IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Rapid and Hierarchical UAV Exploration via Adaptive Regional Viewpoint Generation

Ningbo Bu, Gen Xu, Hao Zheng, Xuehang Wei, Wenshi Chen, Li Lv,
Xiaolu Zhang, Jiangjian Xiao, and Zhiqiang Li

***Abstract*—UAVs exploring large-scale environments rapidly and autonomously face significant challenges. Exploration efficiency is mainly limited by two factors: excessive long-distance back-tracking and frequent low-speed flight. To address the two issues, we propose a spatially hierarchical exploration method with fast regional viewpoint generation. The exploration area is partitioned online into subregions using an hgrid-based spatial decomposition, and the exploration sequence of these subregions is determined by solving a non-closed traveling salesman problem. Within each subregion, viewpoints are selected by optimizing a global frontier-related loss, yielding a set of globally consistent targets. This hierarchical partitioning and global path optimization reduce back-tracking distance. Meanwhile, the chosen viewpoints encourage smaller turns, facilitate obstacle avoidance, and improve coverage, which further decreases low-velocity motion and unnecessary revisits. In addition, we introduce a velocity-loss constraint to refine local trajectories and promote high-speed flight. We validate the proposed approach in both simulation and real-world experiments, where it consistently improves exploration efficiency over several state-of-the-art methods, especially in large-scale environments.**

***Index Terms*—Aerial Systems: Applications; Aerial Systems: Perception and Autonomy; Field Robots.**

I. INTRODUCTION

THE autonomous exploration of unmanned aerial vehicles (UAVs) is crucial for various applications, including inspection [1], 3D reconstruction [2], search [3], and rescue [4]. UAVs are capable of autonomously navigating and mapping unfamiliar environments, even in the presence of electromagnetic interference, without communication signals or posing risks to humans. Therefore, it is critical to create an effective and safe path planning algorithm to enable swift exploration in these environments.

Various exploration algorithms [5], [6] have been developed to tackle the task of autonomous exploration. Sampling-based methods [7] choose the viewpoint that offers the greatest information gain as the next point for exploration, whereas frontier-based methods [8] focus on frontiers that reduce traversal costs. However, many of these sampling and frontier-based techniques tend to get stuck in local optima and display greedy tendencies, leading to UAVs needing to backtrack. In large-scale environments, long-distance back-trackings can significantly hinder exploration efficiency.

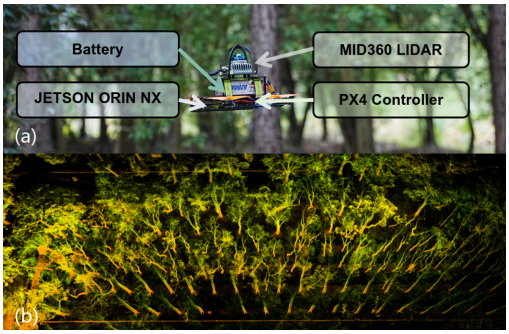

Fig. 1. (a) The quadrotor platform for exploration. (b) Visualized point cloud generated during exploration.

Although certain strategies [9], [10] utilize global optimization to minimize direction changes and address the back-tracking issue, they typically face high computational expenses and infrequent updates. This results in reduced flight-velocity and inconsistent motion planning. As a result, UAVs often operate at slower velocity, which adversely affects exploration efficiency, particularly in large-scale environments.

To tackle the issues of back-tracking and low velocity, we propose a spatial hierarchical exploration method, which optimizes the generation of regional viewpoints. This approach consists of three main steps: 1) Global planning: The exploration space is divided into several subregions using an online hgrid spatial decomposition, and the sequence for exploring these subregions is organized based on a non-closed traveling salesman problem with boundary constraints. This step aims to globally optimize the exploration route, minimizing UAV back-tracking and enhancing exploration efficiency. 2) Rapid regional viewpoint generation: Initially, numerous high-quality candidate viewpoints are quickly created by locating safe spaces near the exploration frontier within each subregion. The optimal viewpoints are then chosen from these candidates using a global loss function that maximizes both coverage and the UAV's velocity at the frontiers. 3) Local trajectory optimization: The trajectories are further refined by applying a velocity loss constrain to enable high-velocity flight.

Manuscript received: March 5, 2025; Revised May 5, 2025; Accepted August 11, 2025. This paper was recommended for publication by Editor G. Loianno upon evaluation of the Associate Editor and Reviewers' comments.(Corresponding author: Zhiqiang Li.)

This work was supported in part by the Zhejiang Province Science and Technology Plan Project under Grant 2024SSYS0091 and Ningbo Science and Technology Innovation 2025 Major Project under Grant 2025Z050.

Ningbo Bu, Gen Xu, Hao Zheng, Xuehang Wei, Wenshi Chen, Li Lv, Xiaolu Zhang, and Jiangjian Xiao are with Zhejiang Key Laboratory of Precision Actuation and Intelligent Robotics, Ningbo Institute of Materials Technology and Engineering of the Chinese Academy of Sciences, Ningbo 315211, China.

Zhiqiang Li is with Donghai Laboratory, Zhoushan 316021 China; He is also with Zhejiang Key Laboratory of Precision Actuation and Intelligent Robotics, Ningbo Institute of Materials Technology and Engineering of the Chinese Academy of Sciences, Ningbo 315211, China. lizhiqiangmy198267@163.com

Ningbo Bu is also with University of Chinese Academy of Sciences, Beijing 100049, China.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

The proposed method has been validated through simulations and real-world experiments. The experimental results indicate that it surpasses some current leading methods regarding flight velocity and exploration duration. Additionally, it also demonstrates resilience of our method in dynamic and intricate environments, highlighting its efficiency in various situations.

