OMEGA: Efficient Occlusion-Aware Navigation for Air-Ground Robot in Dynamic Environments via State Space Model

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

***Abstract*—Air-ground robots (AGRs) are widely used in surveillance and disaster response due to their exceptional mobility and versatility (i.e., flying and driving). Current AGR navigation systems perform well in static occlusion-prone environments (e.g., indoors) by using 3D semantic occupancy networks to predict occlusions for complete local mapping and then computing Euclidean Signed Distance Field (ESDF) for path planning. However, these systems face challenges in dynamic, occlusion scenes due to limitations in perception networks' low prediction accuracy and path planners' high computation overhead.**

In this paper, we propose OMEGA, which contains OccMamba with an Efficient AGR-plAnner to address the above-mentioned problems. OccMamba adopts a novel architecture that separates semantic and occupancy prediction into independent branches, incorporating two mamba blocks within these branches. These blocks efficiently extract semantic and geometric features in 3D environments with linear complexity, ensuring that the network can learn long-distance dependencies to improve prediction accuracy. Semantic and geometric features are combined within the Bird's Eye View (BEV) space to minimise computational overhead during feature fusion. The resulting semantic occupancy map is then seamlessly integrated into the local map, providing occlusion awareness of the dynamic environment. Our AGR-Planner utilizes this local map and employs kinodynamic A* search and gradient-based trajectory optimization to guarantee planning is ESDF-free and energy-efficient. Extensive experiments demonstrate that OccMamba outperforms the state-of-the-art 3D semantic occupancy network with 25.0% mIoU. End-to-end navigation experiments in dynamic scenes verify OMEGA's efficiency, achieving a 96% average planning success rate. Code and video are available at https://jmwang0117.github.io/OMEGA/.

***Index Terms*—Deep learning for visual perception, Autonomous navigation, 3D semantic occupancy prediction**

I. INTRODUCTION

In recent years, air-ground robots (AGRs) have attracted significant attention from both academia [1]–[4] and industry due to their versatile navigation capabilities in the ground and aerial domains. To enhance AGR navigation in occlusion-prone environments, existing works employ sensors, such as depth cameras, to collect point clouds. These point clouds are then processed by 3D semantic occupancy networks [5]–[8], which predict occluded areas and generate a complete local map. Based on the local map, an Euclidean Signed Distance Field (ESDF) map [2], [4] is constructed for path planning.

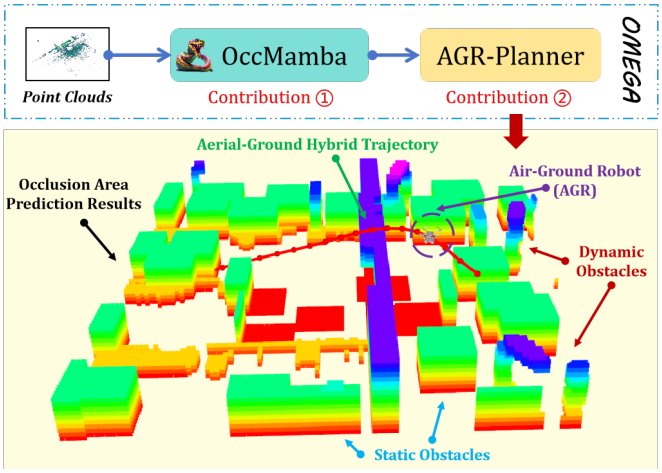

Fig. 1: OMEGA is the first AGR-specific navigation system that enables occlusion-aware mapping and pathfinding in dynamic scenarios. It integrates OccMamba for real-time obstacle prediction from point clouds and updates local maps accordingly, while AGR-Planner rapidly generates reliable paths using the updated local map.

During path searching, ground trajectories are prioritized to improve energy efficiency, while aerial navigation is used only when encountering impassable obstacles.

Unfortunately, while these approaches have shown success in static occlusion-prone environments, such as indoor or forest settings, they struggle in dynamic environments with moving obstacles (Table I). This limitation can be attributed to two main factors: the *perception network's* low prediction accuracy and high inference latency, and the *path planner's* high computational overhead. Firstly, although the use of 3D convolutions [5] and transformers [9], [10] in 3D semantic occupancy networks has improved occupancy prediction accuracy, it has also introduced computational overhead. This increased computational demand challenges real-time inference on resource-constrained AGR devices (e.g., Jetson Xavier NX). Secondly, current AGR path planners allocate nearly 70% of the total local planning time to construct the Euclidean Signed Distance Field (ESDF) map. This extensive processing time is impractical for dynamic environments with rapidly changing scenes, as it heightens the risk of collisions.

Our key insight to address the above limitations is to propose an efficient 3D semantic occupancy network that ensures real-time inference while improving prediction accuracy. We draw inspiration from state space models (SSMs) and

*Corresponding author.
[1]The University of Hong Kong, Hong Kong SAR, China. [2]Shanghai AI Laboratory. [3]Huazhong University of Science and Technology, Wuhan, China. jmwang@cs.hku.hk

their improved versions, such as Mamba [11] and Mamba-2 [12] networks. By integrating Mamba blocks into a new 3D semantic occupancy network architecture, we aim to enable the network to model long-distance dependencies and perform parallel feature processing. This approach facilitates real-time occlusion prediction in highly dynamic environments, resulting in more complete local maps. Furthermore, while *Zhou et al.* [13] developed an ESDF-free path planner for quadcopters, it does not address AGR-specific requirements in dynamic scenarios, particularly energy efficiency and dynamic constraints. Consequently, exploring ESDF-free AGR path planners is crucial to improving overall navigation performance.

