

Supplementary Materials for Autonomous Exploration with Terrestrial-Aerial Bimodal Vehicles

Yuman Gao*, Ruibin Zhang*, Tiancheng Lai, Yanjun Cao, Chao Xu, Fei Gao[†]

1 Bimodal Viewpoint Generation

In this section, we present the bimodal viewpoint generation pipeline and describe a greedy algorithm that exploits submodularity to efficiently select informative viewpoints.

To efficiently generate viewpoints covering frontiers, we first group the frontier voxels into clusters. Frontier cells are clustered into frontier clusters based on their 3D spatial distribution. For a frontier cluster C , we consider two strategies for selecting viewpoints to achieve full coverage:

- i) **Aerial-Modal Strategy (AS)**: The cluster is covered exclusively using aerial viewpoints.
- ii) **Hybrid-Modal Strategy (HS)**: The cluster is first covered using terrestrial viewpoints; if full coverage cannot be achieved, additional aerial viewpoints are selected to complete the coverage.

The AS and HS sets provide diverse modality-specific candidate viewpoints, which serve as informative inputs for subsequent decision-making in BM-MCTS. The detailed procedure is presented in Alg.1 and illustrated in Fig.1.

Specifically, for each frontier cluster C , we first sample raw candidate terrestrial and aerial viewpoints (Lines 1–2). As shown in Fig.1(a), raw aerial viewpoints are generated using cylindrical coordinate sampling within a distance range $[d_{\min}, d_{\max}]$ and azimuth angle φ around the cluster’s normal, across multiple heights along the z -axis. The frontier normal computed via Principal Component Analysis (PCA) and oriented toward the known space. Constraining the sampling range in this manner improves efficiency without sacrificing coverage quality. While, raw terrestrial viewpoints are generated by selecting ground voxels from the nearby grid map and applying a fixed vertical offset. To ensure spatial diversity, a minimum distance is enforced between adjacent terrestrial viewpoints. Importantly, each

*Equal contribution. All authors are with the Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute, Zhejiang University, Huzhou 313000, China.

[†]Corresponding author: Fei Gao. Emails: {ymgao, fgaoaa}@zju.edu.cn

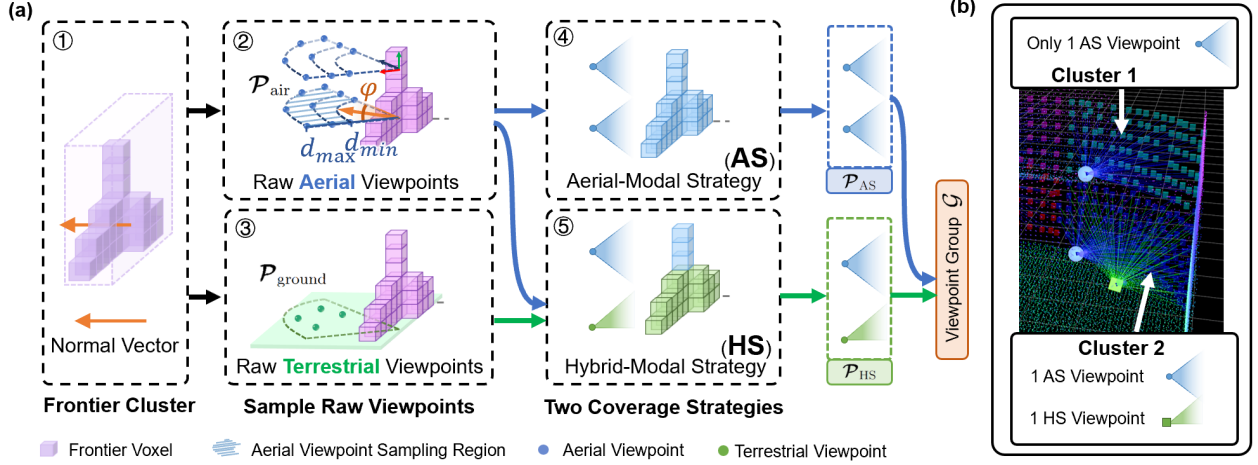


Figure 1: (a): The pipeline of bimodal viewpoint generation with two coverage strategies. ①: A frontier cluster C with a normal computed via PCA and oriented toward the known space. ②-③: Raw aerial viewpoints $\mathcal{P}_{\text{aerial}}$ and raw terrestrial viewpoints $\mathcal{P}_{\text{ground}}$ generation. ④-⑤: Two strategies to select bimodal viewpoints to fully cover C , resulting \mathcal{P}_{AS} and \mathcal{P}_{HS} . (b). An example of AS and HS viewpoints generation in the simulation.

viewpoint is defined as $\mathcal{P}_i = (\mathbf{p}, \phi)$, where \mathbf{p} is the position and ϕ is the yaw angle. The information gain (IG) of \mathcal{P}_i is defined as the number of visible frontier voxels within \mathcal{P}_i 's Field of View (FoV). The yaw angle ϕ is determined as the one maximizing IG, by using a yaw optimization method similar to [1].

Given these raw viewpoints, we then select the viewpoint to fully cover C via two strategies: AS and HS. For AS, which covers the cluster using only aerial viewpoints, in each iteration, the viewpoint with the highest newly gained information ΔIG is selected, where ΔIG is defined as the number of previously unseen frontier voxels observable by the candidate viewpoint. The iteration continues until either the overall coverage of the cluster exceeds 95%, or the maximum ΔIG falls below 15% of the total frontier cell number of C (Line 3-11). The selection of viewpoints under HS follows the same procedure, except that terrestrial viewpoints are prioritized. If the selected terrestrial viewpoints alone cannot achieve sufficient coverage, aerial viewpoints are subsequently introduced to complete the coverage (Line 12-27).

This process produces two alternative viewpoint sets \mathcal{P}_{AS} and \mathcal{P}_{HS} each corresponding to a valid coverage strategy. All viewpoints belonging to the same cluster compose a viewpoint group $\mathcal{G} = \{\mathcal{P}_{\text{AS}}, \mathcal{P}_{\text{HS}}\}$. An example of the viewpoints generated under AS and HS is shown in Fig. 1(b).

Note that the problem exhibits submodularity [2]; that is, as more viewpoints are selected, the marginal IG from selecting an additional viewpoint decreases. Nemhauser et al. [2] proved that a greedy algorithm—which starts from the empty set and iteratively selects the element with the highest marginal gain—guarantees a solution with total reward no less than $(1 - 1/e)$ of the optimal value. Therefore, we adopt a greedy strategy for viewpoint selection in both AS and HS procedures.

Algorithm 1 Generate Bimodal Viewpoints for Cluster C

Input: Frontier cluster C **Output:** AS viewpoints, HS viewpoints

```
1 // Raw Viewpoints Sampling
2  $\mathcal{P}_{\text{ground}} \leftarrow \text{sampleRawGroundViewpoints}(C)$ 
3  $\mathcal{P}_{\text{air}} \leftarrow \text{sampleRawAirViewpoints}(C)$ 

4 // Aerial-Modal Strategy (AS)
5  $\mathcal{P}_{\text{AS}} \leftarrow \emptyset$ ,  $\text{coverage} \leftarrow 0$ 
6 while  $\text{coverage} < 0.95$  do
7    $\mathcal{P} \leftarrow$  viewpoint in  $\mathcal{P}_{\text{air}}$  with max  $\Delta IG$ 
8   if  $\max \Delta IG < 0.15 \times C.\text{size}$  then
9     Break
10   $\mathcal{P}_{\text{AS}} \leftarrow \mathcal{P}_{\text{AS}} \cup \{\mathcal{P}\}$ 
11  update coverage with  $\mathcal{P}$ 
12   $\mathcal{P}_{\text{air}} \leftarrow \mathcal{P}_{\text{air}} \setminus \{\mathcal{P}\}$ 

13 // Hybrid-Modal Strategy (HS)
14  $\mathcal{P}_{\text{HS}} \leftarrow \emptyset$ ,  $\text{coverage} \leftarrow 0$ 
15 while  $\text{coverage} < 0.95$  do
16    $\mathcal{P} \leftarrow$  viewpoint in  $\mathcal{P}_{\text{ground}}$  with max  $\Delta IG$ 
17   if  $\max \Delta IG < 0.15 \times C.\text{size}$  then
18     Break
19    $\mathcal{P}_{\text{HS}} \leftarrow \mathcal{P}_{\text{HS}} \cup \{\mathcal{P}\}$ 
20   update coverage with  $\mathcal{P}$ 
21    $\mathcal{P}_{\text{ground}} \leftarrow \mathcal{P}_{\text{ground}} \setminus \{\mathcal{P}\}$ 

22 while  $\text{coverage} < 0.95$  do
23    $\mathcal{P} \leftarrow$  viewpoint in  $\mathcal{P}_{\text{air}}$  with max  $\Delta IG$ 
24   if  $\max \Delta IG < 0.15 \times C.\text{size}$  then
25     Break
26    $\mathcal{P}_{\text{HS}} \leftarrow \mathcal{P}_{\text{HS}} \cup \{\mathcal{P}\}$ 
27   update coverage with  $\mathcal{P}$ 
28    $\mathcal{P}_{\text{air}} \leftarrow \mathcal{P}_{\text{air}} \setminus \{\mathcal{P}\}$ 

29 return  $\mathcal{P}_{\text{AS}}$ ,  $\mathcal{P}_{\text{HS}}$ 
```

2 Enhancements of Bimodal Motion Planner

To better support our exploration framework, we introduce several key enhancements over our previous design.