The primary contributions are summarized as follows:

- We introduce RUSH, a swift UAV exploration method designed for large-scale environments. It utilizes global optimization to avoid local optima and greedy.
- We design a regional viewpoint generation strategy, which fully utilizes the information of boundaries, the sensor coverage and the obstacle location to generate optimal viewpoints.
- We employ a velocity loss constrain that optimizes local paths for rapid flight while minimizing computational demands, thereby improving exploration efficiency.
- The proposed method integrates exploration frontier with regional partitioning to enable efficient global planning. It alleviates the long-distance back-tracking and the low-velocity flight of the UAV.

II. Related Work

The main components of autonomous exploration include modeling the environment, generating and selecting viewpoints, and planning motion. Of these, the generation and selection of viewpoints serve as the most straightforward measure of exploration autonomy [11], significantly impacting the overall efficiency of the exploration process. If the capabilities for generating and selecting viewpoints are lacking, quadrotor UAVs might end up revisiting the same areas during their exploration.

Frontier-based approaches [8] generally utilize a greedy method to choose the closest frontier to the UAV, allowing for quick access to the frontier. However, this direct use of the frontier as a viewpoint often overlooks safety issues. To improve UAV's exploration efficiency, researchers have proposed to select the frontier areas with safety constraints, which helps reduce velocity varying and supports sustained high-velocity flight [12]. Nonetheless, these approaches focus solely on the frontier's location and fail to consider a broader plan, making it challenging to meet more complex task demands.

Sampling-based viewpoint selection approaches [7] enhance the adaptability of exploration by integrating different gain functions into the viewpoint selection process, which enables more effective exploration. For example, trajectory length has been included as a component of the gain function [13], and information gain serves as a metric for assessing exploration advantages [14]. Nevertheless, these approaches still encounter spatial limitations and do not capture the relationships among multiple viewpoints, leading to considerable trajectory overlaps during global planning.

To address the aforementioned issues, recent studies have demonstrated that simple heuristic methods, such as FAEP [15], which combines boundary baselines [15] and heuristic viewpoint sampling [16], can significantly improve exploration performance. Also, some studies have modeled the viewpoint selection problem as a Traveling Salesman Problem (TSP) to optimize global paths [8]. They perform well in small-scale exploration scenarios. However, in larger-scale environments, it lacks larger-scale constraints, leading to long-distance back-tracking.

The global planners [9], [10] can significantly improve exploration efficiency. The overall framework of such methods includes both local and global planners, which work synergistically to fully explore the space. The local planner samples candidate viewpoints from the explored frontiers and selects the exploration position with the highest information gain, while the global planner performs global position expansion and path generation, or trajectory re-planning [13]. For example, a combination of greedy strategies and global optimization strategies can improve exploration efficiency and reduce computational time [17]. Compared to frontier-based methods, these methods require fewer computational resources and explores unknown space more quickly than sampling-based methods. However, they have a relatively low planning update frequency, which leads to slower flight velocity or inconsistent motion planning.

To address this issue, [18] employ a region decomposition method to divide the exploration space into some subregions and use the traversal sequence of these subregions as the global path, thus solving the problem of high computational resource consumption during large-area exploration. However, this method only considers the distance between viewpoints for obtaining traversal sequence, which may lead to traversing already explored subregions when switching between sug-regions. To overcome this, we propose a region-based viewpoint selection strategy that utilizes the directionality of the global path to select the optimal viewpoint in one subregion closer to the next subregion, thereby enabling UAVs fly at a constant velocity as high as possible. Additionally, we have improved the viewpoint generation scheme to rapidly produce safe and diverse candidate viewpoints, enhancing the quality of the optimal viewpoints.

III. The Proposed Method

This work presents a hierarchical exploration strategy that presents multi-level planning with a velocity-aware design. We couple globally optimized subregion ordering with local viewpoint planning and introduce a novel sampling method that accounts for velocity. To further enhance performance, a velocity-aware loss function is proposed, which further integrates velocity, acceleration, and jerk constraints to make fast and feasible flight. The whole system configuration and pipeline is illustrated in Fig. 2. Our work can be divided into three units, including hierarchical exploration strategy, optimal viewpoint generation, and flight velocity optimization. In the hierarchical exploration, we divide the entire exploration space into some subregions by an online hgrid spatial decomposition, and determine the optimal traversal order between these subregions by solving a non-closed traveling salesman problem. In the optimal viewpoint generation, the optimal viewpoints with maximum coverage and suitable for fast fight are selected from candidate viewpoints by a global loss function. In the flight

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

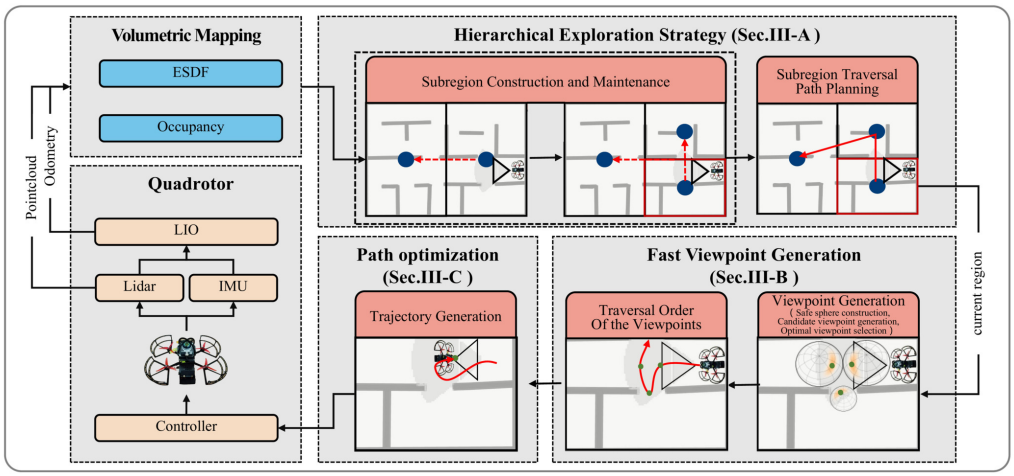

Fig. 2. The framework consists of three main components: hierarchical exploration strategy, optimal viewpoint generation, and flight velocity optimization. The system begins by decomposing the space into subregions using an online hgrid method and determining their traversal order via a non-closed traveling salesman formulation. Next, optimal viewpoints are selected based on global coverage and velocity-aware criteria. Finally, a velocity-aware loss function is introduced to produce fast and feasible trajectories. The entire pipeline is executed and updated continuously until exploration completes.