Based on these insights, We introduce OMEGA (Fig. 2), consisting of two key components: OccMamba and AGR-planner. OccMamba, the first 3D semantic occupancy network based on state-space models (SSMs), departs from previous networks that jointly learn semantics and occupancy by separating these predictions into different branches (Fig. 3). This separation enables specialized learning within each domain, improving prediction accuracy and fully leveraging the complementary properties of semantic and geometric features in the subsequent feature fusion stage. We also integrate novel Sem-Mamba and Geo-Mamba blocks into these branches to capture long-distance dependencies critical for semantic accuracy and occupancy prediction. By projecting features into the bird's-eye view (BEV) space, we reduce fusion latency to achieve real-time inference. Prediction results are merged into the local map using a low-latency update method [4], ensuring an up-to-date and accurate map in highly dynamic environments.

During the planning phase, we propose AGR-Planner, which builds on EGO-Planner [14] for aerial-ground hybrid path planning. To address AGR-specific requirements, particularly energy efficiency and dynamic constraints, we add additional energy costs to motion primitives involving aerial destinations, promoting energy efficiency. Simultaneously, we account for AGRs' non-holonomic constraints by limiting ground control point curvature. Finally, by integrating the obstacle distance estimation method from [14] and backend trajectory optimization, AGR-Planner generates ESDF-free, energy-efficient, smooth, collision-free, and dynamically feasible trajectories.

We first assessed OccMamba on the SemanticKITTI benchmark, comparing its accuracy and inference speed to a leading occupancy network. Then, we tested OMEGA (Fig. 2) in simulated and real dynamic environments, contrasting it with two AGR navigation baselines, showcasing its superior efficiency. Our evaluation reveals:

- **OccMamba is efficient and real-time.** OccMamba achieves state-of-the-art performance (IoU = 59.9) on the SemanticKITTI benchmark and enables high-speed inference (i.e., 22.1 FPS). (§ V-B)
- **OMEGA is efficient.** OMEGA achieved success rates of 98% in the simulation scenarios while having the shortest average movement time. (§ V-C)
- **OMEGA is energy-saving.** The results of real dynamic environment navigation show that OMEGA can save about 18% of energy consumption. (§ V-D)

TABLE I: Comparison with baseline AGR systems.

Method	Dyn. Env.	Occl. Aware	Mov. Time	Succ. Rate	Energy
HDF [1]	✗	✗	✗	✗	✗
TABV [2]	✗	✗	✗	✗	✗
M-TABV [3]	✗	✗	✗	✗	✗
AGRNav [4]	✗	✓	✗	✗	✓
OMEGA (Ours)	✓	✓	✓	✓	✓

II. RELATED WORK

A. Dynamic Navigation System for AGRs

Numerous researchers have explored various aerial-ground robot configurations, such as incorporating passive wheels [2], [15], [16], cylindrical cages [17], or multi-limb [18] onto drones. Recently, *Fan et al.* [1] address ground-aerial motion planning. Their approach initially employs the A* algorithm to search for a geometric path as guidance, favouring ground paths by adding extra energy costs to aerial paths. *Zhang et al.* [2] proposed a path planner and controller capable of path searching, but it relies on an ESDF map. The intensive computation and limited perception of occluded areas lead to a low success rate in path planning and increased energy consumption. *Wang et al.* [4] proposed AGRNav, the first AGR navigation system with occlusion perception capability. Although it performs well in static environments, its simple perception network structure design and the defects of the path planner make it difficult to operate in complex and changing dynamic environments. This research unveils OMEGA, a groundbreaking navigation system engineered for optimal occlusion awareness and path planning in dynamic environments. It is the inaugural initiative to improve AGR navigation under such challenging conditions significantly.

B. 3D Semantic Occupancy Prediction

3D semantic occupancy prediction is crucial for interpreting occluded environments, as it discerns the spatial layout beyond visual obstructions by merging geometry with semantic clues. This process enables autonomous systems to anticipate hidden areas, crucial for safe navigation and decision-making. Research on 3D semantic occupancy prediction can be summarized into three main streams: *Camera-based* approaches capitalize on visual data, with pioneering works like MonoScene by Cao et al. [5] exploiting RGB inputs to infer indoor and outdoor occupancy. Another notable work by *Li et al.* [10] is VoxFormer, a transformer-based semantic occupancy framework capable of generating complete 3D volume semantics using only 2D images. *LiDAR-based* approaches like S3CNet by Cheng et al. [6], JS3C-Net by Yan et al. [19], and SSA-SC by Yang et al. [20], which adeptly handle the vastness and variability of outdoor scenes via point clouds. *Fusion-based* approaches aim to amalgamate the contextual richness of camera imagery with the spatial accuracy of LiDAR data. The Openoccupancy benchmark by Wang et al. [21] is a testament to this synergy, providing a platform to assess the performance of integrated sensor approaches.

