

GeNIE: A Generalizable Navigation System for In-the-Wild Environments

Jiaming Wang^{*,1,†}, Diwen Liu^{1,†}, Jizhuo Chen^{1,†}, Jiaxuan Da¹,
Nuowen Qian¹, Minh Man Tram¹, and Harold Soh^{*,1,2}

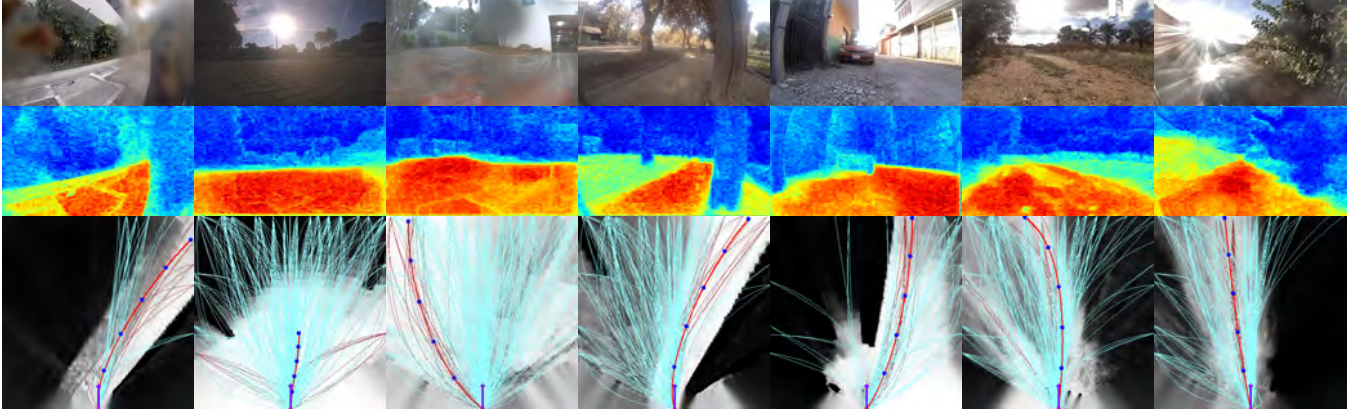


Fig. 1: Diverse input scenes (top), traversability predictions (middle), and final path planning results (bottom) across test environments unseen during training. GeNIE demonstrates strong generalization under challenging conditions, including heavy rain, high-contrast lighting, muddy lenses, and diverse terrain types.

Abstract—Reliable navigation in unstructured, real-world environments remains a significant challenge for embodied agents, especially when operating across diverse terrains, weather conditions, and sensor configurations. In this paper, we introduce GeNIE (Generalizable Navigation System for In-the-Wild Environments), a robust navigation framework designed for global deployment. GeNIE integrates a generalizable traversability prediction model with a novel path fusion strategy that enhances planning stability in noisy and ambiguous settings. We deployed GeNIE in the Earth Rover Challenge (ERC) at ICRA 2025, where it was evaluated across six countries spanning three continents. GeNIE took first place and achieved 79% of the maximum possible score, outperforming the second-best team by 17%, and completed the entire competition without a single human intervention. These results set a new benchmark for robust, generalizable outdoor robot navigation. We have released the codebase, pretrained model weights, and the datasets at <https://clear-nus.github.io/genie/> to support future research in real-world navigation.

Index Terms—Autonomous Vehicle Navigation, Vision-Based Navigation

I. INTRODUCTION

NAVIGATION is a core skill for embodied agents, allowing them to move purposefully through physical environments using inputs from onboard sensors such as cameras and GPS. It enables robots to reach goals, avoid obstacles, and operate autonomously in tasks like package delivery, environmental monitoring, or search and rescue. Over



Fig. 2: Examples of diverse terrains in ERC 2025, illustrating the requirement for robots to identify traversable areas in unstructured environments [1]. Reproduced with permission from the ERC organizers.

the past several years, numerous benchmarks and competitions have been introduced to evaluate navigation systems, including indoor simulation platforms such as Habitat [2] and Gibson [3], as well as outdoor field-based challenges focused on specific domains like subterranean navigation or agricultural environments. While these efforts have significantly advanced the field, most benchmarks evaluate navigation within narrow settings, often assuming a single type of scene, weather condition, or embodiment. As a result, high scores on these benchmarks often fail to generalize when deployed in more diverse or unstructured environments [4], [5].