velocity optimization, a velocity loss constrain is designed to incorporate velocity, acceleration, and jerk constraints for enhancing flight velocity.

A. Hierarchical Exploration Strategy

Most existing global guidance methods only visit the frontier areas and overlook unknown areas beyond, resulting in frequent long-distance back-tracking [19]. To address this problem, we propose a hierarchical exploration strategy, which adopts global planning over the entire exploration space for reducing the frequency of the long-distance back-tracking. It consists of three steps.Firstly, we use a hierarchical grid-based decomposition [20] to divide the entire exploration space into multiple subregions with different resolutions. These subregions are represented by $\{A_1, A_2, \ldots, A_N\}$. Then, we determine the traversal order of these subregions by solving a Traveling Salesman Problem (TSP) [21] under the constraints of reducing back-tracking. Finally, within each subregion, we employ the FUEL method [8] to perform greedy and rapid exploration using incremental frontier structures.

1) Subregion Construction and Maintenance: We construct an iterative decomposition strategy that partitions the environment into multiple subregions. Specifically, we propose a longest-side-first splitting policy: each subregion is recursively generated by dividing its exploration space along the longest axis. The division process is constrained by a predefined decomposition depth of balancing computational efficiency and memory usage. If a subregion is square, we default to splitting along the x-axis. Compared to uniform-size partitioning [18], it adapts to spatial irregularities, avoids generating excessively elongated regions, facilitates the construction of coherent and efficient flight trajectories, and yields a more balanced structure which simplifies subsequent maintenance and state tracking.

Each subregion is assigned a unique identifier and organized within a hierarchical tree structure, which encodes spatial containment relationships. To further support global path planning based on the Traveling Salesman Problem (TSP), we maintain the visiting order of all subregions using a linked list. The order of nodes in the list corresponds to the planned traversal sequence, allowing efficient dynamic insertion, deletion, and reordering operations. This design ensures both consistency and flexibility in region management.

The system continuously identifies and marks the subregion currently occupied by the UAV as "active". Subregions that have been fully explored are marked as "completed" and removed from the list, reducing redundant computation and planning overhead.

2) Subregion Traversal Path Planning: To effectively plan the traversal paths among subregions, we construct a comprehensive cost matrix C_{tsp} that encodes the travel costs between every pair of subregions. Formally, the cost matrix $C_{\text{tsp}} \in \mathbb{R}^{n\times n}$ is defined as:

$$C_{\text{tsp}} = \begin{bmatrix} 0 & C_{1,2} & \cdots & C_{1,n} \\ C_{2,1} & 0 & \cdots & C_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n,1} & C_{n,2} & \cdots & 0 \end{bmatrix} \tag{1}$$

where $C_{i,j}$ represent the visit cost between subregion A_i and subregion A_j. Note that in non-closed TSP, the path does not require returning to the starting point, so the diagonal elements of the matrix are zero. $C_{i,j}$ is defined as:

$$C_{i,j} = \begin{cases} \lambda \cdot C_{path}(A_i, A_j) + \beta \cdot C_{vel}(A_i, A_j) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{2}$$

where λ and β are weight coefficients for adjustment, and $C_{path}(A_i, A_j)$ and $C_{vel}(A_i, A_j)$ are the path cost and velocity-related cost between subregion A_i and subregion A_j, respectively. $C_{path}(A_i, A_j)$ is defined as:

$$C_{path}(A_i, A_j) = \text{AstarPathLength}(A_i, A_j) \tag{3}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

which represents the length of the path between two subregion centers, and is calculated by A* algorithm [22]. C_{vel} is defined as:

$$C_{vel}(A_i, A_j) = \frac{\Theta(A_i, A_j)_{\text{sum}}}{R(A_i, A_j)} \tag{4}$$

where $\Theta_{\text{sum}}(A_i, A_j)$ represent the sum of the absolute values of all turning angles along the path. The larger the turning angle, the lower the flight velocity of the UAV is usually. The curvature radius of a path describes the degree of its bending, and $R(A_i, A_j)$ represents the minimum curvature radius of a path. The smaller the curvature radius, the lower the flight velocity of the UAV is.

Finally, we employ a Traveling Salesman Problem (TSP) to determine the ordering of traveling subregions. Specifically, each subregion A_i is associated with an exploration state variable $e_i \in \{0, 1\}$, indicating whether it still needs to be visited.

$$e_i = \begin{cases} 1, & \text{if subregion } A_i \text{ needs to be explored} \\ 0, & \text{if subregion } A_i \text{ has been fully explored} \end{cases} \tag{5}$$

The objective function is to minimize the total path cost, represented as:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} C_{ij} \cdot x_{ij} \\ \text{s.t.} \quad & \sum_{j=1, j\neq i}^{n} x_{ij} \leq e_i, \quad \forall i \in \{1, \ldots, n\} \\ & \sum_{i=1, i\neq j}^{n} x_{ij} \leq e_j, \quad \forall j \in \{1, \ldots, n\} \\ & u_i - u_j + n x_{ij} \leq n - 1, \\ & \quad \forall i \neq j,\ i, j \in \{2, \ldots, n\} \end{aligned} \tag{6}$$

Where $x_{ij} \in \{0, 1\}$ represent binary decision variable, $x_{ij} = 1$ denotes that the robot travels directly from subregion A_i to A_j, and C_{ij} is the associated travel cost.We adopt the Miller–Tucker–Zemlin (MTZ) subtour elimination constraints [23] to ensure that each subregion is visited once. Specifically, we introduce integer auxiliary variables $u_i \in [2, n]$ to represent the relative position of subregion A_i in the visiting.