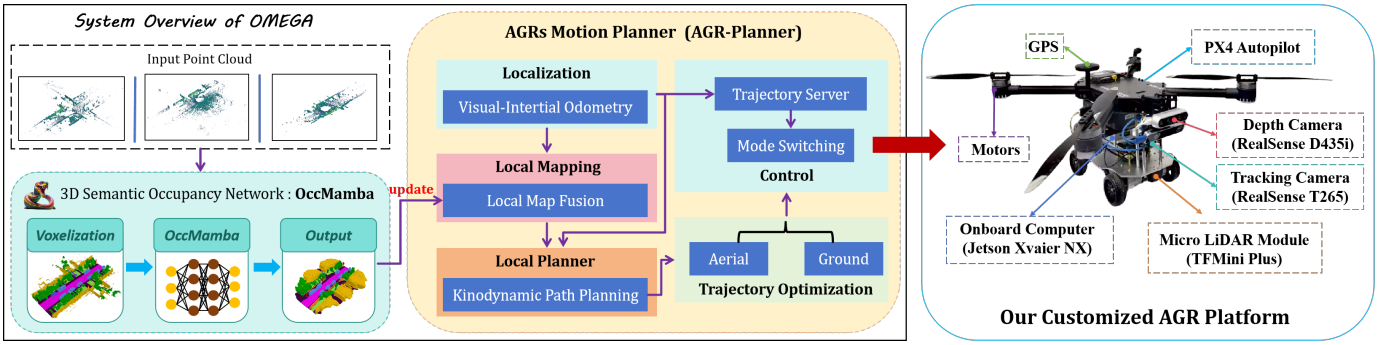

Fig. 2: OMEGA system architecture. The perception network (i.e., OccMamba) and AGR-planner run asynchronously on the onboard computer, connected through a query-based map update method from [4] to ensure real-time local map updates with prediction results.

C. State Space Models (SSMs) and Mamba

State-Space Models (SSMs), including the S4 model, have proven effective in sequence modelling, particularly for capturing long-range dependencies, outperforming traditional CNNs and Transformers. The Mamba model [12] builds on this by introducing data-dependent SSM layers, offering enhanced adaptability and significantly improved efficiency in processing long sequences. This has led Mamba to excel in various domains, including point cloud analysis [22] and video understanding [23], due to its scalability and versatility. Leveraging Mamba's strengths, we developed OccMamba, a pioneering network for 3D semantic occupancy prediction optimized for AGR navigation in occlusion-rich environments. OccMamba integrates custom mamba blocks to adeptly handle long-range dependencies essential for understanding complex scenes. This advancement broadens Mamba's utility and sets new benchmarks in 3D semantic occupancy prediction efficiency.

III. 3D Semantic Occupancy Network of OMEGA

OccMamba (Fig. 3) features three branches: semantic, geometric, and BEV fusion. The semantic and geometric branches are supervised by multi-level auxiliary losses, which are removed during inference. Multi-scale features generated by these two branches are fused in the BEV space to alleviate the computational overhead caused by dense feature fusion.

A. OccMamba Network Structure

***Semantic Branch*:** The semantic branch consists of a voxelization layer and three encoder Sem-Mamba Blocks with the same structures. The input point cloud $P \in \mathbb{R}^{N\times 3}$ is converted to a multi-scale voxel representation at a voxel resolution s by the voxelization layer. These voxels are then aggregated by maximum pooling to obtain a unified feature vector for each voxel. The vectors from different scales are merged to form the final voxel feature V_{f_m}, with a size of $L\times W\times H$, where f_m represents the voxel index. The semantic features $\{S_f^1, S_f^2, S_f^3\}$ are projected into the bird's-eye view (BEV) space by assigning a unique BEV index to each voxel based on its f_m value. Features sharing the same BEV index are aggregated using max pooling, resulting in sparse BEV features. These sparse features are then densified using the densification function of Spconv [24] to produce dense BEV features $\{BEV_f^{sem,1}, BEV_f^{sem,2}, BEV_f^{sem,3}\}$.

***Sem-Mamba Block*:** Mamba [12], a selective state-space model, has recently outperformed CNN- and Transformer-based approaches in various vision tasks due to its efficient long-distance sequence modelling and linear-time complexity. These characteristics make Mamba promising for improving 3D semantic occupancy prediction accuracy while maintaining fast reasoning in dynamic environments. Inspired by Mamba's success, we introduce the Sem-Mamba block as the semantic branch encoder in our network to enable efficient semantic representation learning. Specifically, state space models introduce hidden states $h(t) \in \mathbb{R}^N$ to map inputs $x(t) \in \mathbb{R}^L$ to outputs $y(t)$, with the continuous state space dynamics governed by:

$$h^{'}(t) = \mathbf{A}h(t) + \mathbf{B}x(t), y(t) = \mathbf{C}h(t) \tag{1}$$

Using a time scale parameter $\triangle$, the Mamba model discretizes the continuous parameters, yielding the discretized state space equations:

$$\overline{\mathbf{A}} = exp(\triangle\mathbf{A}), \overline{\mathbf{B}} = (\triangle\mathbf{A})^{-1}(exp\triangle(\mathbf{A}) - \mathbf{I}) \cdot \triangle\mathbf{B} \tag{2}$$

$$h(t) = \overline{\mathbf{A}}h_{t-1} + \overline{\mathbf{B}}x_t, y_t = \mathbf{C}h_t \tag{3}$$

The global convolution kernel $\overline{\mathbf{K}} \in \mathbb{R}^L$ is used to calculate the output y:

$$\overline{\mathbf{K}} = (C\overline{B}, C\overline{AB}, ..., C\overline{A}^{m-1}\overline{B}), y = x * \overline{\mathbf{K}} \tag{4}$$

In each Sem-Mamba Block, BEV features $\{BEV_f^{sem,1}, BEV_f^{sem,2}, BEV_f^{sem,3}\}$ serve as the input x to the Mamba module (Fig. 3). By applying discretized SSM dynamics and global convolution kernels, the mamba block effectively processes the BEV features, resulting in an enhanced feature representation with richer long-distance dependencies. This enhanced representation improves the performance of semantic occupancy prediction by capturing and strengthening the spatial relationships between semantic elements, while the mamba block's properties ensure real-time reasoning in dynamic environments.