First, instead of assuming a predefined ground plane (e.g., at a fixed height $z = 0$), we employ an online ground extraction module based on 3D perception, which dynamically

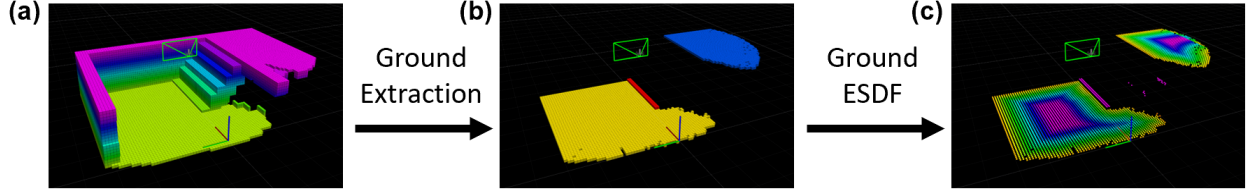


Figure 2: Ground extraction and ground ESDF generation. (a). Grid map. (b). Extracted ground part from the grid map. (c). ESDFs of the ground segments.

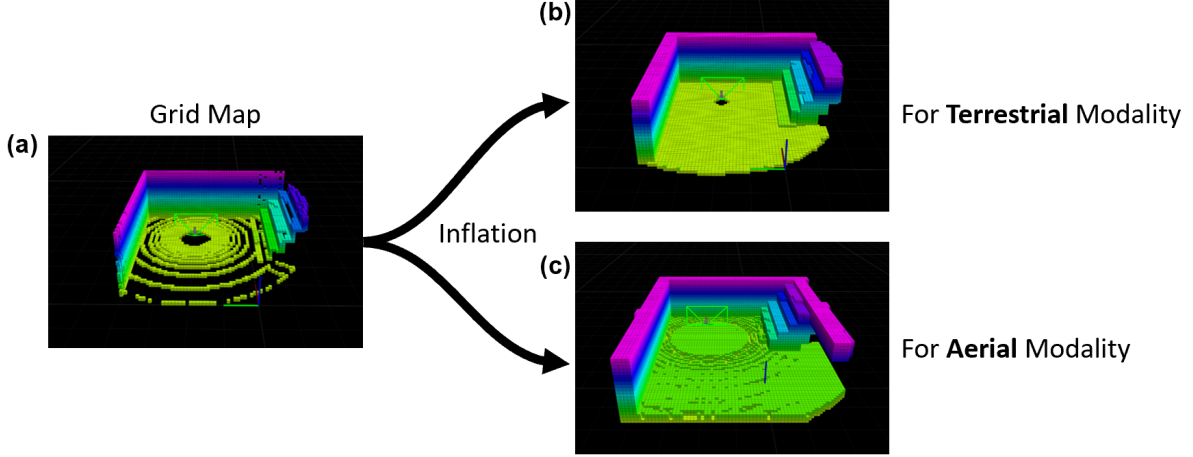


Figure 3: Grid map generation for different modalities. (a). The voxel grid map before inflation. (b). The inflated grid map for terrestrial path searching. The z axis of ground segments will not be inflated. (c). The inflated grid map for aerial path searching.

segments traversable surfaces from the environment, as shown in Fig. 2(b). Specifically, we extract ground segments from the voxel grid map using an incremental breadth-first search (BFS). A voxel is identified as a ground cell if it has a sufficient number of neighboring voxels at a similar height within a local window. A newly detected ground cell will inherit the ground segment ID from the neighboring old one. Fig. 2(b) shows two ground segments with two different IDs. When a new portion of the grid map is built, we use BFS to continue expanding ground segments from neighboring ground-consistent cells. Importantly, we check each ground segment, and if the number of connected voxels is too small, it will not be treated as valid ground—such as the small red patch in Fig. 2(b). The online terrain perception enables the planner to generate feasible paths in complex, multi-level environments without relying on prior structural assumptions.

Second, rather than discarding all point cloud data below a certain height threshold to avoid constructing a ground-influenced grid map—as done in our previous work—we now build two separate voxel grid maps tailored for each modality as shown in Fig. 3. For the terrestrial modality, the ground part of the grid map will not be inflated along the z axis, enabling the terrestrial path searching to search closely on the ground. For aerial modality, we will perform standard isotropic inflation in all three dimensions, enabling collision-free aerial path searching.

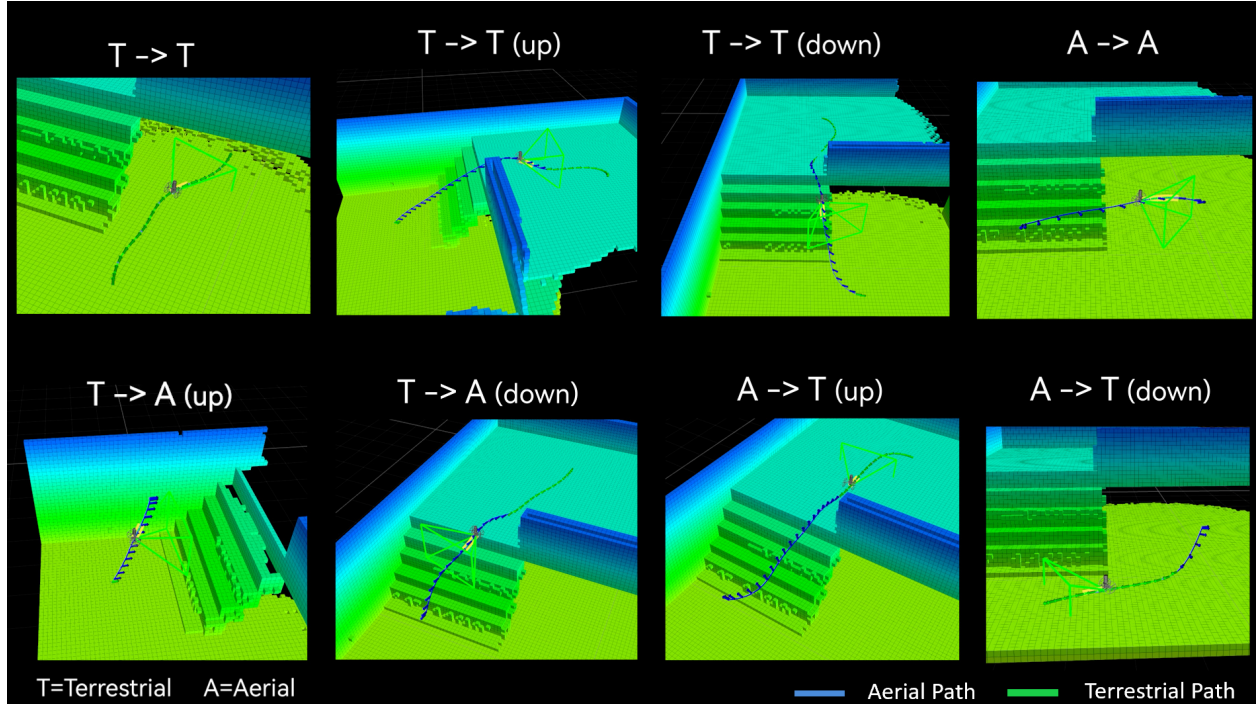


Figure 4: Results of the bimodal motion planner in simulation.

Third, unlike the previous method that always preferred the terrestrial modality by applying a constant penalty to aerial actions, our approach handles the two modalities separately. In the kinodynamic path search front end, we expand motion primitives based on the modality of the target: for aerial target, only aerial primitives are used; for terrestrial target, both terrestrial and aerial primitives are expanded, with penalties on aerial ones to enable switching only when necessary (e.g., stair climbing or obstacle overcrossing). In the back end, we apply modality-specific constraints: aerial segments are bounded by aerial velocity and acceleration limits, while terrestrial segments additionally obey nonholonomic dynamics. The back-end design is the same as our prior work [3].

