AGRNav: Efficient and Energy-Saving Autonomous Navigation for Air-Ground Robots in Occlusion-Prone Environments

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

***Abstract*— The exceptional mobility and long endurance of air-ground robots are raising interest in their usage to navigate complex environments (e.g., forests and large buildings). However, such environments often contain occluded and unknown regions, and without accurate prediction of unobserved obstacles, the movement of the air-ground robot often suffers a suboptimal trajectory under existing mapping-based and learning-based navigation methods. In this work, we present AGRNav, a novel framework designed to search for safe and energy-saving air-ground hybrid paths. AGRNav contains a lightweight semantic scene completion network (SCONet) with self-attention to enable accurate obstacle predictions by capturing contextual information and occlusion area features. The framework subsequently employs a query-based method for low-latency updates of prediction results to the grid map. Finally, based on the updated map, the hierarchical path planner efficiently searches for energy-saving paths for navigation. We validate AGRNav's performance through benchmarks in both simulated and real-world environments, demonstrating its superiority over classical and state-of-the-art methods. The open-source code is available at https://github.com/jmwang0117/AGRNav.**

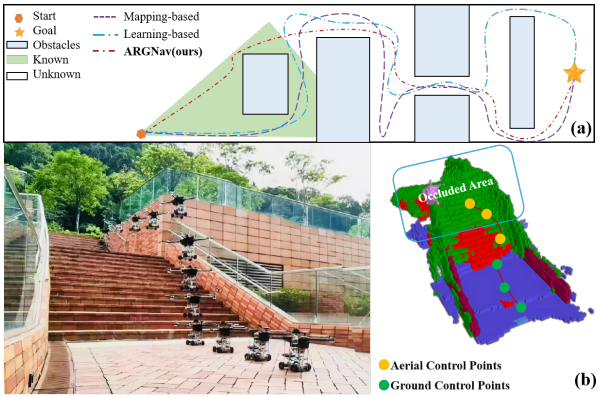

Fig. 1: **(a)** Previous navigation systems had problems predicting occlusions, resulting in higher collision probabilities and suboptimal pathways that consumed more energy. **(b)** By predicting occlusions in advance, AGRNav can minimize and avoid collisions, resulting in efficient and energy-saving paths.

I. INTRODUCTION

Air-ground robots (AGR), which are known for their outstanding mobility and long endurance, have been gaining significant interest lately and show great potential for applications in search and rescue tasks [1]–[3]. Existing works [4]–[6] have demonstrated success in fast air-ground hybrid path planning, particularly in simple and unobstructed scenarios. However, AGR navigating complex environments (e.g., forests or buildings) with occluded and unknown areas faces a dilemma since obstacles in these areas significantly affect the results of path planning, i.e., high collision probability and suboptimal energy consumption.

To enable efficient and energy-saving navigation for air-ground robots in occluded environments, existing ***mapping-based*** methods [4], [5] use sensors (e.g., cameras or LiDAR) to construct a local occupancy grid map and an Euclidean Signed Distance Field (ESDF) map [7] for fast path planning. However, since the sensors' limitation is perceiving only visible obstacles (Fig. 1a), the constructed maps exclude obstructions in occluded areas, which increases the risk of collisions and leads to higher energy consumption from unnecessary aerial paths.

In contrast, existing ***learning-based*** methods employing semantic scene completion networks [8]–[10] to predict obstacle distribution in occluded areas and then enable the path planner to reduce unnecessary paths to achieve energy savings. Some networks use 3D convolutions [11] for enhanced prediction accuracy; however, their memory-intensive nature and high inference latency make them unsuitable for real-time robotic applications. While some work [12], [13] focuses on developing lightweight networks and achieving success in real-time inference, the network's limited ability to capture features and contextual information makes its prediction accuracy drop significantly. Additionally, addressing the update delay issue is also important, as delays may cause the path planner to ignore predicted obstacle distribution, thereby leading to similar problems as in mapping-based methods.

To tackle the high inference latency of memory-intensive networks and the low prediction accuracy of lightweight networks due to their inability to capture useful features, our key observation involves integrating lightweight convolutions and self-attention mechanisms into the network. The former allows the network to perform real-time inference tasks on robotic devices, while the latter enhances the network's ability to learn long-distance dependencies and capture contextual information, which is beneficial for improving the accuracy of prediction. Moreover, regarding update delay issues in existing methods that depend on map merging and result in repeated updates of occupied voxels, one potential method is only querying and updating the occupancy status of free voxels after scanning to ensure low latency.