***Geometry Branch*:** The geometric branch (Fig. 3) begins with an input layer using a $7 \times 7 \times 7$ kernel and comprises three Geo-Mamba blocks. Each Geo-Mamba block maintains a consistent architecture, combining a residual block with a Mamba block and a BEV projection module. The residual

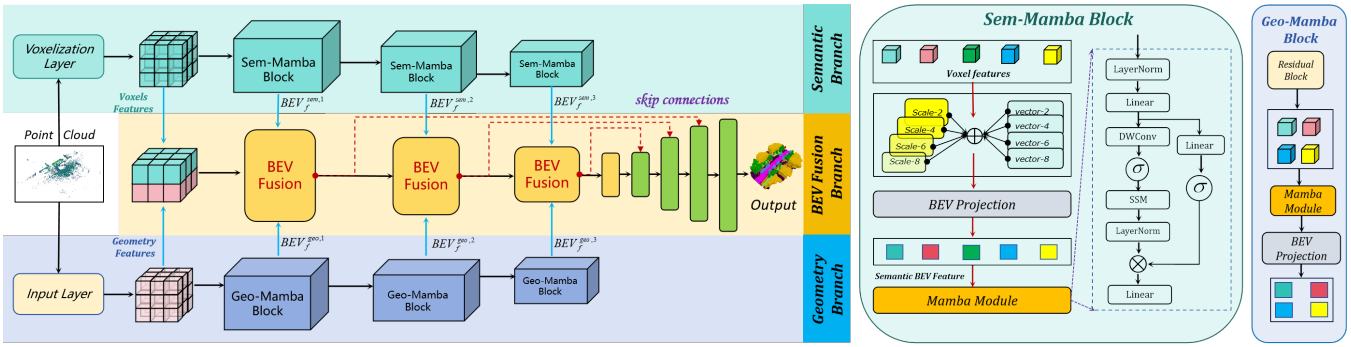

Fig. 3: The overview of the proposed OccMamba. It consists of semantic, geometry and BEV fusion branches. Meanwhile, lightweight MLPs serve as auxiliary heads during training, attached after each encoder block in the semantic and completion branches for voxel predictions. At the inference stage, these heads are detached to preserve a streamlined network architecture.

block first processes the voxels $V \in \mathbb{R}^{1\times L\times W\times H}$ obtained from the point cloud, and its output features serve as the input x to the Mamba block. The Mamba block generates multi-scale dense geometric features $\{G_f^1, G_f^2, G_f^3\}$, which enrich the captured geometric details. By leveraging the Mamba block's ability to capture long-range dependencies with linear complexity, the geometric branch can effectively process and refine the geometric information within the voxels. Subsequently, the dense 3D features are aligned along the z axis, and 2D convolutions are applied to compress the feature dimensions. This step produces dense BEV features $\{BEV_f^{com,1}, BEV_f^{geo,2}, BEV_f^{geo,3}\}$, which are suitable for fusion with the semantic features.

***BEV Feature Fusion Branch*:** Our BEV fusion network adopts a U-Net architecture with 2D convolutions. The encoder consists of an input layer and four residual blocks, where the resolution of the first three residual blocks corresponds to the resolution of the semantic and geometric branches. After each residual block, a feature fusion module from [25] is employed to take the output of the previous stage and the semantic/geometric representation at the same scale as the input. This module outputs fused features that contain informative semantic context and geometric structure. The decoder of the BEV fusion network upscales the compressed features from the encoder three times, through skip connections. These skip connections allow the decoder to recover spatial details that may have been lost during the encoding process, thereby improving the overall quality of the output. Finally, the last convolution layer of the decoder generates the 3D semantic occupancy prediction: $\mathcal{O} \in \mathbb{R}^{((C_n+1)*L)*H*W}$.

B. Optimization

Based above of our methods above, there are four main terms in our loss function. The semantic branch is optimized using lovasz loss [26] and cross-entropy loss [27]. The semantic loss L_{sem} is the sum of the loss at each stage, expressed as follows:

$$L_{sem} = \sum_{i=1}^{3}(L_{cross,i} + L_{lovasz,i}) \tag{5}$$

The training loss L_{com} for this branch is calculated as follows:

$$L_{com} = \sum_{i=1}^{3}(L_{binary\_cross,i} + L_{lovasz,i}) \tag{6}$$

where i denotes the $i-th$ stage of the completion branch and L_{binary_cross} indicates the binary cross-entropy loss. The BEV loss L_{bev} is :

$$L_{bev} = 3 \times (L_{cross} + L_{lovasz}) \tag{7}$$

We train the whole network end-to-end. The overall objective function is:

$$L_{total} = L_{bev} + L_{sem} + L_{com} \tag{8}$$

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

IV. AGR Motion Planner

We introduce AGR-Planner, a novel gradient-based local planner tailored for AGRs built upon EGO-Planner [14]. It features a Kinodynamic A* algorithm for efficient pathfinding and a gradient-based method for trajectory optimization, streamlining the planning process.

A. Kinodynamic Hybrid A* Path Searching

Our AGR-Planner begins by generating a preliminary "initial trajectory" ι (see Fig. 4a), which initially disregards obstacles by incorporating random coordinate points, anchored by the start and end locations. Subsequently, for any "collision trajectory segment" found within obstacles, we employ the kinodynamic A* algorithm to create a "guidance trajectory segment" τ. This segment is defined using motion primitives rather than straight lines for edges during the search, incorporating additional energy consumption metrics for flying (Fig. 4a). This approach encourages the planning of ground trajectories, resorting to aerial navigation only when confronting significant obstacles, thus optimizing energy efficiency.