Lastly, to improve safety near discontinuities in terrain, we compute the Euclidean Signed Distance Field (ESDF) for each ground segment as shown in Fig. 2(c). The ESDF provides distance-to-edge information for back-end trajectory optimization, allowing the system to plan safe aerial trajectories near boundaries and avoid unintended falls. As shown in the terrestrial-to-terrestrial ($T \rightarrow T$ (down)) trajectory in Fig. 4, the TABV switches to the aerial modality before the edge to ensure safety.

Altogether, these enhancements allow the TABV to perform motion planning in unknown, multi-level environments, resulting in a fully functional and autonomous bimodal exploration system. We present the improved bimodal motion planning in simulation (Fig. 4) and real world (Fig. 5).

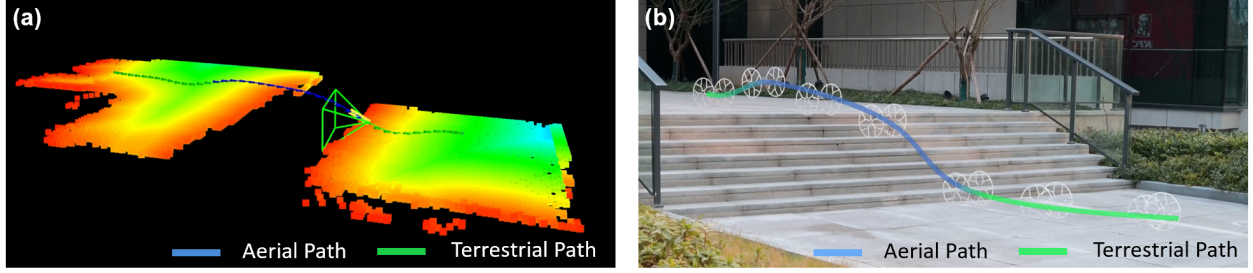


Figure 5: Real-world multi-layer motion planning experiment. (a). The planned bimodal trajectory and ESDF of extracted ground. (b). The snapshot of the real-world experiment.

3 Topological Graph

3.1 Construction of the Topological Graph

We construct and maintain a topological graph (topo-graph) during exploration to record the spatial connectivity traversed by the robot. This allows efficient global path queries, even between distant frontier clusters, by estimating navigation distances directly on the graph.

The topo-graph is built using a simple rule: a new node is added every fixed distance along the robot’s trajectory. For each new node, we identify nearby nodes within a predefined Euclidean distance threshold. If a nearby node is visible from the current node (determined via ray-casting in the grid map) and the topological distance between them on the existing graph exceeds a threshold, an edge is created between the two. This process is then recursively applied to the neighbors of the connected node, enabling sparse yet meaningful graph connectivity and avoiding overly dense links.

Additionally, each new node is always connected to the most recent node in the topo-graph, ensuring continuity along the robot’s path.

An result of topo-graph is presented in Fig. 6.

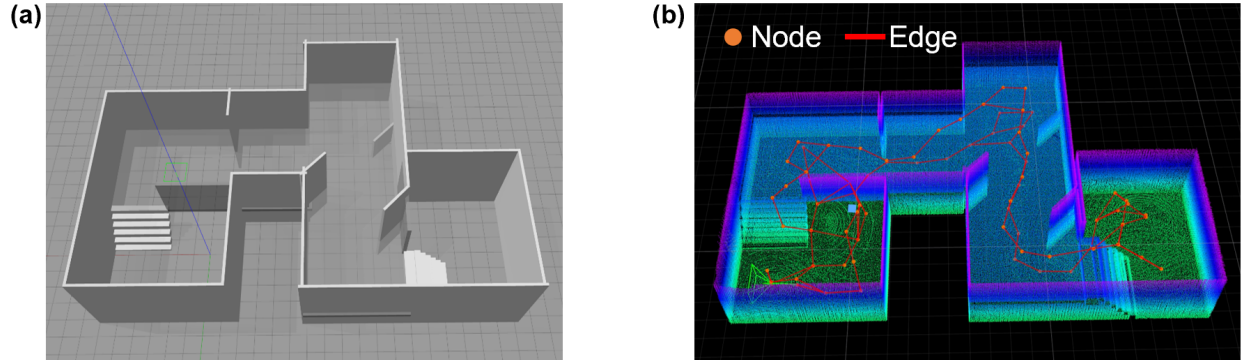


Figure 6: (a). The exploration scene in the Gazebo simulator. (b). The result of topo-graph at the end of exploration.

3.2 Path Searching using Topological Graph

By performing an A^* search on the topo-graph, we obtain a fast conservative path length estimation. Specifically, when estimating the path length between two viewpoints (In the simulation step in BM-MCTS), we use a three-stage strategy based on their Euclidean distance and search time constraints:

- i) If the straight-line distance is less than a predefined threshold (`dis_thres`), we first attempt an A^* search on the voxel grid map for a more accurate estimate. If this search exceeds a time threshold without success, we fall back to topological graph search.
- ii) If the straight-line distance exceeds the threshold, or if the voxel map search fails due to timeout, we perform a faster A^* search on the topological graph. This provides a coarser but efficient approximation of the path.
- iii) If both methods fail to find a feasible path, we conservatively assign a large cost value to indicate infeasibility.

This hybrid estimation strategy balances accuracy and computational efficiency, supporting real-time performance in the BM-MCTS framework.

4 Replanning and Exit Mechanism

The estimated energy and time consumption may not fully reflect real-world factors such as terrain resistance or aerodynamic effects, leading to inaccuracies. Additionally, as the robot moves, new information becomes available. Therefore, it is essential to *replan* in real time. Replanning is triggered under the following conditions:

- i) The selected local goal lies in a frontier cluster whose coverage exceeds 50%, meaning the majority of its frontier cells no longer qualify as frontier cells (i.e., they are no longer free cells adjacent to unknown space).
- ii) The previously planned trajectory has been fully executed.
- iii) A predefined time interval has elapsed.

In addition, the system must also decide when to terminate exploration and return to the departure station. This decision is triggered under the following circumstances:

- i) *No valid solution is found by BM-MCTS.* This occurs when all nodes are pruned, indicating that there is no cluster that can be visited while still preserving sufficient energy and time to return the departure station.
- ii) *The remaining energy or time is insufficient to safely return.* Specifically, the remaining energy is no greater than $E(\mathcal{G}_i, \mathbf{p}_H)$ or the remaining time is no greater than $T(\mathcal{G}_i, \mathbf{p}_H)$, where both are conservative estimates of the cost to return to the departure station. We adopt a *conservative return strategy* that always uses the same modality as recorded in the topological graph to ensure the feasibility and safety of the return.

- iii) *No frontier clusters remain* in the environment, indicating that all unknown regions have been explored.

5 BM-MCTS Algorithm Analysis

Note that in the simulation process, we assume $M = (T + A)/2$ to generate a coarse guidance path, introducing inaccuracies. However, the estimation inaccuracy is progressively corrected during the planning process. If the estimation E_r is less than the true value, the tree will not tend to expand through it. But due to the exploration property of the UCB rule, each node has a nonzero probability of being selected and further estimated. If the estimation E_r is overestimated, the node is more likely to be selected to expand. After expansion, the overestimation will be reduced. And the convergence of such selection policy is proved [4].

As for the time complexity of the path searching algorithm, the path between \mathcal{G} s needs to be computed only when new frontier clusters are generated. The path between \mathcal{P} s is computed during the tree expansion only for the newly generated \mathcal{P} s. The time complexity of both steps is $\mathcal{O}(UV)$, where U represents the number of new \mathcal{P} s or \mathcal{G} s, and V represents the number of existing ones. Furthermore, a topo-graph is utilized to accelerate this process.

Additionally, the BM-MCTS is an anytime algorithm, meaning that it can produce solutions at any time during its execution and continually improve the quality of these solutions as it is given more computational resources, until the tree is fully expanded.

6 Extreme Resource Limitation Case

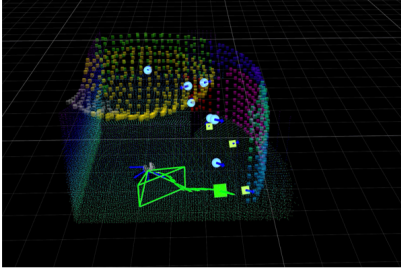
In this section, we evaluate the performance of our exploration system under an extreme resource-constrained scenario. As illustrated in Fig. 7, the exploration begins with an energy budget of 25 and a time budget of 30, which is significantly lower than the amount required to cover all available viewpoints.