Based on the above observations, we present ***AGRNav***, a

*denotes corresponding author.
[1]J Wang, Z Sun, X Guan, T Shen, Z Zhang, T Duan, D Huang, H Cui is with the University of Hong Kong. [2]H Cui is with the Shanghai AI Laboratory. [3]S Zhao is with the Huawei Technologies, Co. Ltd.

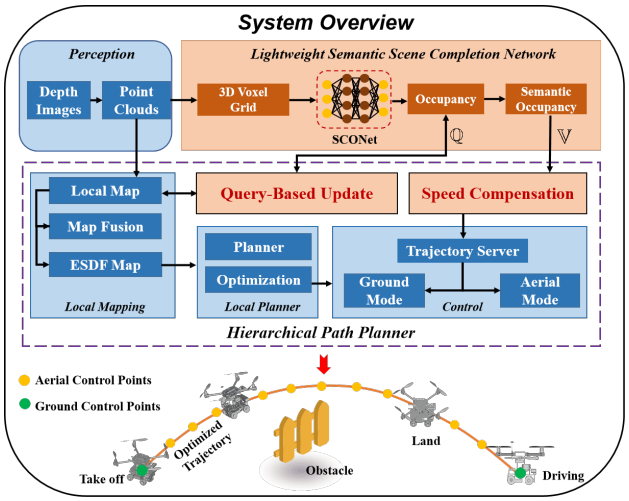

Fig. 2: The overview of our proposed Framework: AGRNav. $\mathbb{Q}$ denotes that the free voxels in the grid map query and update their occupancy status from the predicted occupancy map. $\mathbb{V}$ denotes that predicted semantics is turned into speed compensation.

novel efficient and energy-saving path-planning framework. The framework consists of two key components, the first one is a lightweight semantic scene completion network (SCONet), which is deployed on AGR and performs fast inference to accurately predict obstacle distribution and semantics. SCONet processes 3D voxel grids using depth-separable convolutions [14] rather than 3D convolutions, which greatly decreases the number of calculations. Furthermore, to enable SCONet to capture rich and dense contextual information as well as features of occlusion areas, it integrates two self-attention mechanisms. This keeps the network lightweight while enhancing its feature extraction capabilities (Fig.1b).

The hierarchical path (i.e., aerial and ground path) planner (in Fig. 2) utilizes a query-based method for low-latency occupancy updates. With the accurate predictions of SCONet, the planner minimizes collisions and energy consumption while searching for paths on an updated map containing scanned and predicted obstacles. Furthermore, it offers speed compensation for the robot using the semantics predicted, allowing for acceleration in passable areas, e.g., roads.

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

- **AGRNav is efficient.** AGRNav achieves a 98% success rate in occlusion environments while also being low-latency in updating prediction results to the grid map.
- **AGRNav is energy-saving.** By predicting obstacle distribution in advance, unnecessary aerial paths are substantially reduced, resulting in a 50% decrease in energy consumption compared to the baseline.
- **SCONet is lightweight and accurate.** SCONet enables fast (i.e., 15 FPS) and accurate inference and achieves state-of-the-art performance (IoU = 55.12) on the SemanticKITTI benchmark.

II. Related Work

A. Autonomous Navigation of Air-Ground Robots

The escalating interest in the adaptability and versatility of AGR has led to a surge of research and innovations in the field. Although many researchers prioritize mechanical structure design [2]–[5] to minimize weight and volume, it is crucial to acknowledge that establishing an efficient and energy-saving navigation framework that empowers AGR to navigate in complex environments carries greater significance. Despite this, there remains room for improvement and further investigation in current air-ground robot navigation frameworks. For example, [5] presented air-ground path planning work, but due to the absence of trajectory refinement methods, the resulting trajectories lack smoothness and dynamic feasibility. [4] proposed an energy-saving and fast autonomous navigation framework, but its "aggressive" planning strategy increases the risk of collision when navigating complex and occluded areas.