B. Gradient-Based B-spline Trajectory Optimization

***B-spline Trajectory Formulation*:** In trajectory optimization (Fig. 4b), 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 [4] the value of a B-spline can be evaluated as:

$$\Theta(u) = [1, u, ..., u^p] \cdot M_{p_b+1} \cdot [Q_{i-p_b}, Q_{i-p_b+1}, ..., Q_i]^T \quad (9)$$

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:

$$V_i = \frac{Q_{i+1} - Q_i}{\triangle t}, A_i = \frac{V_{i+1} - V_i}{\triangle t}, J_i = \frac{A_{i+1} - A_i}{\triangle t} \quad (10)$$

***Collision Avoidance Force Estimation*:** Inspired by [13], 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. 4a). 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:

$$D_{ij} = (Q_i - p_{ij}) \times v_{ij} \quad (11)$$

Because the guide path τ is energy-saving, the generated path is also energy efficient (in Fig. 4a). To discover multiple viable paths that navigate through different local minima in a dynamic environment, we apply the method described in [14]. This method generates distance fields in various directions by reversing vector v_1 to obtain $v_2 = -v_1$. Following this, a search process identifies a new anchor point p_2 on the surface of the obstacle along v_2, as shown in Fig. 4a. This approach enables us to evaluate alternative routes and select the most cost-effective path for navigation.

***Air-Ground Hybrid Trajectory Optimization*:** Based on the special properties of AGR bimodal, we first adopt the following cost terms designed by *Zhou et al.* [14]:

$$\min_Q J_{AGR} = \sum \lambda_\phi J_\phi \quad (12)$$

where $\phi = \{s, c, d, t\}$ and the subscripted λ indicates the corresponding weights. In addition, J_s is the smoothness penalty, J_c is for collision, J_d is for dynamically feasibility, and J_t is for terminal progress. $\lambda_s, \lambda_c, \lambda_d, \lambda_t$ are weights for each cost terms. Meanwhile, J_s and J_t belongs to *minimum error* which minimize the total error between a linear transformation of decision variables $L(Q)$ and a desired value D. J_c and J_d belongs to *soft barrier constraint* which penalize decision variables exceeding a specific threshold ζ. Subsequently, based on our observations, AGR also 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:

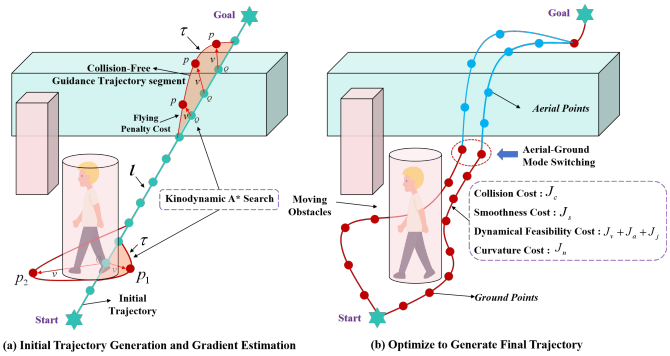

Fig. 4: Illustration of AGR-Planner and topological trajectory generation.

$$J_n = \sum_{i=1}^{M-1} F_n(Q_{ti}) \quad (13)$$

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

$$F_n(Q_{ti}) = \begin{cases} (C_i - C_{max})^2, C_i > C_{max}, \\ 0, C_i \leq C_{max} \end{cases} \quad (14)$$

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:

$$\min_Q J_{AGR} = \lambda_s J_s + \lambda_c J_c + \lambda_d J_d + \lambda_d J_d + \lambda_n J_n \quad (15)$$

The optimization problem is addressed with the NLopt. Meanwhile, the same methods as in [4] are used for trajectory tracking and control, as well as additional mode switching.

V. Evaluation

In this section, we evaluate OccMamba on the SemanticKITTI benchmark and select the pre-trained model with the highest IoU for offline deployment and inference. Integrating OccMamba with AGR-Planner, we establish a comprehensive **Omega System**. Next, we assess the autonomous navigation efficiency of AGR using Omega in simulated and real-world dynamic environments, focusing on metrics including planning success rate, average movement and planning time, and energy consumption.

A. Evaluation setup

3D Semantic Occupancy Network: We trained OccMamba using the outdoor SemanticKITTI dataset on a single NVIDIA

TABLE II: **3D Semantic occupancy prediction results on SemanticKITTI test set.** The C and L denote camera and LiDAR.