Despite the limited resources, the TABV successfully explores a portion of the environment and returns to the departure station with remaining energy and time of $E_r = 5$ and $T_r = 10$, respectively. The remaining energy is not exactly zero because the remaining frontier clusters are considered unreachable. The estimated energy was insufficient to cover any of these clusters and return to the departure station safely.

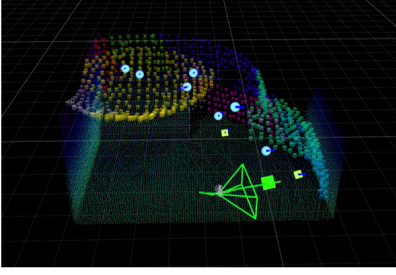
7 Details of A Large Two-story House Exploration

In this section, we provide additional details about the experiment, which were omitted in the main paper due to space limit. We test our system in the Gazebo with dynamics simulation as Fig. 8 shown. The test environment is a two-story scene of size 18 m \times 24 m \times 4 m, featuring two staircases. The TABV is equipped with a 360° LiDAR sensor with a vertical FoV of -30° to 30° and a maximum range of 4.5 meters. Note that our method is compatible with any sensor that has a limited FoV. In this simulation, we use a LiDAR with a constrained vertical FoV, while in the simulation study in Sec. VII.A (Fig. 7 of the main

$E_r = 22, T_r = 27$



$E_r = 17, T_r = 22$



$E_r = 9, T_r = 14$

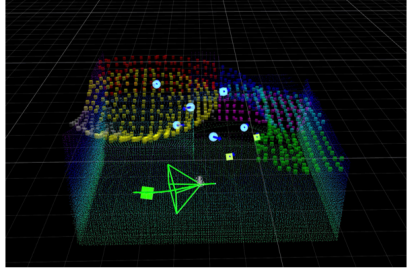


Figure 7: Exploration with energy budget = 25, which is much less than the amount for covering all existing viewpoints.

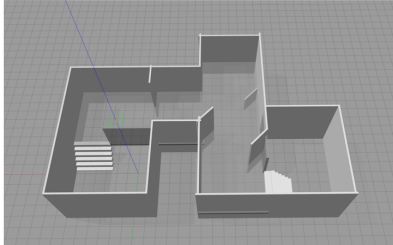
paper), a depth sensor with a horizontal FoV of 90° , vertical FoV of 60° , and a sensing range of 3.5 meters is used. The FoV is explicitly passed as a parameter to the planner, and the viewpoint generation process dynamically adapts to the sensor's FoV.

The exploration task is with an energy budget of 900 and a time budget of 900. During the early stage (from t_0 to t_2), both energy and time budgets are sufficient, so information gain is prioritized.

- t_0-t_1 : On the first floor, due to its low ceiling, the observation quality of terrestrial and aerial modalities is similar. Since aerial modality consumes more energy, the terrestrial modality is preferred.
- t_1-t_2 : An informative aerial-only viewpoint appears, prompting a switch to the aerial modality. Upon entering a two-story-high hall, aerial viewpoints provide higher information gain, so the robot continues exploring in aerial modality and lands on the second floor after finishing the hall exploration.
- t_2-t_3 : With energy consumption accumulating and remaining budget limited, and aerial modality no longer providing an advantage for information gain, the robot switches back to terrestrial modality to finish exploring the second floor and then flies downstairs.
- t_3-t_4 : A few remaining aerial-only unknown regions near the starting area are covered using the aerial modality.

The entire exploration process is completed in 243 seconds.

(a)



(b)

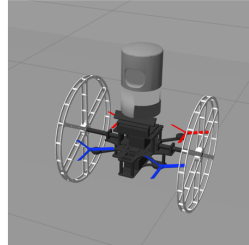


Figure 8: Gazebo simulation. (a). A large two-story house scene. (b). The TABV model with dynamics in Gazebo.

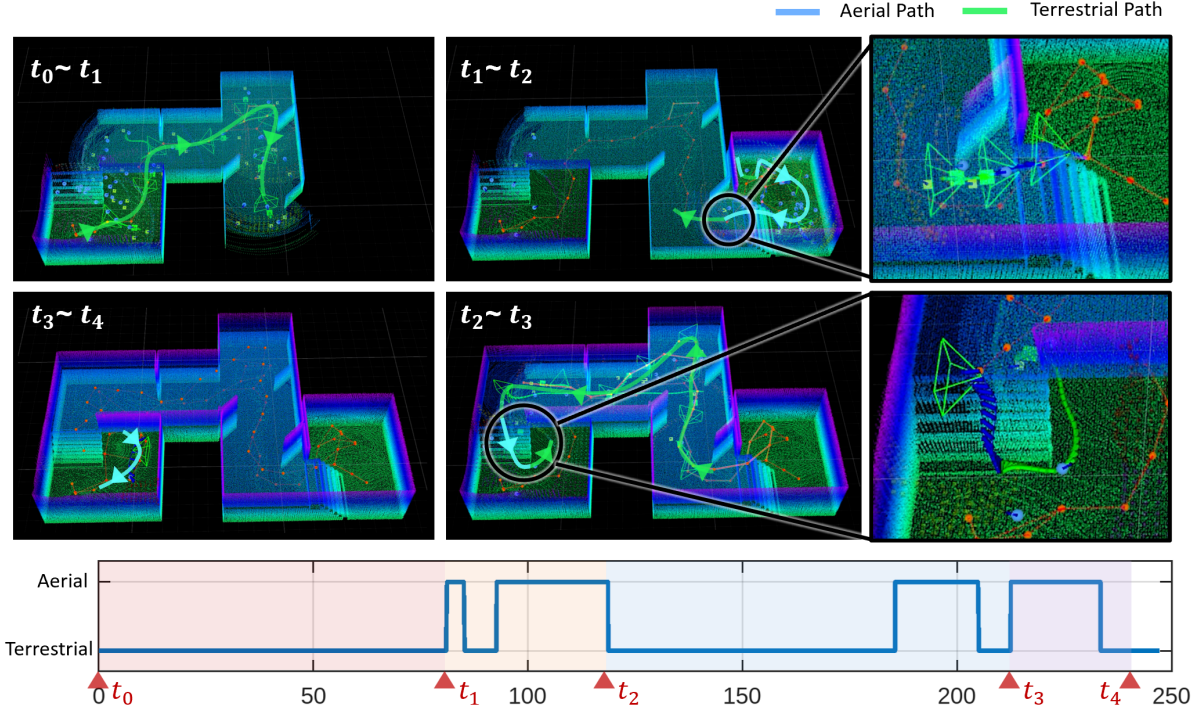


Figure 9: Results of the two-story house exploration.

8 Supplementary Comparisons

To better demonstrate the effectiveness of our bimodal exploration system, we present comprehensive comparisons across multiple levels of abstraction. We provide comparisons at three levels:

- i) **Framework-level comparison (Sec. 8.1):** Since there is no existing exploration framework specifically designed for bimodal systems, we compare our complete TABV exploration framework against a state-of-the-art purely aerial exploration framework (FUEL [5]). This comparison highlights the advantages of integrating both aerial and terrestrial modalities at the framework level, including improved exploration capability and computational efficiency in large-scale environments.
- ii) **Modality configuration comparison (Sec. 8.2):** We conduct ablation studies comparing our bimodal system against three baselines: **UAV-only**, **UGV-only**, and **UGV-cross** (a rule-based bimodal configuration that switches modalities solely based on traversability). This comparison highlights the benefit of integrating both mobility modalities, and making modality-switching based on energy and time constraints.
- iii) **Resource-aware decision-making algorithm comparison (Sec. 8.3):** We compare our decision-making algorithm, which optimizes viewpoint sequences under energy and time constraints, against classical methods based on the Travelling Salesman Problem (TSP) and Next-Best-View (NBV) strategies [6]. This highlights the efficiency and adaptability of our planning algorithm in energy and time constrained exploration.

These three levels of comparison jointly validate the contributions of our system design: the integrated framework, the coordinated use of dual modalities, and the planning algorithm under resource constraints.

8.1 Exploration Framework-Level Comparison

We perform a framework-level comparison between our bimodal exploration system and FUEL [5], a powerful framework for fast UAV exploration that represents the state-of-the-art in aerial exploration.