B. Fast Viewpoint Generation

The overview of generating viewpoints is briefly described as follows. First, we construct an active voxel grid map in the subregion. Each voxel can be in one of three states: free, occupied, or unknown. We then identify free voxels that are directly adjacent to unknown voxels (yellow grids in Fig. 3) as frontier voxels. When a group of connected frontier voxels exceeds a threshold, we cluster them and define the cluster region as a frontier C_i. Next, a safe sphere is constructed. It's diameter is the maximum length of the frontier and there is no obstacles in the sphere. The candidate viewpoints are sampled in the sphere by a polar-coordinate Gaussian distribution. Each candidate viewpoint is quantitatively evaluated along three dimensions: its frontier coverage including the number of observable frontier voxels and the estimated amount of unknown space it can obtain; its spatial relationship to the current UAV position including both distance and direction; and its consistency with the UAV's current yaw orientation. Finally, a heuristic function is used to select the optimal viewpoint for each frontier, and only the optimal viewpoint is used for the trajectory planning. The details are as follows:

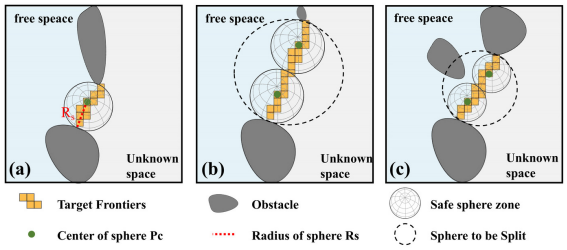

Fig. 3. This figure shows the process of generating safe spheres. When the diameter of the sphere is smaller than the sensor's detection distance and there are no obstacles in the sphere, it corresponds to the scenario (a). When the diameter of the sphere exceeds the sensor's detection distance, it is divided into two spheres, as shown in (b). When there are obstacles inside the sphere, it is also divided into two spheres, as shown in (c).

1) Safe sphere Construction: Inspired by the bubble-based exploration strategy [17], we develop a simple yet effective method for generating safe spheres around frontiers. The procedure is illustrated in Fig. 3.

Then, we generate a safe sphere around each frontier. Specifically, the center $\mathbf{p}_c$ of the sphere is defined as the geometric center of the axis-aligned bounding box (AABB) [8] enclosing the frontier. The radius R_s of the sphere is set to half of the maximum distance between any two frontier voxels within the cluster, which can be formulated as:

$$R_s = \min\left(\min_{i=1,\ldots,N} \|\mathbf{p}_i - \mathbf{p}_c\|_2, \frac{\text{Length}(C_i)}{2}, L_s\right) \tag{7}$$

If the sphere encounters the following situations, the frontier needs to be split: 1) the diameter of the sphere exceeds the sensor's detection distance L_s, or 2) there are obstacles $\mathbf{p}_i \in \mathbb{R}^3$ in the sphere. The splitting process is as follows: split the frontier into two frontiers with the same length, and generate two spheres. This process is repeated until all spheres don't meet the conditions 1) and 2). Additionally, stop splitting when the radius of the sphere is less than 2 meters, even if there are still obstacles inside. There is no need to split the sphere into smaller ones, as the UAV can quickly observe both sides of the obstacles.

2) Candidate Viewpoint Generation: As shown in Fig. 4, we generate candidate viewpoints on a sphere by polar coordinates with Gaussian sampling. A candidate viewpoint $\mathbf{v}_i$ is defined as:

$$\begin{aligned} \mathbf{v}_i &= \mathbf{p}_c + d_s \cdot \begin{bmatrix} \sin(\phi + \phi_f)\cos(\theta + \theta_f) \\ \sin(\phi + \phi_f)\sin(\theta + \theta_f) \\ \cos(\phi + \phi_f) \end{bmatrix} \\ d_s &\sim \mathcal{N}(\frac{R_s}{2}, \sigma_1^2), \theta \sim \mathcal{N}(0, \sigma_2^2), \phi \sim \mathcal{N}(0, \sigma_3^2) \end{aligned} \tag{8}$$

Where $\mathbf{p}_c$ is the center of the sphere, d_s denote the distance from the viewpoint to the center, θ is the azimuth angle , and

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

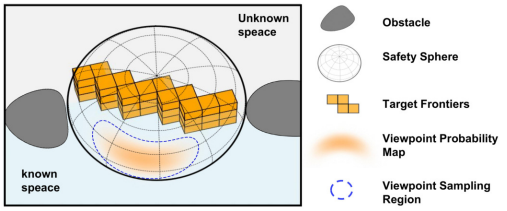

Fig. 4. The region circled by blue dashed line represents the range for viewpoint generation, which are determined by Eq.(10) and lies in the known region between the safe sphere and the target frontier. Within the region, viewpoints are generated using the polar coordinate Gaussian sampling. As the yellow in the region deepens, the probability of generating candidate viewpoints increases.

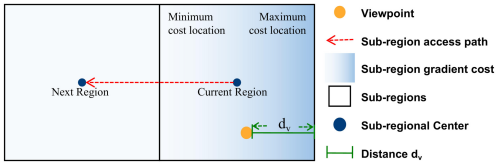

Fig. 5. The schematic diagram of gradient-guided loss distribution. The gradient-guided loss gradually decreases along the direction from the current subregion to the next subregion.