Method	Modality	IoU ↑	mIoU ↑	road (15.30%)	sidewalk (11.13%)	parking (1.12%)	other-grnd (0.56%)	building (14.1%)	car (3.92%)	truck (0.16%)	bicycle (0.03%)	motorcycle (0.03%)	other-veh. (0.20%)	vegetation (39.3%)	trunk (0.51%)	terrain (9.17%)	person (0.07%)	bicyclist (0.07%)	motorcyclist (0.05%)	fence (3.90%)	pole (0.29%)	traf.-sign (0.08%)	FPS
MonoScene [5]	C	34.2	11.1	54.7	27.1	24.8	5.7	14.4	18.8	3.3	0.5	0.7	4.4	14.9	2.4	19.5	1.0	1.4	0.4	11.1	3.3	2.1	1.1
OccFormer [9]	C	34.5	12.3	55.9	30.3	31.5	6.5	15.7	21.6	1.2	1.5	1.7	3.2	16.8	3.9	21.3	2.2	1.1	0.2	11.9	3.8	3.7	1.8
VoxFormer [10]	C	43.2	13.4	54.1	26.9	25.1	7.3	23.5	21.7	3.6	1.9	1.6	4.1	24.4	8.1	24.2	1.6	1.1	13.1	6.6	5.7	8.1	1.5
TPVFormer [28]	C	34.3	11.3	55.1	27.2	27.4	6.5	14.8	19.2	3.7	1.0	0.5	2.3	13.9	2.6	20.4	1.1	2.4	0.3	11.0	2.9	1.5	1.0
LMSCNet [29]	L	55.3	17.0	64.0	33.1	24.9	3.2	38.7	29.5	2.5	0.0	0.0	0.1	40.5	19.0	30.8	0.0	0.0	0.0	20.5	15.7	0.5	21.3
SSC-RS [25]	L	59.7	24.2	**73.1**	44.4	38.6	17.4	**44.6**	36.4	5.3	10.1	5.1	11.2	44.1	26.0	41.9	4.7	2.4	0.9	30.8	15.0	7.2	16.7
SCONet [4]	L	56.1	17.6	51.9	30.7	23.1	0.9	39.9	29.1	1.7	0.8	0.5	4.8	41.4	27.5	28.6	0.8	0.5	0.1	18.9	21.4	8.0	20.0
JS3C-Net [19]	L	56.6	23.8	64.0	39.0	34.2	14.7	39.4	33.2	**7.2**	**14.0**	**8.1**	**12.2**	43.5	19.3	39.8	**7.9**	**5.2**	0.0	30.1	17.9	15.1	1.7
M-CONet [21]	C&L	55.7	20.4	60.6	36.1	29.0	13.0	38.4	33.8	4.7	3.0	2.2	5.9	41.5	20.5	35.1	0.8	2.3	**0.6**	26.0	18.7	15.7	1.4
Co-Occ [30]	C&L	56.6	24.4	72.0	43.5	42.5	10.2	35.1	**40.0**	6.4	4.4	3.3	8.8	41.2	**30.8**	40.8	1.6	3.3	0.4	**32.7**	**26.6**	**20.7**	1.1
OccMamba (Ours)	L	**59.9**	**25.0**	72.9	**44.8**	**42.7**	**18.1**	44.2	36.1	3.5	12.3	6.0	10.1	**44.6**	29.5	**42.1**	5.9	2.9	0.4	32.2	17.6	8.1	**22.1**

TABLE III: 3D occupancy results on SemanticKITTI [31] validation set.

Method	IoU (%)	mIoU (%)	Prec. (%)	Recall (%)	Params(M)	FLOPs(G)	Mem. (GB)
MLP/CNN-based							
Monoscene [5]	37.1	11.5	52.2	55.5	149.6	501.8	20.3
NDC-Scene [32]	37.2	12.7	-	-	-	-	20.1
Symphonies [8]	41.9	14.9	62.7	55.7	59.3	611.9	20.0
Transformer-based							
OccFormer [9]	36.5	13.5	47.3	60.4	81.4	889.0	21.0
VoxFormer [10]	57.7	18.4	69.9	76.7	57.8	-	15.2
TPVFormer [28]	35.6	11.4	-	-	48.8	946.0	20.0
CGFormer [33]	45.9	16.9	62.8	63.2	122.4	**314.5**	19.3
Mamba-based (Ours)							
OccMamba	**58.6**	**25.2**	**77.8**	70.5	**23.8**	505.1	**3.5**

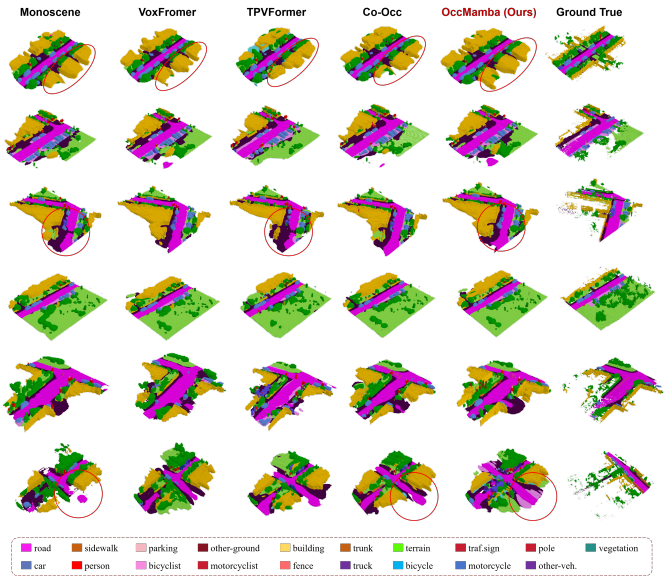

Fig. 5: Results of a qualitative comparison on the SemanticKITTI validation set are presented, showcasing various models.

3090 GPU. This dataset [31] focuses on semantic occupancy prediction with LiDAR point clouds and camera images. Specifically, the ground truth semantic occupancy is represented as the $[256, 256, 32]$ voxel grids. Each voxel is $[0.2m, 0.2m, 0.2m]$ large. OccMamba was trained for 80 epochs with a batch size of 6, using the AdamW optimizer [34] at an initial learning rate of 0.001, and input point cloud augmentation by random flipping along the $x - y$ axis.

Simulation Experiment: Our simulation experiments utilized a laptop with Ubuntu 20.04 and an NVIDIA RTX 4060 GPU. We crafted a dynamic $20m \times 20m \times 5m$ simulation environment, replete with stationary and moving obstacles to mimic occlusions and unknown regions (Fig 6). The AGR, equipped with OMEGA, successfully navigated this complex space, processing point clouds to avoid collisions.