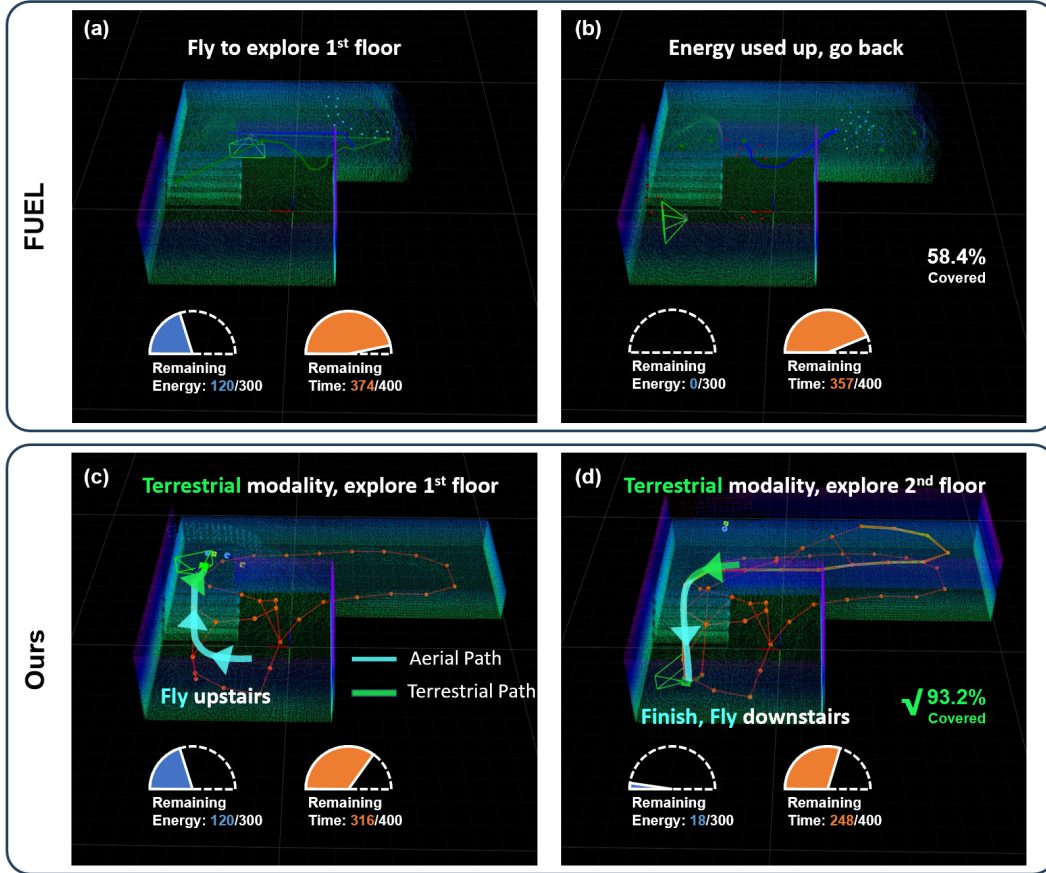


Figure 10: Exploration process of Case 1 (E300, T400) in the simulation. (a)(c). Result when remaining energy = 120. (b)(d). Result when energy is used up.

FUEL is a hierarchical framework designed for purely aerial exploration in complex unknown environments. It incrementally maintains a global Frontier Information Structure (FIS) to support efficient exploration planning. FUEL employs a TSP-based method to generate a global coverage path, followed by local viewpoint refinement and time-optimal trajectory generation. For fair benchmarking, we ported FUEL into our Gazebo simulation and executed its planned trajectories via the same NMPC controller used for TABV. Except for adjusting velocity and acceleration limits as well as the frontier cluster size to match our own system parameters, we did not modify any other components of FUEL. Notably,

Table 1: Framework-Level Comparison

Case	Method	Coverage (%)	Energy Used	Time Taken (s)	Remarks
Case 1 (E300, T400)	FUEL	58.4	300*	48	Energy shortage
	Ours	93.2✓	282*	152	Almost covered
Case 2 (E800, T100)	FUEL	95.4✓	700	100*	Almost covered
	Ours	94.6✓	460	94*	Almost covered

1. ✓ denotes that environment almost covered, * denotes using up the resource.
2. **Bold** entries denote the best results for each metric in the corresponding case.

FUEL does not support ground segmentation, terrestrial viewpoint generation, or decision-making mechanisms for modality switching. Therefore, this section focuses on comparing the overall frameworks. A more detailed comparison of algorithmic performance under the terrestrial-aerial exploration context will be provided in Section 8.3.

As shown in Tab. 1, we compare our exploration framework with FUEL in two cases. In case 1 (E300, T400), under limited energy conditions, FUEL lacks terrestrial mobility and thus fails to complete the exploration task efficiently due to the high energy cost of sustained flight. The exploration details of case 1 are shown in Fig. 10. In case 2 (E800, 100), under tight time constraints, both systems rely primarily on the aerial modality, resulting in comparable performance.

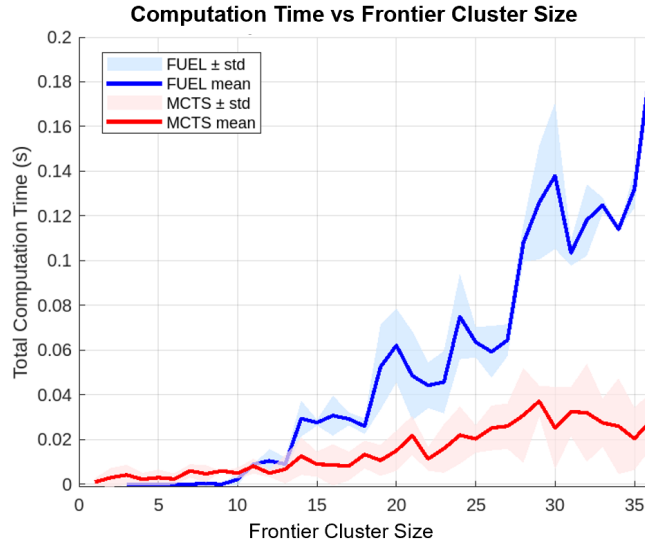


Figure 11: Computation time comparison between FUEL and ours.

Additionally, in large-scale environments, FUEL suffers from low computational efficiency due to the absence of the topo-graph, which is explicitly constructed and utilized in our framework to support scalable planning. The relationship between computation time and the number of frontier clusters is illustrated in Fig. 11. For FUEL, the computation time includes both cost matrix construction and TSP solving; for our method, it accounts for the entire process of Bimodal Monte Carlo Tree Search (BM-MCTS) with ten iterations. The same BM-MCTS parameters are used in the algorithm-level comparison in Sec. 8.3.

In summary, the results shows that the proposed method surpasses FUEL in energy-constraint case and presents comparable performance in time-constraint case. In addition, the proposed MCTS-based approach reveals computational stability against increasing frontier cluster size, while FUEL fails.

8.2 Modality Configuration Comparison

To systematically assess the impact of different modality configurations on exploration performance under resource constraints, we conduct ablation studies comparing our proposed bimodal system with three baseline variants:

- **UAV-only**: A single-modality system relying solely on aerial exploration.
- **UGV-only**: A ground-only exploration system, with no aerial capability.
- **UGV-cross**: A hybrid-modal strategy that switches to aerial modality *only* when no viewpoint can be reached via traversable terrestrial paths.
- **Ours**: The full version of our BM-MCTS-based bimodal exploration system with strategic modality switching.

These systems are evaluated in the simulation conditions in two budget-constrained cases. The results are shown in Tab. 2 and Fig. 12. The modification for each variant is only the exploration planner part, and they use the same motion planner.

This comparison also directly responds to the reviewer’s concern that our results might be achievable by simply combining the two modalities by flying over untraversable parts of the environment and using terrestrial modality in the rest of the environment (corresponding to the **UGV-cross** variant).

Table 2: Modality Configuration Comparison

Case	Method	Coverage (%)	Energy Used	Time Taken (s)	Remarks
Case 1 (E300, T400)	UAV-only	65.1	300*	43	Energy shortage
	UGV-only	61.5	90	90	2 nd floor unreachable
	UGV-cross	92.1✓	236*	67	Almost covered
	Ours	93.2✓	282*	48	Almost covered
Case 2 (E800, T100)	UAV-only	94.6✓	460	94*	Almost covered
	UGV-only	61.5	96	96*	2 nd floor unreachable
	UGV-cross	66.4	136	100*	Time shortage
	Ours	94.6✓	460	94*	Almost covered

1. ✓ denotes that environment almost covered, * denotes using up the resource.
2. **Bold** entries denote the best results for each metric in the corresponding case.
3. UGV-cross: a baseline strategy that switches to aerial modality only when no viewpoint can be reached via traversable terrestrial paths.