ϕ is elevation angle. θ_f and ϕ_f are the azimuth and elevation angles pointing to the frontier F, ensuring the candidate viewpoints are directed toward potentially unexplored regions. Unlike traditional methods that require heuristic safety evaluations, our approach samples directly from a "safe sphere" centered at the current position. As a result, the candidate viewpoints V are rapidly generated, as illustrated in Fig. 4.

3) Optimal Viewpoint Selection: The optimal viewpoints can be determined by two criteria: 1) maximizing the coverage C_f of the frontier and 2) facilitating high-velocity flight of the UAV.

The frontier coverage is computed by counting unknown voxels hit by truncated rays from the viewpoint within the sensor's FoV. For the second criterion, we proposed a velocity cost C_v, which is defined as:

$$C_v = \lambda_y \cdot (1 - \cos(\theta)) + e^{-\lambda_d \cdot D_o} \tag{9}$$

where λ_y and λ_d are two balancing factors, and D_o is the distance from the viewpoint to the nearest obstacle. Empirically, we set λ_y and λ_d to 1. The first term of the equation is to minimize the turning angle of the UAV during flight, while the second term ensures the turning radius is as large as possible.

Finally, the overall cost for each viewpoint is calculated as:

$$C(V_{p_i}) = \lambda_{cv} \cdot C_v + \lambda_{cf} \cdot C_f \tag{10}$$

The viewpoint with the lowest cost are chosen as the best observation point, effectively improving the overall efficiency. Additionally, the cost function combines the distance D_o from the viewpoint to the nearest obstacle, reducing collision risks for high-velocity flight. Furthermore, by dynamically adjusting viewpoints in real-time based on the flight status and environmental information, the UAV can maintain navigation performance in the challenge environment.

4) Traversal Order Of The Viewpoints: After generating the viewpoints, we utilize the subregion visitation order obtained in Sec.III-A to determine the direction from the current subregion to the next one. Along this direction, we construct a gradient-based loss function and use the TSP solver to determine the optimal traversal order of the viewpoints. The distance from the viewpoint to the boundary of the region in the opposite direction along this path is denoted as d_v, as illustrated in Fig. 5. A differentiable potential cost function is formulated along this direction. The position of the viewpoint within the subregion contributes to a reverse gradient loss, denoted as:

$$C_e = \frac{1}{d_v^2} \tag{11}$$

After obtaining the visiting order, we can determine the direction from the current subregion to the next subregion. Along this direction, we establish a gradient-guided loss within single subregion, namely:

$$C_d = \lambda_e C_e + \lambda_p C_p \tag{12}$$

where λ_e and λ_p are adjustment factors. C_p denotes the path length between viewpoints calculated by the A* algorithm [22], aiming to minimize the overall cost. As shown in Fig. 5, along the direction from the current subregion to the next subregion, the guided loss gradually decreases.

Once obtaining the gradient-guided cost of each viewpoint, the traveling order of viewpoints can be optimized by solving a Traveling Salesman Problem (TSP) [24].

C. Path Optimization by a Velocity Loss Constraint

After determining the visiting order of viewpoints, we compute a UAV trajectory that supports high average speed. The trajectory generation follows the unified two-stage framework in [25], consisting of path searching and subsequent optimization.

In the searching stage, we employ the hybrid-state A* algorithm [25] to generate an initial dynamically feasible trajectory. The optimization stage formulates the trajectory as a uniform B-spline, defined by its degree p_b, a set of control points $\{Q_1, Q_2, \ldots, Q_N\} \subset \mathbb{R}^3$, and a knot vector $\{t_1, t_2, \ldots, t_M\} \subset \mathbb{R}$ with $M = N + p_b$. For computational efficiency, the knots are uniformly spaced with constant interval $\Delta t = t_{m+1} - t_m$. The objective function is defined as follows:

$$\begin{aligned} \min_{\mathcal{Q}} \; J = \lambda_v & \sum_{i=1}^{N-1} \left(\max\left(0, v_{\max} - \|V_i\|\right)\right)^2 \\ & + \lambda_{\mathrm{f}} \sum_{i=1}^{N-1} \left(\max\left(0, \|V_i\| - v_{\max}\right)\right)^2 \\ & + \lambda_{\mathrm{f}} \sum_{i=1}^{N-2} \left(\max\left(0, \|A_i\| - a_{\max}\right)\right)^2 \\ & + \lambda_j \sum_{i=1}^{N-3} \|J_i\|^2 \end{aligned} \tag{13}$$

where λ_v, λ_f, and λ_j are weighting coefficients, and $v_{\max}$, $a_{\max}$ denote the maximum allowable velocity and acceleration,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

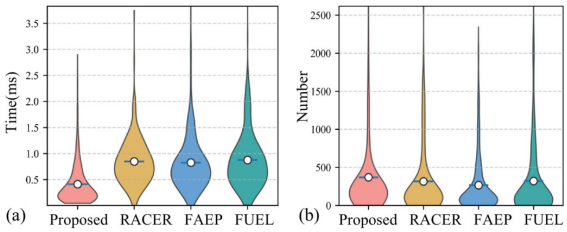

Fig. 6. Generation time distribution and coverage of candidate viewpoints. The violin plot illustrates the data distribution, with width representing density and the blue line indicating the mean position. (a) The figure illustrates the distribution and mean of viewpoint generation time. (b) The figure shows the distribution and mean of unexplored voxels covered by per viewpoint.