B. Navigation in Predicted Maps

Autonomous navigation with low collision probability and energy savings by predicting obstacle distribution in occluded areas has shown promising results in recent studies. However, existing methods face limitations in complex environments and high-speed navigation scenarios. For instance, [15] introduces novel perception algorithms and a controller that incorporate predicted occupancy maps for high-speed navigation. Despite its potential, the method struggles to handle complex and obstacle-dense environments due to simplistic scene design and a lower map update frequency ($\approx$ 3 Hz). Similarly, [16] employs a conditional neural process-based network to predict map turns but relies on heuristic approaches for motion planning in unknown environments. This results in greedy and inefficient trajectories without considering the unobserved environment's structure. Lastly, [12] proposed OPNet, a method that predicts occupancy grids for path planning and performs well in simple environments. However, the method faces challenges in large-scale occluded scenes because its network does not have the ability to capture the characteristics and contextual information of occluded areas.

C. Semantic Scene Completion and Occupancy Mapping

Robot sensors with narrow fields of view, such as LiDAR and depth cameras, make it difficult to monitor occluded areas. The majority of the research on predicting occluded region occupancy using limited sensor data has focused on semantic scene completion approaches. Notable works include SSCNet [10] by *Song et al.,* which uses depth images to predict occupancy and semantics for voxels. Monoscene [8] by *Cao et al.,* only requires monocular RGB images and leverages a novel 2D-3D feature projection bridge to predict occupancy and semantics for voxels. However, these memory-intensive methods are unsuitable for real-time inference on robots' devices since Monoscene [8] and VoxFormer's [9] GPU memory exceeds 10 GB during inference.

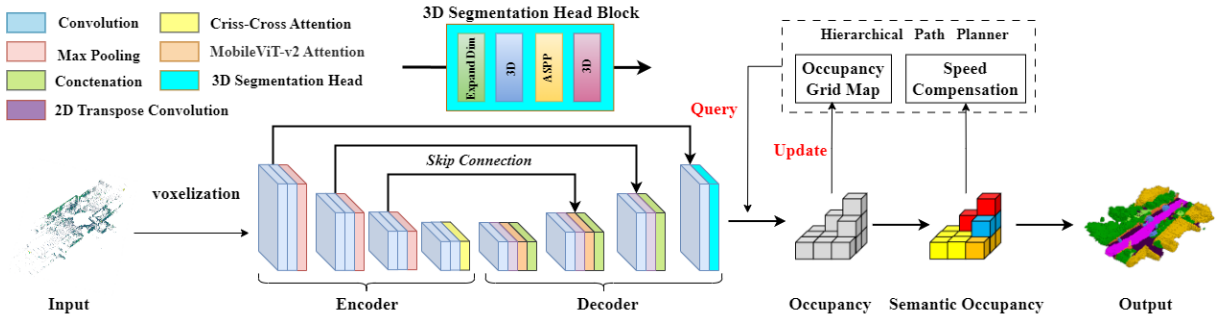

Fig. 3: **SCONet: Lightweight Semantic Scene Completion Network.** Our network employs a self-attention-driven U-Net architecture, featuring depthwise separable convolutions and segmentation heads, to perform efficient 3D scene completion and semantic segmentation.

III. System Overview

Fig.2 illustrates the proposed framework ***AGRNav***, featuring some key components: (1) the lightweight semantic scene completion network SCONet (Section IV); (2) the query-based low-latency occupancy update method (Section V-A); (3) the hierarchical path planner (Section V-B) search air-ground hybrid paths on the updated map which contains scanned obstacles and predicted obstacles.

IV. Semantic Scene Completion Network

A. SCONet Network Structure

We proposed a lightweight semantic scene completion network (SCONet) to predict the distribution of obstacles in occluded regions, as shown in Fig. 3. Point clouds are transformed into a 3D sparse voxel grid, serving as input for our 4-level U-Net style network. Each voxel is assigned a semantic label $\mathcal{L} = [i, l_1, l_2, ..., l_N], i = 0, 1$, where N is the number of semantic classes, $i = 0$ represents free voxels, and $i = 1$ represents occupancy voxels. This design allows for the effective prediction of obstacle distribution and their corresponding semantics using partial scans. The specific encoder and decoder structures are as follows:

Encoder. Instead of using memory-intensive 3D convolutions [11], we use considerably lighter depthwise separable convolutions [14] along the X and Y dimensions in the encoder, changing the height dimension (Z) into a feature dimension. This design learns features at a lower resolution while allowing direct processing of 3D voxel grids.

Decoder. By employing deconvolutions in the decoder, we up-sample the feature maps and subsequently concatenate output results to lower levels, which enhances the information flow while enabling our network to learn high-level features from coarser resolutions. Lastly, the semantics will be predicted through a 3D segmentation block that has a series of dense and dilated convolutions [17].