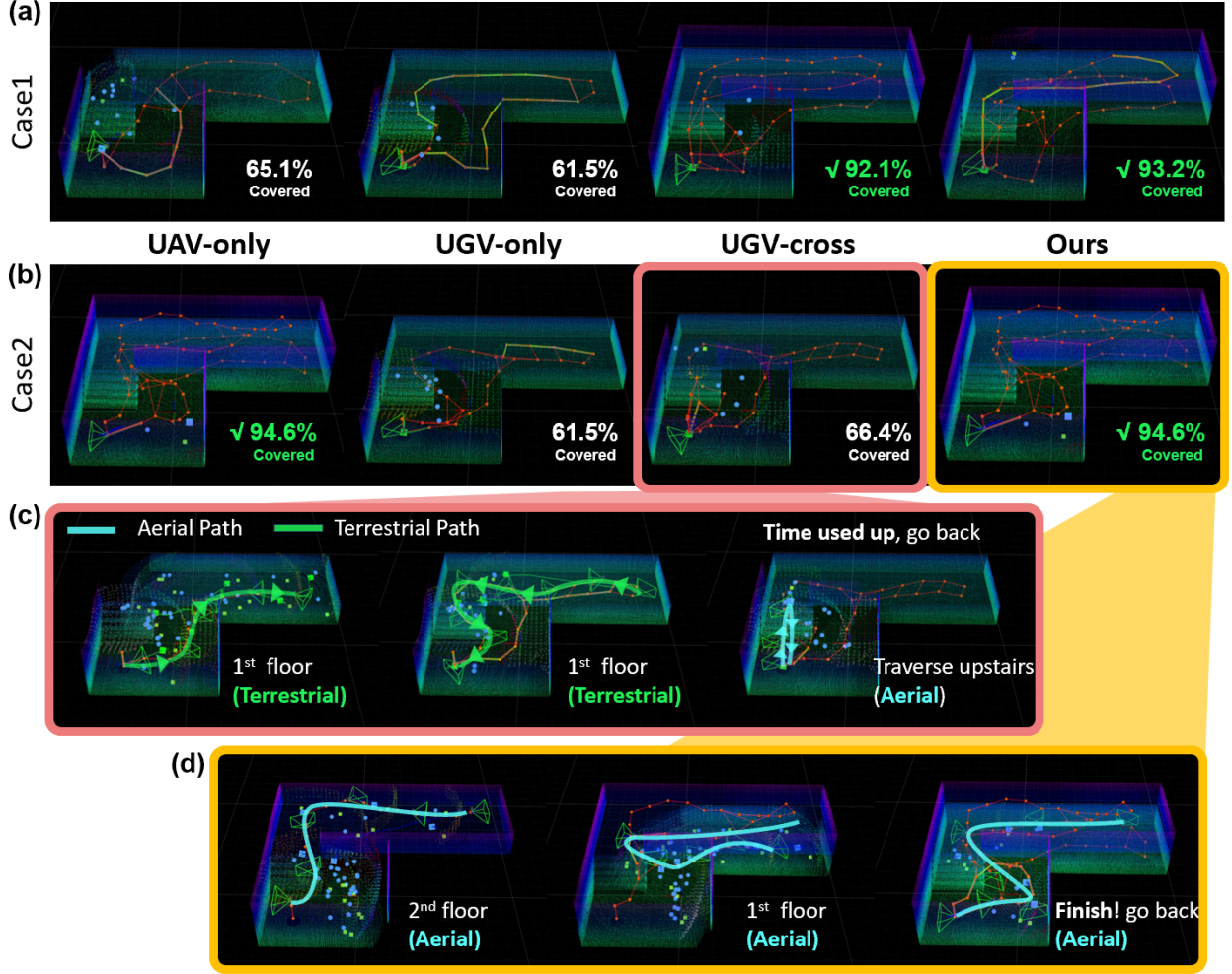


Figure 12: Modality configuration comparisons under two cases. (a). Case 1, where energy budget = 300, time budget = 400. (b). Case 2, where energy budget = 800, time budget = 100. (c). The exploration process of **UGV-cross** system in case 2. The **UGV-cross** variant fails to explore the whole environment when time budget is tightly limited. (d). The exploration process of our system in case 2. Our system can always maintain high coverage of the unknown environment under different resource conditions.

In case 1 (E300, T400) (Fig. 12(a)), where energy budget is tighter, the **UAV-only** system consumes energy rapidly due to sustained flight and is only able to cover 65.1% of the environment before depleting its resources. The **UGV-only** system is unable to access the second floor and therefore can only complete exploration on the ground level. Both the **UGV-cross** and our system are able to almostly complete the full exploration task in this scenario. Note that their energy usage in the table does not reach the maximum budget, this is because one remaining frontier cluster is deemed unreachable: the estimated energy was insufficient to cover that cluster and return to the departure station. Therefore, the robot opted to return directly, leaving that cluster unexplored.

In case 2 (E800, T100) (Fig. 12(b)), where time budget is tighter, the **UAV-only** system completes full exploration within the time budget. The **UGV-only** system again

fails to reach the second floor and is limited to covering the first level. The **UGV-cross** system switches to aerial modality only when the next viewpoint is unreachable in terrestrial modality. As a result, it completes exploration on the first floor, but runs out of time while attempting to ascend via aerial modality, and has to return before reaching the second level. Its coverage is therefore limited to 66.4%. The entire **UGV-cross**'s exploration process is visualized in Fig. 12(c). In contrast, our system anticipates the time constraint and proactively switches to aerial modality, completing the full exploration.

In summary, our system can dynamically adapt to varying energy and time budgets. The BM-MCTS-based planner estimates future resource consumption and evaluates nearby viewpoints with their respective modality and position, enabling flexible and strategic switching to maximize environment coverage under constraints.

8.3 Resource-aware Decision-making Algorithm Comparison

We compare our BM-MCTS method with NBV-based and TSP-based method. The NBV-based method adopts a greedy policy by selecting the viewpoint with the highest immediate gain, while the TSP-based method computes a global tour over all viewpoints to minimize total cost.

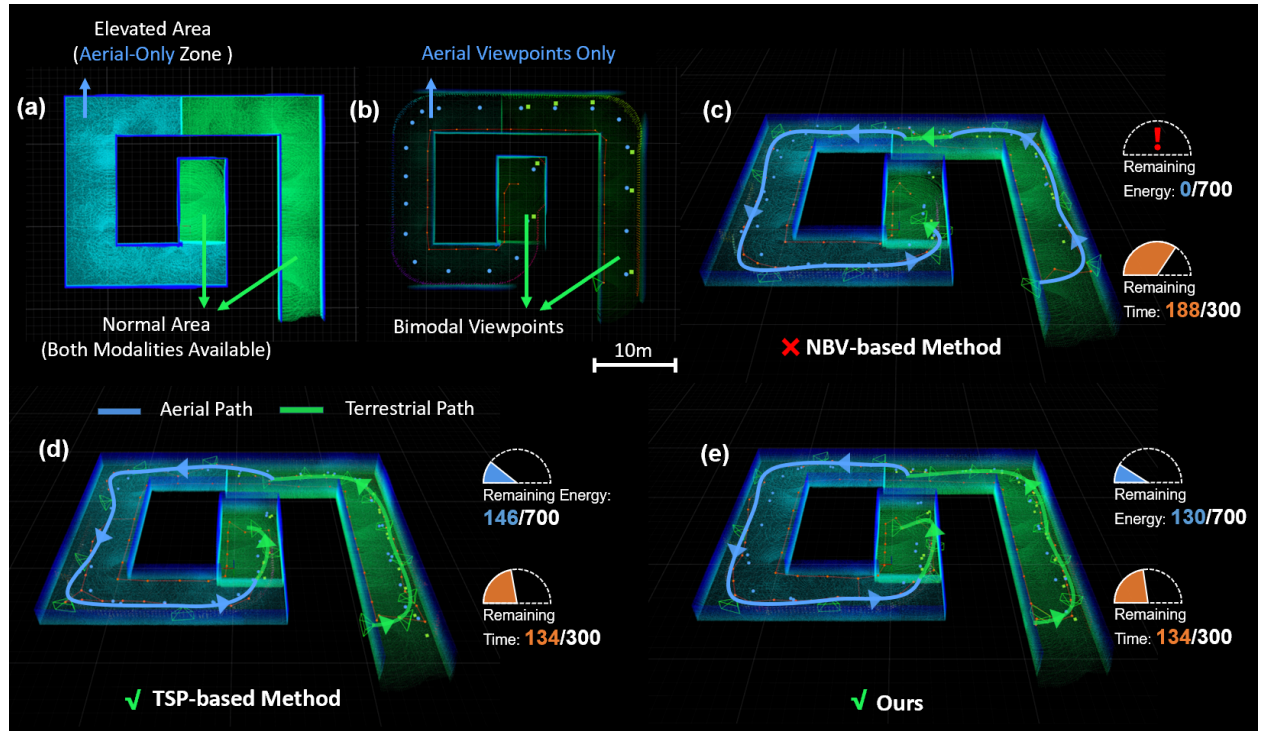


Figure 13: Resource-aware decision-making algorithm comparisons (a). The simulation environment. The blue elevated area represents a region that is untraversable by the terrestrial modality. (b). A half-explored environment that explored by a fix trajectory. (c). The exploration result of the NBV-based method. (d). The exploration result of the TSP-based method. (e). The exploration result of our method.