TABLE I
COMPARATIVE ANALYSIS OF VIEWPOINT GENERATION PERFORMANCE

Methods	Time (ms)	Coverage	Valid Viewpoints
Proposed	0.412	370.09	23.97
RACER	0.848	315.80	21.12
FAEP	0.826	265.36	23.27
FUEL	0.874	319.83	24.27

respectively. The velocity V_i, acceleration A_i, and jerk J_i are defined as follows:

$$V_i = \frac{Q_{i+1} - Q_i}{\Delta t}, \quad A_i = \frac{V_{i+1} - V_i}{\Delta t}, \quad J_i = \frac{A_{i+1} - A_i}{\Delta t} \tag{14}$$

The objective function comprises three components: a velocity incentive term that encourages high-speed flight, a feasibility term that penalizes violations of velocity and acceleration limits, and a smoothness term that minimizes jerk to ensure dynamic consistency and trajectory continuity.

IV. Experimental Results

A. Comparison of Candidate Viewpoint Sampling Strategies

To validate the effectiveness of our viewpoint sampling strategy, we conducted a comparative experiment in a typical structured indoor environment which is multi-room connectivity and has dynamic obstacle distribution for comparing three viewpoint sampling strategies: A comparison is conducted between our proposed strategy and the implementations of the RACER, FAEP, and FUEL frameworks. The structured indoor environment used for evaluation, as depicted in Fig. 9(a), has a volume of approximately 900 m^3.

Experimental results demonstrate that the proposed method significantly improves computational efficiency. As shown in Fig. 6(a) and Table I, the average time of generating a viewpoint is only 48.6% of that of RACER, 49.8% of FAEP, and 47.1% of FUEL.

As shown in Fig. 6(b) and Table I, it also outperforms RACER, FAEP, and FUEL in coverage by 17.2%, 39.4%, and 15.7%, respectively, demonstrating its superior observation capability.

B. Ablation Study on the Viewpoint Cost Function

We conducted an ablation study on the proposed viewpoint selection cost function (Eq. 10). In the experiments, the parameter λ_{cf} was fixed at 1, while λ_{cv} was varied from 0 to 50. When $\lambda_{cv} = 0$, viewpoint selection relies solely on the number of observable frontiers, consistent with the strategies employed by RACER, FAEP, and FUEL. As λ_{cv} increases, the effect of our proposed cost function becomes increasingly apparent.

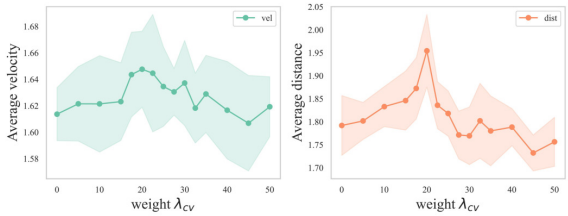

Fig. 7. The impact on the average flight velocity with different weight values in the proposed cost function. The impact of different weights in the proposed cost function on (a) the average flight velocity and (b) the corresponding average minimum distance to obstacles along the entire trajectory. The shaded areas represent the variation range across multiple trials.

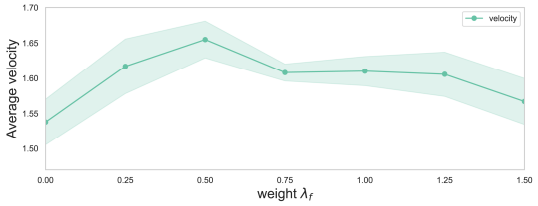

Fig. 8. The impact of different values of the velocity incentive weight λ_v on the average flight velocity. The shaded area is the standard deviation across multiple trials.

The results, illustrated in Fig. 7, demonstrate the effectiveness of the heuristic term. Specifically, increasing λ_{cv} leads to noticeable improvements in both the average flight velocity and the mean distance to obstacles along the trajectory, with the optimal performance within the range of $\lambda_{cv} = 15$ to 25.

C. Ablation Study on the Velocity Loss Constraint

We conducted an ablation study on the velocity loss term in the trajectory optimization objective (Eq. (14)). In this experiment, the weights of the feasibility term ($\lambda_f = 5$) and the smoothness term ($\lambda_j = 10$) were fixed to match the settings used in the FUEL paper, while the velocity incentive weight λ_v was varied from 0 to 1.5. As illustrated in Fig. 8, increasing λ_v significantly boosts the UAV's average flight speed, indicating the effectiveness of the velocity incentive in enhancing high-speed flight. However, when λ_v exceeds 0.5, the performance begins to decline gradually, meaning the existence of an optimal range for this parameter.

D. Comparative Experiments in Simulated Environments

In this section, we provide a comprehensive comparison of our method, RACER [18], FUEL [8] and FAEP [15] in three simulated environments including a classic maze scenario [8], an outdoor forest scenario [17], and a large maze scenario. These scenarios are illustrated in Fig. 9. All methods were evaluated on a high-performance computer equipped with an Intel Core i9-12900K@5.20GHz processor operating.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