B. Two GPU Memory-Efficient Self-attention Mechanisms

The above design makes SCONet suitable for deployment on robotic devices for real-time inference. However, the network lacks the ability to capture contextual information and features in occluded areas, which is essential for improving prediction accuracy in such areas. Therefore, we have integrated two self-attention mechanisms into our architecture: *Criss-Cross Attention (CCA)* [18] and *MobileViT-v2 Attention* [19]. CCA, which is positioned after the encoder (in Fig. 3), enhances the network's ability to learn long-distance dependencies by collecting contextual information in horizontal and vertical directions. This enables the establishment of connections between distant features, which leads to more effective predictions of obstacles and semantics in occluded environments by comprehending the relationships among various elements (e.g., roads and walls).

Specifically, CCA takes the feature map $\mathbf{H} \in \mathbb{R}^{C \times W \times H}$ output from the fourth convolutional layer of the encoder as input, and then through two convolutional layers with 1×1 filters to acquire feature maps Q, K and attention maps A via affinity operation. Affinity can be defined as follows:

$$d_{i,u} = \mathbf{Q_u \Omega_i, u^T} \tag{1}$$

where $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{C' \times W \times H}$, $\mathbf{A} \in \mathbb{R}^{(H+W-1) \times (W \times H)}$, Q_u is a vector at each position u in the spatial dimension of feature maps Q, Ω_u is a set combined all feature vectors from K which are in the same row or column with position u. The contextual information is collected by an aggregation operation defined as follows:

$$\mathbf{H}_u^{'} = \sum_{i=0}^{H+W-1} \mathbf{A}_{i,u} \mathbf{\Phi}_{i,u} + \mathbf{H}_u \tag{2}$$

where $\mathbf{H}_u^{'}$ is a feature vector at position u and i means channel. The contextual information is added to local feature H to augment the voxel-wise representation. The CCA's ability to learn long-distance dependencies enables SCONet to effectively understand the context and relationships between various structural elements in the environment, resulting in accurate obstacle prediction. To further achieve finer-grained semantic scene completion, such as trees and cars, and better capture features of regions in occluded areas, which in turn contributes to the reduction of robot collision probability, we integrated MobileViT-v2 Attention [19] into the first (resolution 1:8) and second (resolution 1:4) layers of SCONet's Decoder. This integration allows the extraction of diverse resolution fine-grained features, which further enhances the completion of areas of occluded regions, i.e., improves obstacle prediction accuracy. With a

latency of just 3.4 ms [19], MobileViT-v2 Attention allows SCONet to maintain stronger feature capture capabilities while remaining lightweight. Mathematically, MobileViT-v2 Attention [19] can be defined as:

$$\mathbf{y} = \left(\underbrace{\sum \left(\overbrace{\sigma(\mathbf{x}W_I)}^{c_\mathbf{s} \in \mathbb{R}^k} * \mathbf{x}W_K \right)}_{c_\mathbf{v} \in \mathbb{R}^d} * \mathrm{ReLU}(\mathbf{x}W_V) \right) W_O \tag{3}$$

where $\mathbf{x}$ as input and $*$ means broadcastable element-wise multiplication and $\sum$ means summation operations. $W_O \in \mathbb{R}^{d \times d}$ means linear layer with weights. a ReLU activation to produce an output $\mathbf{x}_V \in \mathbb{R}^{k \times d}$.

V. Safe Air-Ground Hybrid Path Planner

The hierarchical path planner, building on the aerial-ground integration proposed by Zhang et al. [4], adeptly merges a query-based occupancy update mechanism, kino-dynamic trajectory searching methodologies, and a gradient-based spline optimizer. Our hierarchical planner facilitates the creation of energy-efficient hybrid trajectories and enhances overall planning efficiency.