For the algorithm-level comparison, we retain the same bimodal viewpoint generation and motion planning components, and replace only the BM-MCTS module with either the NBV-based method or the TSP-based method.

For the NBV-based method, we directly evaluate all candidate viewpoints and select the one with the minimum score for execution. To ensure consistency with the scoring mechanism used in the BM-MCTS selection process, we define the score of a viewpoint \mathcal{P} based on three components: (1) $IG(\mathcal{P})$, the information gain, measured by the number of frontier cells visible from \mathcal{P} ; (2) $\kappa_{E_r}(\mathcal{P})$, a penalty term based on the remaining energy after visiting \mathcal{P} ; and (3) $\kappa_{T_r}(\mathcal{P})$, a penalty term based on the remaining time after visiting \mathcal{P} . The overall score function is defined identically to Equation (6b) in the revised main paper:

$$G(\mathcal{P}) = -N(IG(\mathcal{P})) + \kappa_{E_r}(\mathcal{P}) + \kappa_{T_r}(\mathcal{P}), \quad (1)$$

where $N(x)$ linearly normalizes x to the range $[\epsilon, 1]$, and $\epsilon = 0.05$ denotes the minimum normalized value. The energy and time penalty terms are computed using exponential functions: $\kappa_{E_r}(\mathcal{P}) = \exp(-a_1 \cdot E_r(\mathcal{P})/E_{all} + b_1)$, $\kappa_{T_r}(\mathcal{P}) = \exp(-a_2 \cdot T_r(\mathcal{P})/T_{all} + b_2)$, where E_{all} and T_{all} denote the total energy and time budgets, respectively, and $E_r(\mathcal{P})$, $T_r(\mathcal{P})$ represent the remaining energy and time after reaching \mathcal{P} from the current pose. The hyperparameters a_1 , a_2 , b_1 , and b_2 are set identically to those used in the BM-MCTS algorithm.

For the TSP-based method, we adapt the standard approach to accommodate the bimodal exploration context. For each frontier cluster, we consider two alternative strategies: visiting viewpoints sampled under the aerial modality or under the hybrid (terrestrial-aerial) modality, as described in Sec. IV.A. Therefore, for n frontier clusters, there exist 2^n possible modality assignments, and we solve the TSP for each combination to find the optimal visiting sequence. For each candidate sequence $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$, where each \mathcal{P}_i corresponds to a selected viewpoint in a frontier cluster, we compute its score using the same formulation:

$$G(\mathcal{P}) = -N(IG_d(\mathcal{P})) + \kappa_{E_r}(\mathcal{P}) + \kappa_{T_r}(\mathcal{P}), \quad (2)$$

where $IG_d(\mathcal{P})$ denotes the discounted average information gain of the entire viewpoint sequence \mathcal{P} (to be consistent to the IG-related item in Equation (5a) in the revised main paper), computed as:

$$IG_d(\mathcal{P}) = \frac{IG(\mathcal{P}_1) + \gamma_{IG} IG(\mathcal{P}_2) + \dots + \gamma_{IG}^{n-1} IG(\mathcal{P}_n)}{1 + \gamma_{IG} + \dots + \gamma_{IG}^{n-1}}, \quad (3)$$

where γ_{IG} is the discount factor (same as γ_{IG} in the revised main paper Alg. 2). And the energy/time terms $\kappa_{E_r}(\mathcal{P})$ and $\kappa_{T_r}(\mathcal{P})$ represent the remaining resources after completing the full trajectory and returning to the departure station. The sequence with the minimum $G(\mathcal{P})$ is selected for execution. While the TSP-based method provides a globally optimal solution, the BM-MCTS algorithm relies on its simulation phase to estimate energy and time costs for portions of the search tree that have not yet been expanded.

We compare the three algorithms in the simulation shown in Fig. 13(a). To ensure a fair comparison under stochastic decision-making, we employ a simulation environment with a single-path topological structure to ensure consistent spatial topology in exploration across all methods. In the elevated area, only aerial viewpoint can be generated, and in the

normal area, viewpoints of both modalities can be generated. We first make the environment half-explored with a fixed trajectory (Fig.13(b)), and pre-generate some frontier clusters and viewpoints. Then, we start the exploration with energy budget = 700 and time budget = 300.

The results indicate that the NBV-based method is inherently shortsighted. Since it only considers the immediate next viewpoint, it fails to proactively switch to the terrestrial modality early enough in anticipation of the later aerial-only zone. As a result, the robot depletes its energy budget before completing the full exploration. In contrast, both the TSP-based method and our proposed BM-MCTS approach perform long-horizon planning. They are able to foresee the upcoming aerial-only region and conserve energy in advance by utilizing the more energy-efficient terrestrial modality earlier in the mission.

However, obtaining a globally optimal solution with the TSP-based method comes at a high computational cost. As illustrated in Fig. 14, the computation time increases rapidly as the number of frontier clusters grows.

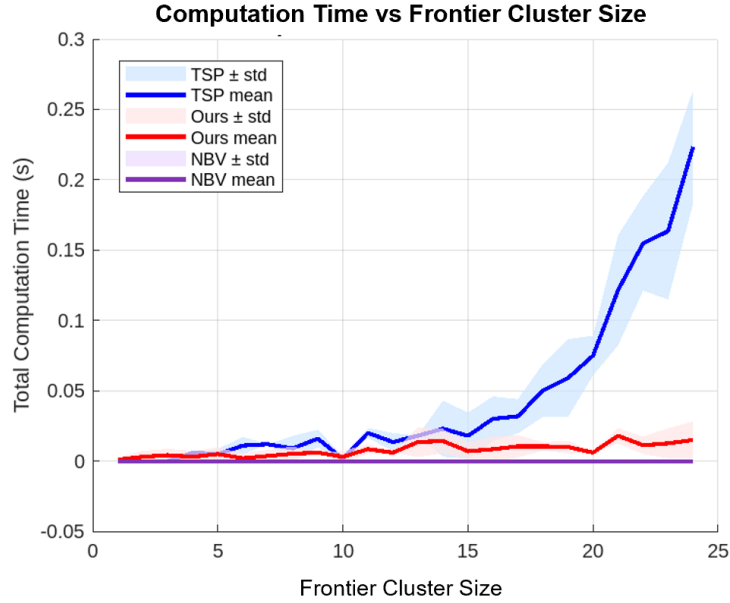


Figure 14: Computation time comparison among NBV-based, TSP-based and our algorithms.

9 Details of Problem Formulation

9.1 Hyperparameter Selection of the Penalty Function

Recall our motivation in the context of search and rescue operations. Disasters such as earthquakes and fires can alter the structure of environments, posing significant risks to human responders. Deploying robots mitigates these risks and enables exploration and data collection in areas that are otherwise inaccessible to humans.

In many disaster scenarios, communication infrastructure is often compromised. As a result, ensuring data retrieval by having robots return to the departure station becomes the

most reliable solution. Furthermore, the primary objective of exploration in such environments is to locate and rescue trapped individuals. Timely access to firsthand situational data is crucial for rescue teams, often within a limited timeframe [7], which introduces constraints on the expected exploration duration.

Given this practical context, ensuring sufficient remaining energy to return to the departure station is of utmost importance. In contrast, time constraints are generally softer—minor overruns are typically tolerable. As such, we design the penalty functions accordingly: the energy-related penalty is sharper, while the time-related penalty is flatter and only becomes influential when sufficient energy remains. Therefore, we apply a sharper penalty function for remaining energy and a flatter one for remaining time. Accordingly, we set the parameters as $a_1 = b_1 - \log(0.3)$, $b_1 = \log(10)$, $a_2 = b_2 - \log(0.7)$, and $b_2 = \log(3)$, as shown in Fig. 15.