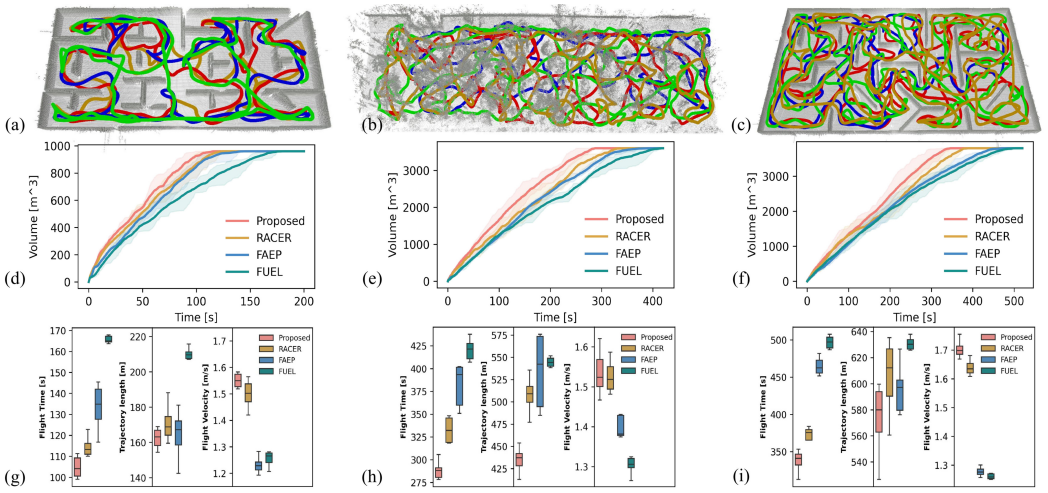

Fig. 9. Simulation comparison on different scenarios. (a), (b), and (c) correspond to the maze, outdoor scenes and large maze, respectively. The red, yellow, blue, and green trajectories represent the flight paths of our method,RACER, FAEP, and FUEL, respectively. (d), (e), and (f) show the exploration volume over time. (g), (h), and (i) show the specific distribution of flight time, trajectory length, and flight velocity.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT METHODS

Scene	Method	Exploration Time (s)			Trajectory Length (m)			Exploration Velocity (m/s)		
		Ave	Max	Min	Ave	Max	Min	Ave	Max	Min
Maze	Proposed	**104.79**	111.24	99.04	**162.38**	168.93	154.40	**1.55**	1.58	1.52
	RACER	114.44	122.81	110.06	170.78	188.18	159.48	1.50	1.56	1.42
	FAEP	133.82	145.44	116.76	165.01	181.04	142.47	1.23	1.28	1.19
	FUEL	167.50	178.90	163.72	210.12	215.81	207.03	1.26	1.28	1.21
Outdoor Forest	Proposed	**285.93**	305.76	262.82	**438.07**	469.21	412.90	**1.53**	1.62	1.47
	RACER	332.74	348.27	318.20	507.71	535.72	476.94	1.52	1.59	1.48
	FAEP	381.57	402.32	350.91	534.15	575.87	484.77	1.40	1.43	1.37
	FUEL	420.46	437.70	407.19	548.39	578.38	538.29	1.30	1.32	1.26
Large Maze	Proposed	**337.76**	353.31	312.65	**572.67**	599.64	526.59	**1.69**	1.76	1.60
	RACER	378.34	383.87	373.23	619.89	635.38	600.46	1.64	1.68	1.61
	FAEP	464.76	481.79	451.79	595.26	626.64	576.21	1.28	1.33	1.26
	FUEL	496.89	507.68	486.80	628.34	637.96	611.14	1.27	1.30	1.25

Fig. 9(a)-(c) shows the flight trajectories of different methods under various scenario. In Fig. 9(d)-(f), semi-transparent shaded regions delineate the upper and lower bounds from repeated trials, while solid trajectories represent the mean values. Fig. 9(g)-(i) statistically presents the distribution characteristics of flight time, trajectory length, and flight velocity in these scenarios. It is evident that the FUEL method performs poorly, with the longest flight time and longest trajectory length, and significant fluctuations in velocity, failing to avoid long-distance back-tracking flights. The FAEP method often segments the interior into multiple regions, leading to frequent round trips. Although the RACER method outperforms both FAEP and FUEL, its performance is still inferior to our method.

Table II presents the exploration performance of all methods. Overall, our method has higher exploration efficiency across all scenes. Specifically, the proposed method not only has a shorter exploration time than those of the RACER, FAEP and FUEL methods, but also maintains a higher velocity and relatively shorter trajectory length.

E. Experiments in Real-world Environments

We also conducted experiments in real-world scenarios including a forest, an underground parking garage, and an indoor warehouse, representing outdoor cluttered, indoor structured, and semi-open environments, respectively. The UAV

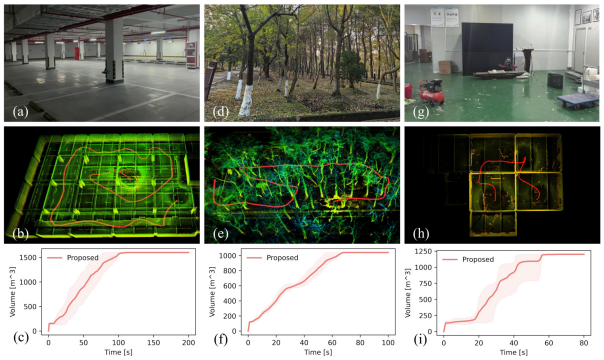

Fig. 10. The real-world experiments are made in (a), (d) and (g), corresponding to the forest, underground parking and warehouse scenes, respectively. (b) , (e) and (h) display the flight trajectory. (c) , (f) and (i) illustrate the exploration performance in these two scenarios.

TABLE III
RUNTIME OF THE PROPOSED METHOD

Scene	Hierarchical Exploration	Viewpoint
Garage	49ms	65ms
Forest	48ms	52ms
Warehouse	43ms	35ms

was required to autonomously localize and map using onboard LiDAR and IMU data without external positioning systems (such as GPS), and to explore all unknown environments. Our hardware platform, shown in Fig. 1, uses PX4 as the low-level control system, an NVIDIA NX onboard computing platform, and a five-inch propeller-powered system. MID360 is used as the perception sensor for the real-world experiments. The UAV's maximum flight velocity was set to 2.0 m/s, and the maximum acceleration to 2.0 m/s^2.

The underground parking garage is approximately $30 \times 26 \times 2\,\mathrm{m}^3$, with walls and numerous stone pillars, along with a large number of water pipes, lights, and electrical wires on the ceiling. The UAV with our method finished the exploration in 102.03 seconds. The trajectory length was 148.42 meters, and

the average flight velocity was 1.45 m/s.