Real-world Experiment: In our real-world experiments, OMEGA was operational on a customed AGR platform, illustrated in Fig. 2. The platform employed a RealSense D435i for point cloud acquisition, a T265 camera for visual-inertial odometry, and a Jetson Xavier NX for real-time onboard computation. OccMamba was optimized with TensorRT for efficient offline inference, guaranteeing smooth and responsive navigation. Our evaluation is around the following key questions:

RQ1: How does the efficiency of OccMamba compare to baseline models? (See § V-B)

RQ2: How effectively does OMEGA perform navigation tasks within dynamic simulation environments? (See § V-C)

RQ3: How proficiently does OMEGA handle navigation in five real-world dynamic scenarios? (See § V-D)

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

Quantitative Results: OccMamba has achieved state-of-the-art performance on the SemanticKITTI hidden test dataset, boasting a completion IoU of 59.9% and a mIoU of 25.0% (Table II). While recent methods (e.g., M-CONet [21] and Co-Occ [30]) have shown commendable performance in semantic occupancy prediction tasks, OccMamba outperforms the latter by a notable margin of 5.8% in IoU. This leap in accuracy is achieved solely by utilising point cloud data, without the need for other modalities inputs, simplifying integration and deployment in real-world robotic applications. In addition, OccMamba's efficiency parallels its performance; it integrates a linear-time mamba block within a compact, multi-branch network architecture and employs BEV feature fusion, culminating in a lean framework, weighing only 23.8MB. Meanwhile, OccMamba operates on a modest 3.5GB of GPU

TABLE IV: Ablation study on SemanticKITTI validation set.

Method	IoU ↑	mIoU ↑	Prec.	Recall	F1
OccMamba	58.6	25.2	77.8	70.5	73.9
w/o Geo-Mamba Block	58.2	24.7	76.4	69.8	72.8
w/o Sem-Mamba Block	57.8	24.1	76.1	69.5	72.5

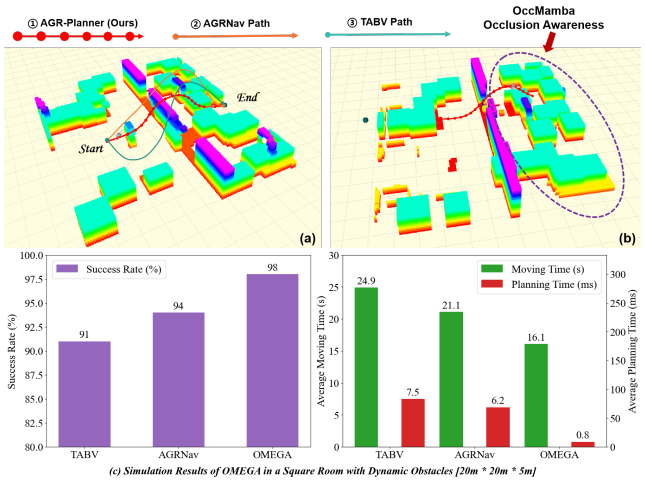

Fig. 6: Comparative Analysis of OMEGA with Baseline Systems AGRNav [4] and TABV [2]: Quantitative and Qualitative Insights in Simulation Environments.

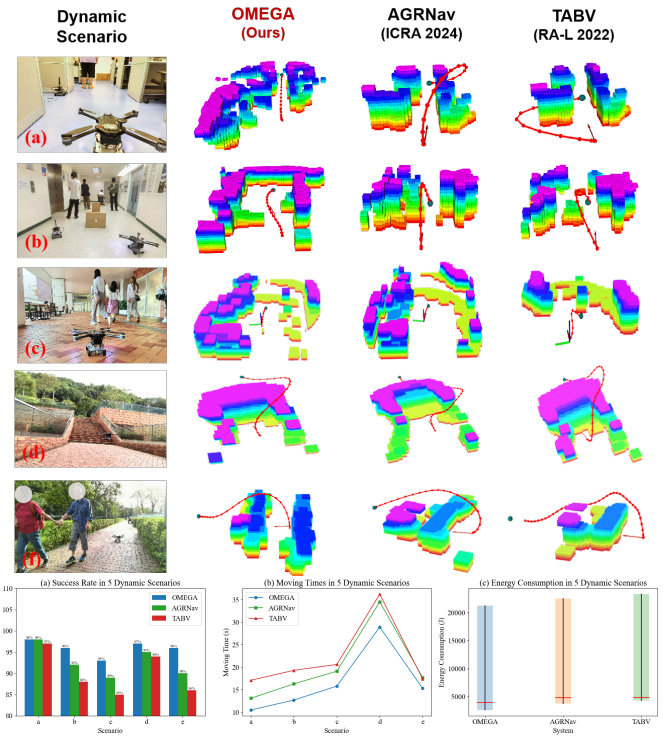

Fig. 7: Comparative Analysis of OMEGA with Baseline Systems AGRNav [4] and TABV [2]: Quantitative and Qualitative Insights in 5 real-world dynamic environments.

memory per training batch and impresses with an inference speed of 22.1 FPS (Table III), markedly outpacing existing methods like Monoscene [5] and TPVFormer [28] by a factor of 20.

Qualitative Results: In our visualizations on the SemanticKITTI validation set (Fig. 3), OccMamba showcases outstanding predictions of semantic occupancy in targeted occlusion areas. It exhibits exceptional performance, especially in complex categories such as "vegetation" and "terrain", consistent with the quantitative outcomes presented in Table II. These reliable predictions are crucial for subsequent path planning.

Ablation Study: The ablation experiments on the SemanticKITTI validation set (Table IV) highlight the importance of the Sem-Mamba and Geo-Mamba blocks in our network architecture. By separating semantic and geometric feature processing, our design promotes focused representation learning and leverages the synergistic effects of these aspects. Removing the Sem-Mamba block leads to a significant 4.37% decrease in mIoU, emphasizing its crucial role in precise semantic segmentation and nuanced classification of scene elements.

