticipates the distribution of obstacles behind the wall, reducing the risk of collisions. In Fig. 6b, SCONet effectively detects hidden obstacles, allowing AGRNav to create a shorter and smoother path, ensuring energy conservation. Finally, in Fig. 6c, AGRNav recognizes the optimal landing spot by predicting unseen obstacles behind bushes. Additionally, semantic information helps in velocity compensation, leading to shorter motion times.

VI. 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] 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.

[6] 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.

[7] 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.

[8] 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.

[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] 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.

[12] Xin Zhou, Jiangchao Zhu, Hongyu Zhou, Chao Xu, and Fei Gao. Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4101–4107. IEEE, 2021.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

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

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

VII. APPENDIX

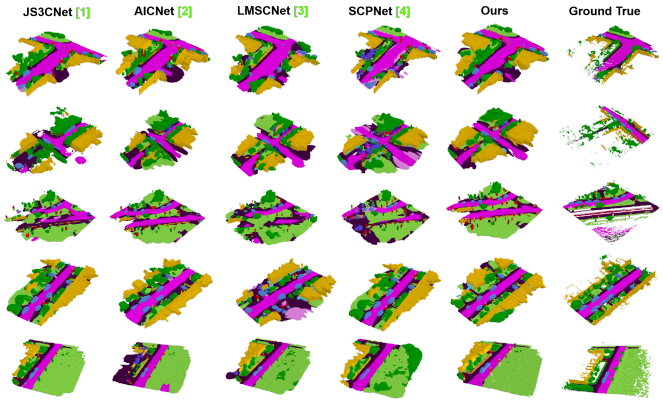


Fig. 9: Four methods were used to plan paths in a simulated square room. AGRNav demonstrates the ability to predict the distribution of obstacles in occluded areas.