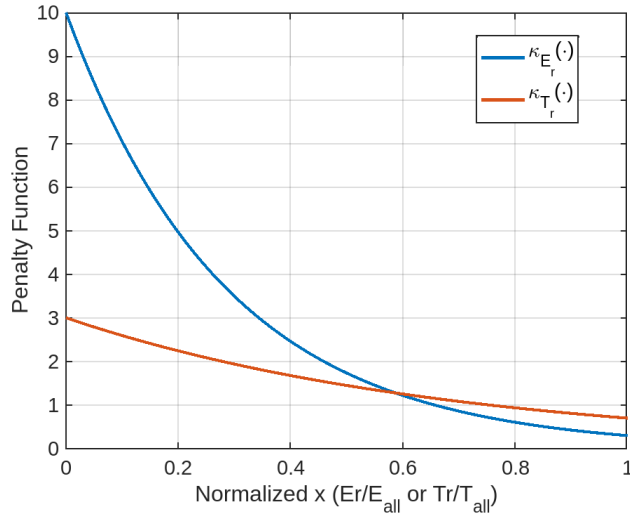


Figure 15: Penalty functions of remaining energy and time.

Such penalty functions also drive the robot to cope with environmental uncertainty by maintaining a safety margin.

9.2 IG Normalization

For IG, we apply a normalization function $N(\cdot)$ to obtain $N(R_p(v_i))$, where R_p denotes the IG-related process gain. $N(\cdot)$ linearly maps the value to the range $[\epsilon, 1]$, where $\epsilon = 0.05$ denotes the minimum normalized score.

For a node v , this normalization is performed over all v 's child nodes $\{v_i \mid i = 0, \dots, n\}$, based on their corresponding IG-related process gain $R_p(v_i)$. The normalization function is defined as:

$$N(R_p(v_i)) = \begin{cases} 0.5, & \text{if } |R_{p,\max} - R_{p,\min}| \leq \delta \\ \epsilon + (1 - \epsilon) \cdot \frac{R_p(v_i) - R_{p,\min}}{R_{p,\max} - R_{p,\min}}, & \text{otherwise} \end{cases}$$

where $R_{p,\min}$ and $R_{p,\max}$ are the minimum and maximum values among the child nodes, δ is a small threshold that defines a dead zone for normalization, preventing numerical instability when the difference between $R_{p,\max}$ and $R_{p,\min}$ is very small.

We adopt this normalization to ensure that the scores of all nodes are mapped to a comparable, dimensionless scale. This allows the IG-related term to be balanced meaningfully against the energy/time penalties. Note that in our bimodal viewpoint generation stage, each viewpoint is guaranteed to have a positive IG (i.e., it observes nonzero frontier voxel), so we use a small floor value like $\epsilon = 0.05$ to represent $R_{p,\min}$ after normalization.

10 Details of Hardware and FoV

In Fig. 16, we provide a photograph of the TABV platform that illustrates its hardware components, including the sensor placement and actuation mechanisms.

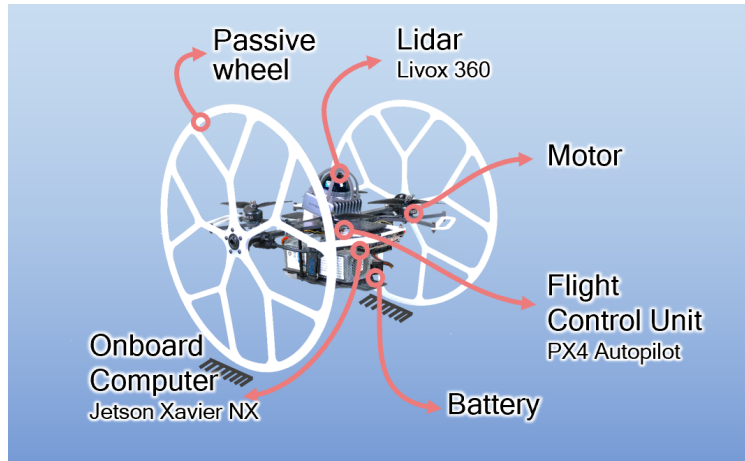


Figure 16: The hardware of the TABV.

Although the sensor remains the same, the effective FoV differs between modalities due to its vertically symmetric mounting on the TABV. This design balances two competing needs: i) In terrestrial modality, more downward visibility is needed to percept the ground, especially the nearby ground, which is critical for motion planning. ii) In general, upward visibility is essential for perceiving the surrounding environment. As a result of this compromise, the effective perceptual FoV in terrestrial modality is approximately half that in aerial modality, as shown in Fig. 17(a).

Note that our method is compatible with any sensor with a limited FoV, and the FoV is explicitly passed as a parameter to the planner. The viewpoint generation process dynamically adapts to the specified FoV of the sensor. Specifically, the FoV parameter will be used in the coverage calculation in Alg. 1.

In the simulation experiments presented in Sec. VII.A (Fig. 7 of the main paper), we use a depth sensor with a horizontal FoV of 90° , vertical FoV of 60° , and a sensing range of 3.5 meters. For the large-scale exploration task in a two-story house in Gazebo (Fig. 8), the TABV is equipped with a 360° LiDAR sensor with a vertical FoV ranging from -30° to 30° and a maximum sensing range of 4.5 meters. In the real-world experiments, we mount

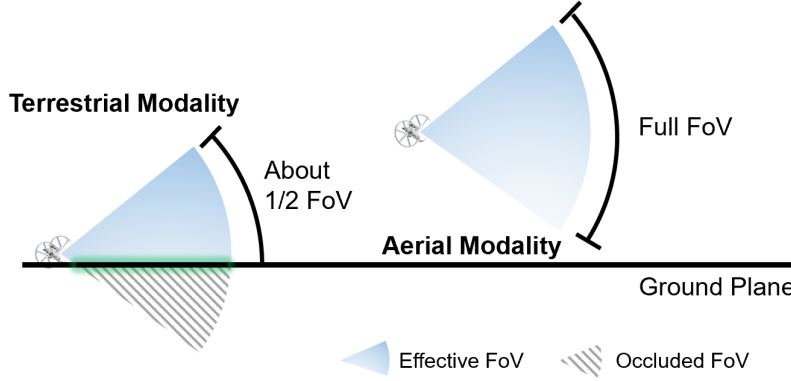


Figure 17: The effective FoV of different modalities.

a Livox Mid-360 LiDAR with a tilted configuration to obtain a vertically symmetric FoV. Due to this mounting scheme, the LiDAR effectively observes only the front-facing 180° , leading to a usable FoV of approximately $60^\circ \times 180^\circ$. Additionally, to facilitate real-world validation and account for the relatively sparse point cloud density of the Livox Mid-360 at longer distances, we set the effective sensing range to 5 meters to ensure reliable perception.

11 Limitation and Future Work

- i) **Locally myopic resource estimation.** The estimation of required energy and time is based solely on the currently observed environment. This local perspective may lead to shortsighted planning decisions. In future work, we plan to incorporate environment prediction modules as priors to enable more informed and globally-aware resource allocation during exploration.
- ii) **High System complexity.** Our proposed system integrates multiple components: viewpoint generation, topological graph construction, modality-aware decision-making, bimodal mapping, localization, bimodal motion planning and bimodal MPC controller, which makes the overall pipeline relatively complex and harder to maintain. We plan to explore more end-to-end learning frameworks, particularly for control and decision-making, to reduce system complexity and minimize the gap between modules.
- iii) **Simplified terrain analysis.** The current method determines ground traversability based on extracted ground surfaces from a grid map. However, this process is relatively simplistic, and its accuracy depends heavily on the resolution of the grid map. In future work, we plan to maintain a more detailed elevation map and perform traversability analysis based on terrain structure to improve the reliability of ground path planning.

References

- [1] C. Witting, M. Fehr, R. Bähneemann, H. Oleynikova, and R. Siegwart, “History-aware autonomous exploration in confined environments using mavs,” in *2018 IEEE/RSJ In-*

- ternational Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.
- [2] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions—i,” *Mathematical programming*, vol. 14, pp. 265–294, 1978.
 - [3] R. Zhang, Y. Wu, L. Zhang, C. Xu, and F. Gao, “Autonomous and adaptive navigation for terrestrial-aerial bimodal vehicles,” vol. 7, no. 2, pp. 3008–3015, 2022.
 - [4] W. Chen and L. Liu, “Pareto monte carlo tree search for multi-objective informative planning,” in *Proc. of Robot.: Sci. and Syst.*, 2019.
 - [5] 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.
 - [6] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang, “A two-stage optimized next-view planning framework for 3-d unknown environment exploration, and structural reconstruction,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1680–1687, 2017.
 - [7] K. P. P. Abhilash and A. Sivanandan, “Early management of trauma: The golden hour,” *Current Medical Issues*, vol. 18, no. 1, pp. 36–39, 2020.