A. Query-Based Low-Latency Occupancy Update

The SCONet network generates a predicted occupancy grid map with occupied and free voxels. Typically, this map is merged with scan-based occupancy grid maps to construct the ESDF map for planning. The time complexity of this merge operation is $O(N)$, where N is the number of voxels since it needs to traverse and combine information from both grid maps. To achieve efficient navigation and obstacle avoidance, we proposed a query-based update method with low latency. Specifically, $f(x, S_{\text{pred}})$ represents the query operation, which checks whether the voxel x exists within the predicted occupied voxel set S_{pred}. If x is predicted to be occupied (i.e., $x \in S_{\text{pred}}$), then $f(x, S_{\text{pred}}) =$ occupied; otherwise, the status of x remains free. By focusing on M relevant free voxels, where $M \leq N$, this method reduce the time complexity to $O(M)$.

$$S_{\text{updated}}(x) = \begin{cases} \text{occupied}, & \text{if } f(x, S_{\text{pred}}) = \text{occupied} \\ \text{free}, & \text{otherwise} \end{cases} \tag{4}$$

B. Efficient and Energy-saving Hierarchical Path Planner

Different from the rough path search method of *Fan et al.* [5], we also further optimize the trajectories (contains ground and aerial trajectories), that is, set the trajectories as a p_b degree uniform B-spine with control points $\mathbf{P} = \{\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_N\}$. In particular, the optimization and generation of trajectories are mainly divided into ground and aerial trajectories. When optimizing ground trajectories, we assume that the AGR moves on flat ground, so we only need to consider the two-dimensional motion control point, denoted as:

$$\mathbf{P}_g = \{\mathbf{P}_{t0}, \mathbf{P}_{t1}, \mathbf{P}_{t2}, \mathbf{P}_{t3}, ..., \mathbf{P}_{tM-1}, \mathbf{P}_{tM}\} \tag{5}$$

where $\mathbf{P}_{ti} = (x_{ti}, y_{ti}), i \in [0, M]$. Meanwhile, the aerial trajectory control points are denoted as: $\mathbf{P}_a$, We also use the following cost terms designed by *Zhou et al.* [7] to refine the trajectory:

$$f_1 = \lambda_s f_s + \lambda_c f_c + \lambda_f (f_v + f_a) \tag{6}$$

where $\lambda_s, \lambda_c, \lambda_f$ are weights for each cost terms. f_s, f_c, f_v and f_a are smoothness, collision cost, soft limits on velocity and acceleration. We set the AGR to move in the ground mode, its speed is parallel to the yaw angle. In addition, considering that our ARG adopts the Akaman structure if the trajectory is too curved, there will be a huge error, so we enforce a cost on $\mathbf{P}_g$ to limit the curvature of the terrestrial trajectory, The curvature at $\mathbf{P}_{ti}$ is defined as:

$$\mathcal{C}_i = \frac{\triangle \beta_i}{\mathcal{P}_{ti}} \tag{7}$$

where $\triangle \beta_i = |tan^{-1} \frac{\triangle y_{ti+1}}{\triangle x_{ti+1}} - tan^{-1} \frac{\triangle y_{ti}}{\triangle x_{ti}}|$. Therefore, this cost can be formulated as:

$$f_n = \sum_{i=1}^{M-1} F_n(\mathbf{P}_{ti}) \tag{8}$$

Lastly, the overall objective function is formulated as follows:

$$f_{total} = \lambda_s f_s + \lambda_c f_c + \lambda_f (f_v + f_a) + \lambda_n f_n \tag{9}$$

and we use a non-linear optimization solver $NLopt$[2] to solve this optimization problem. After path planning is completed, a setpoint on the generated trajectory is selected according to the current timestamp and then sent to the controller. The settings and selections of the aerial and ground setpoint are the same as in [4].

VI. Experiments

We evaluate AGRNav's improvement by comparing it against two mapping-based approaches and one learning-based method in two simulated environments. Moreover, we test AGRNav in three complex real-world scenarios employing a custom robot, showcasing its energy-saving advantages in practical navigation. By documenting the average energy consumption per second for AGR amidst driving and flying, we also establish a foundation for energy usage evaluation in simulated tests. Ultimately, we analyze SCONet's accuracy and real-time performance on the SemanticKITTI dataset.

A. 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.

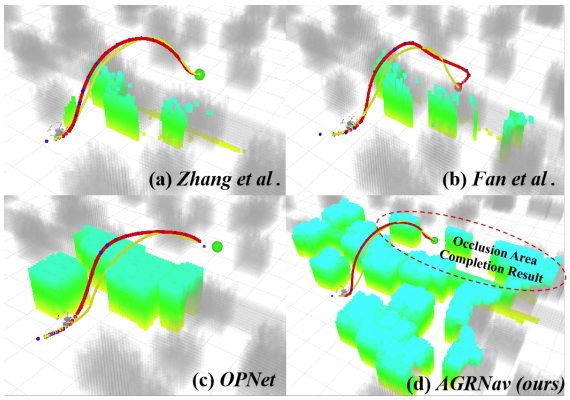

Fig. 4: Four methods were used to plan paths in a simulated square room. AGRNav demonstrates the ability to predict the distribution of obstacles in occluded areas.