C. Simulated Air-Ground Robot Navigation

In a square room (Fig 6a), we conducted a comparative analysis of our OMEGA system, TABV [2] and AGRNav [1]. Through 100 trials with varied obstacle placements, we evaluated the average moving time, planning time, and success rate (i.e., collision-free) of each system (Fig. 6c). In dynamic environments, our system OMEGA demonstrates an exceptional planning success rate of 98% (Fig. 6c). This achievement stems not solely from OccMamba's real-time and precise mapping of occluded areas, which facilitates complete local map acquisition for planning, but also from the innovative design of our AGR-Planner, which operates without the need for an ESDF map. When benchmarked against two ESDF-reliant navigation baselines, our AGR-Planner reduces planning time by a substantial 89.33% (Fig. 6c). This synergy between OccMamba's detailed environmental perception and AGR-Planner's efficient path computation guarantees swift and reliable navigation for AGRs in complex, dynamically changing, and visually obstructed scenarios.

D. Real-world Air-Ground Robot Navigation

We demonstrated OMEGA's superior efficiency and energy conservation in 5 dynamic scenarios with a velocity cap of 1.5 m/s. Each navigation method was tested 10 times per scenario to ensure reliable performance comparison. We achieved a notable 96% average success rate (Fig. 7a) in 5 dynamic scenarios with OMEGA, exhibiting lower average energy consumption relative to competitors (Fig. 7c). Specifically, in scenarios B and C, OMEGA recorded energy reductions of 22.1% and 17.28%, respectively, against AGRNav. These improvements are largely due to our OccMamba module's swift computational ability to predict obstacle distributions in non-visible areas, enabling the AGR system to avoid potential impediments. When integrated with the AGR-Planner, this foresight allows for the rapid generation of multiple candidate paths, optimizing for both energy efficiency and reduced traversal time (Fig. 7b). Additionally, our AGR-Planner's

design includes penalty terms for aerial travel, inherently preferring less energy-intensive ground routes. These strategic advancements highlight OMEGA's effectiveness in navigating and conserving energy in dynamic environments.

VI. CONCLUSION

In this letter, we present the OMEGA, a cutting-edge advancement for air-ground robot (AGR) navigation in dynamic settings. The system is underpinned by the OccMamba module, which adeptly distinguishes between semantic prediction and scene completion, employing mamba blocks for swift and accurate real-time semantic occupancy mapping. This results in comprehensive local maps devoid of occlusions. Building on these refined maps, the AGR-Planner component of OMEGA formulates safe, smooth, and dynamically sound trajectories. Our rigorous testing indicates that OccMamba surpasses contemporary methods with a 59.9% IoU at 22.1 FPS. When integrated into the OMEGA system, the synergy between OccMamba and AGR-Planner culminates in a remarkable 96% average success rate for planning in actual dynamic scenarios.

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] 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)*. IEEE, 2023, pp. 1070–1077.

[4] 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," *arXiv preprint arXiv:2403.11607*, 2024.

[5] 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.

[6] 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.

[7] P. Tang, Z. Wang, G. Wang, J. Zheng, X. Ren, B. Feng, and C. Ma, "Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction," *arXiv preprint arXiv:2404.09502*, 2024.

[8] H. Jiang, T. Cheng, N. Gao, H. Zhang, W. Liu, and X. Wang, "Symphonize 3d semantic scene completion with contextual instance queries," *arXiv preprint arXiv:2306.15670*, 2023.

[9] Y. Zhang, Z. Zhu, and D. Du, "Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9433–9443.

[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] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[12] T. Dao and A. Gu, "Transformers are ssms: Generalized models and efficient algorithms through structured state space duality," *arXiv preprint arXiv:2405.21060*, 2024.

[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] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] X. Yang, H. Zou, X. Kong, T. Huang, Y. Liu, W. Li, F. Wen, and H. Zhang, "Semantic segmentation-assisted scene completion for lidar point clouds," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3555–3562.

[21] X. Wang, Z. Zhu, W. Xu, Y. Zhang, Y. Wei, X. Chi, Y. Ye, D. Du, J. Lu, and X. Wang, "Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 850–17 859.

[22] D. Liang, X. Zhou, X. Wang, X. Zhu, W. Xu, Z. Zou, X. Ye, and X. Bai, "Pointmamba: A simple state space model for point cloud analysis," *arXiv preprint arXiv:2402.10739*, 2024.

[23] K. Li, X. Li, Y. Wang, Y. He, Y. Wang, L. Wang, and Y. Qiao, "Videomamba: State space model for efficient video understanding," *arXiv preprint arXiv:2403.06977*, 2024.

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

[25] J. Mei, Y. Yang, M. Wang, T. Huang, X. Yang, and Y. Liu, "Ssc-rs: Elevate lidar semantic scene completion with representation separation and bev fusion," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1–8.

[26] 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.

[27] 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.

[28] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3d semantic occupancy prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9223–9232.

[29] 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.

[30] J. Pan, Z. Wang, and L. Wang, "Co-occ: Coupling explicit feature fusion with volume rendering regularization for multi-modal 3d semantic occupancy prediction," *IEEE Robotics and Automation Letters*, 2024.

[31] 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.

[32] J. Yao, C. Li, K. Sun, Y. Cai, H. Li, W. Ouyang, and H. Li, "Ndc-scene: Boost monocular 3d semantic scene completion in normalized device coordinates space," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2023, pp. 9421–9431.

[33] Z. Yu, R. Zhang, J. Ying, J. Yu, X. Hu, L. Luo, S. Cao, and H. Shen, "Context and geometry aware voxel transformer for semantic scene completion," *arXiv preprint arXiv:2405.13675*, 2024.

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