The complex forest is approximately $40 \times 13 \times 2\,\mathrm{m}^3$, with high vegetation density and uneven terrain. Obstacles include trees of various heights, shrubs, and randomly distributed rocks. The UAV completed the exploration task in 70.31 seconds. The flight path length was 101.1 meters, with an average speed of 1.44 m/s. Fig. 10 presents the detailed results.

The indoor warehouse consists of four connected rooms, each approximately $20 \times 20 \times 3\,\mathrm{m}^3$. The UAV completed the exploration in this structured space with a trajectory length of 41.0 meters and an average speed of $0.8\,\mathrm{m/s}$.

We analyzed the average running time in real experiments as shown in Table III. Hierarchical Exploration refers to the runtime of the Hierarchical Exploration Strategy (Sec. III-A), including Subregion Construction and Maintenance and Subregion Traversal Path Planning, while Viewpoint denotes the runtime of Fast Viewpoint Generation (Sec. III-B), covering Viewpoint Generation and Traversal Order of the Viewpoints.

The results demonstrate that the proposed method avoids redundant paths and has robust performance across diverse environments.

V. Conclusion

This paper proposes a spatial hierarchical exploration method that rapidly generates optimal viewpoints and optimizes flight velocity, aiming to enhance autonomous exploration efficiency in large-scale environments. It comprises three key stpng: global planning, rapid regional viewpoint generation and local trajectory optimization. The global planning divides the exploration space into subregions using an online hgrid spatial decomposition, and determining the order of exploration for these subregions by solving a non-closed traveling salesman problem. The rapid regional viewpoint generation chooses optimal viewpoints from these subregions based on a global loss function related to frontiers. The path optimization uses a velocity loss constrain to improve local trajectory. Our method can reduce the UAV's back-tracking and enable UAVs to maintain maximum flight velocity during exploration, minimize time spent in large-scale environments, and improve exploration efficiency. Experiments conducted in both simulated and real-world environments demonstrate the effectiveness of our method.

In the future work, we will focus on the UAVs exploration of dynamic scenarlos, including motion obstacles and structured environments. Hierarchical decomposition in structured environments should be further improved by adding layout constraints.

References

[1] Z. Xu, B. Chen, X. Zhan, Y. Xiu, C. Suzuki, and K. Shimada, "A vision-based autonomous uav inspection framework for unknown tunnel construction sites with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4983–4990, 2023.

[2] S. H. Gazani, M. Tucsok, I. Mantegh, and H. Najjaran, "Bag of views: An appearance-based approach to next-best-view planning for 3d reconstruction," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 295–302, 2023.

[3] M. Petrlík, P. Petráček, V. Krátký, T. Musil, Y. Stasinchuk, M. Vrba, T. Báča, D. Hert, M. Pecka, T. Svoboda, and M. Saska, "Uavs beneath the surface: Cooperative autonomy for subterranean search and rescue in darpa subt," *Field Robotics*, vol. 3, pp. 1–68, 01 2023.

[4] J. Xu, X. Fan, H. Jian, C. Xu, W. Bei, Q. Ge, and T. Zhao, "Yoloow: A spatial scale adaptive real-time object detection neural network for open water search and rescue from uav aerial imagery," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[5] K. Naderi, J. Rajamäki, and P. Hämäläinen, "Rt-rrt* a real-time path planning algorithm based on rrt," in *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, 2015, pp. 113–118.

[6] J. Huang, B. Zhou, Z. Fan, Y. Zhu, Y. Jie, L. Li, and H. Cheng, "Fael: Fast autonomous exploration for large-scale environments with a mobile robot," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1667–1674, 2023.

[7] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.

[8] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.

[9] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, 2020, wiley Online Library.

[10] C. Cao, H. Zhu, H. Choset, and J. Zhang, "Tare: A hierarchical framework for efficiently exploring complex 3d environments." in *Robotics: Science and Systems*, vol. 5, 2021, p. 2.

[11] Y. Hu, J. Geng, C. Wang, J. Keller, and S. Scherer, "Off-policy evaluation with online adaptation for robot exploration in challenging environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3780–3787, 2023.

[12] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2135–2142.

[13] X. Chen, J. Zheng, and Q. Hu, "A hybrid planning method for 3d autonomous exploration in unknown environments with a uav," *IEEE Transactions on Automation Science and Engineering*, 2023.

[14] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.

[15] Y. Zhao, L. Yan, H. Xie, J. Dai, and P. Wei, "Autonomous exploration method for fast unknown environment mapping by using uav equipped with limited fov sensor," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 5, pp. 4933–4943, 2024.

[16] J. Yu, H. Shen, J. Xu, and T. Zhang, "Echo: An efficient heuristic viewpoint determination method on frontier-based autonomous exploration for quadrotors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5047–5054, 2023.

[17] B. Tang, Y. Ren, F. Zhu, R. He, S. Liang, F. Kong, and F. Zhang, "Bubble explorer: Fast uav exploration in large-scale and cluttered 3d-environments using occlusion-free spheres," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1118–1125.

[18] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-uav system," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1816–1835, 2023.

[19] Y. Zhang, X. Chen, C. Feng, B. Zhou, and S. Shen, "Falcon: Fast autonomous aerial exploration using coverage path guidance," *IEEE Transactions on Robotics*, 2024.

[20] X. Kan, H. Teng, and K. Karydis, "Online exploration and coverage planning in unknown obstacle-cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5969–5976, 2020.

[21] G. Pataki, "Teaching integer programming formulations using the traveling salesman problem," *SIAM review*, vol. 45, no. 1, pp. 116–123, 2003.

[22] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[23] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.

[24] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations research*, vol. 21, no. 2, pp. 498–516, 1973.

[25] 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.