TABLE I: Quantitative results in two simulation scenarios.

Env.	Method	Succ. (%)	Time (s)	Leng. (m)	Power (W)
Square Room	Fan's [5]	85.0	13.13	33.79	919.07
	Zhang's [4]	95.0	12.05	23.09	793.30
	OPNet [12]	91.0	12.90	32.12	888.04
	AGRNav(Ours)	**98.0**	**11.02**	**21.82**	**434.55**
Corridor	Fan's [5]	88.0	21.24	33.10	565.24
	Zhang's [4]	97.0	**16.97**	30.69	519.20
	OPNet [12]	90.0	18.45	32.85	534.11
	AGRNav(Ours)	**98.0**	17.50	**29.82**	**445.61**

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 and success rate (i.e., no collisions) for all 4 methods. In particular, the energy consumption of the four methods is calculated using the energy consumed per second by our customized robot when flying and driving in the real environment (Table 2). 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 (i.e., average consumption per second is 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 [4] 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.

Fig. 5: Our customized air-ground robot (AGR).

Mode	Average Power	Time (mins)
Fly	987.61 W	14
Hover	532.07 W	26
Ground	197.52 W	55

TABLE II: The power consumption comparison among different states: fly state, hovering state and rolling on the ground state.

B. Real-world Air-Ground Robot Navigation

Our custom AGR platform, in Fig. 5, is composed of a quadrotor with a 600mm diagonal wheelbase. This platform employs the Prometheus [20] 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. 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.

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

Method	*IoU*	*Prec.*	*Recall*	*FPS*	*mIoU*
SSCNet [10]	29.83	31.71	**83.40**	12.00	9.53
SG-NN [21]	31.26	31.60	54.50	12.00	9.90
J3S3Net [22]	51.10	40.23	61.09	1.70	23.80
LMSCNet [13]	50.76	64.94	62.55	13.50	17.01
S3CNet [23]	45.60	48.79	77.13	1.20	**29.50**
OPNet [12]	41.06	72.43	78.61	13.00	-
SCONet (our)	**55.12**	**85.02**	63.47	**15.00**	18.61

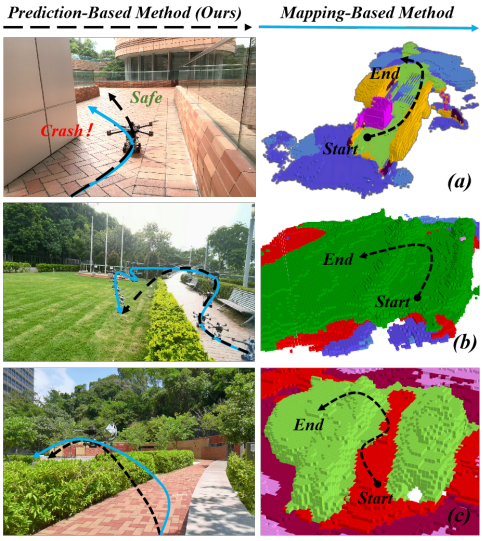

Fig. 6: Autonomous navigation experiments of AGR in 3 complex real environments.

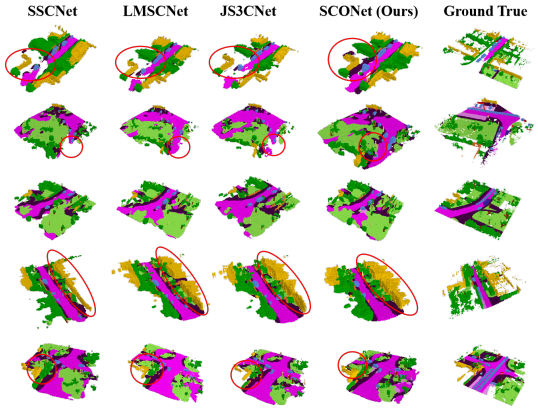

Fig. 7: Qualitative results of SCONet on the validation set of SemanticKITTI.

C. Semantic Scene Completion Network (SCONet)

Pre-trained Model. We evaluate SCONet's performance using the SemanticKITTI benchmark [24]. This benchmark provides 3D voxel grids from semantically labelled HDL-64 LiDAR scans in complex outdoor urban environments. The input and ground truth grids are sparse, with dimensions of $256 \times 256 \times 32$ and a $0.2m$ voxel size. Our evaluation focuses on completion metrics like IoU, precision, and recall, along with the semantic metric mIoU. We use the original training and validation splits and apply x-y flipping augmentation for better generalization. The training employs the Adam optimizer with a 0.001 learning rate, scaled by 0.98 per epoch. Running on a machine containing 4 NVIDIA RTX3090, the training converges in approximately 24 hours.

Quantitative Results. Table 3 shows that our SCONet surpasses competing methods, achieving the highest completion metric IoU score of 55.12. This evaluation was performed on the official server using a hidden test set. Notably, despite having a lower mIoU than S3CNet and J3S3Net, SCONet's inference speed is approximately 15 times faster. The use of depthwise separable convolutions in the encoder, rather than resource-heavy 3D convolutions, contributes to this real-time efficiency (15 FPS on a single RTX 3090 GPU).

Method	IoU ↑	mIoU ↑
SCONet (ours)	54.92	17.69
w/o Depth-Separable Convolution	54.15	17.26
w/o Criss-Cross Attention	52.80	16.37
w/o MobileViT-v2 Attention	53.56	17.11

TABLE IV: Ablation study of our model design choices on the SemanticKITTI [24] validation set.

Qualitative Results. As depicted in Fig. 7, SCONet, when compared to the baseline model, demonstrates improved completion of various structural objects, such as vehicles and trees (row 5). Even in areas obstructed by trees or walls, the network successfully manages the completion, as evidenced in rows 1 and 4. This ability plays a crucial role in facilitating efficient and safe path planning for subsequent stages.

Ablation experiment results. Ablation studies on the SemanticKITTI validation set (Table 4) highlight the significance of two key components in our network: self-attention mechanisms and depth-separable convolutions. 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, MobileViT-v2 Attention captures local scene features, such as occluded areas, with low computational overhead. *Without MobileViT-v2 Attention* leads to a 2.47% decline in IoU. Furthermore, depth-separable convolutions significantly reduce the number of parameters.

VII. Conclusions

In this paper, we introduce AGRNav, an efficient and energy-saving autonomous navigation framework for air-ground robots, featuring the key component SCONet, which outperforms state-of-the-art models in prediction accuracy and inference time. Additionally, a hierarchical path planner, improved by a query-based low-latency update method, considers obstacles in occluded areas to generate paths. This approach not only minimizes collision risk but also reduces energy consumption by 50% compared to the baseline by cutting down high-energy aerial paths. The system's robustness has been extensively validated through experiments in both simulated and real-world environments, with videos available in the supporting materials.

ACKNOWLEDGMENT

The work is supported in part by National Key R&D Program of China (2022ZD0160200), HK RIF (R7030-22), HK ITF (GHP/169/20SZ), the Huawei Flagship Research Grants in 2021 and 2023, and HK RGC GRF (Ref: HKU 17208223), the HKU-SCF FinTech AcademyR&D Funding Schemes in 2021 and 2022, and the Shanghai Artificial Intelligence Laboratory (Heming Cui is a courtesy researcher in this lab).

REFERENCES

[1] A. Kalantari and M. Spenko, "Design and experimental validation of hytaq, a hybrid terrestrial and aerial quadrotor," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4445–4450.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] H. T. Suh, X. Xiong, A. Singletary, A. D. Ames, and J. W. Burdick, "Energy-efficient motion planning for multi-modal hybrid locomotion," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7027–7033.

[7] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[8] 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.

[9] 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.

[10] 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.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[12] L. Wang, H. Ye, Q. Wang, Y. Gao, C. Xu, and F. Gao, "Learning-based 3d occupancy prediction for autonomous navigation in occluded environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4509–4516.

[13] 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.

[14] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[15] K. D. Katyal, A. Polevoy, J. Moore, C. Knuth, and K. M. Popek, "High-speed robot navigation using predicted occupancy maps," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5476–5482.

[16] A. Elhafsi, B. Ivanovic, L. Janson, and M. Pavone, "Map-predictive motion planning in unknown environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8552–8558.

[17] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[18] 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.

[19] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," *arXiv preprint arXiv:2206.02680*, 2022.

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

[21] A. Dai, C. Diller, and M. Nießner, "Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 849–858.

[22] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. 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*, vol. 35, no. 4, 2021, pp. 3101–3109.

[23] 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.

[24] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307.