In real-world scenarios, however, navigation systems must

*Corresponding authors: {jiaming, harold}@comp.nus.edu.sg

† These authors contributed equally to this work.

¹School of Computing, National University of Singapore, Singapore;

²Smart Systems Institute, National University of Singapore

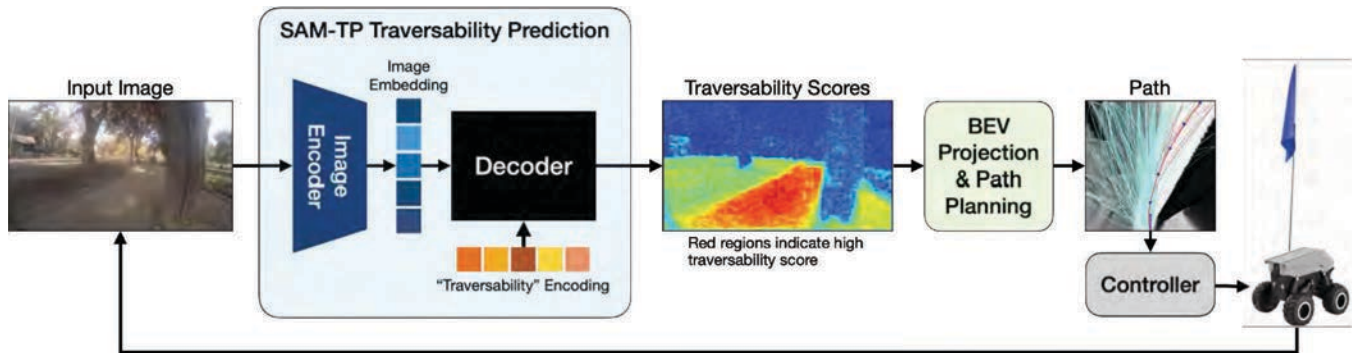


Fig. 3: Overview of the GeNIE system. Given an RGB input image, the SAM-TP module predicts navigable regions in the image space. These predictions are projected into a bird’s-eye view (BEV) cost map. Path fusion is then performed to identify coherent and safe traversable paths, followed by path selection based on alignment with the goal direction. Finally, the control module outputs linear and angular velocities to follow the selected path.

operate reliably across a wide variety of environments, including different terrains, lighting conditions, and sensor configurations, as illustrated in Figure 1. Domain shifts such as changing weather, unexpected obstacles, or cultural variations in street layout can easily degrade system performance if models have not learned invariant features or robust policies [5]. In our context here, we specifically refer to *generalization* as the ability of a system to operate robustly across (i) different terrain types imposed by testing in new locations, (ii) different visual conditions caused by weather and lighting variations, and (iii) different embodiments when deployed on distinct robotic platforms. This framing aligns with the “Visual + Behavioral” axes of generalization described in [6].

The Earth Rover Challenge (ERC) [7] serves as a rigorous testbed for evaluating precisely this kind of generalization. In this competition, identical sidewalk robots are deployed across **three continents and six countries**, spanning both urban and rural settings, and exposed to diverse real-world conditions, including variations in weather, and terrain (see Fig. 2). Each robot is a low-cost, four-wheeled platform equipped only with noisy sensors, including a front and rear RGB camera, an IMU, wheel encoders, and a GPS unit with limited accuracy, while lacking any onboard computation. All perception and control must run remotely over a 4G link, where visual data arrives at only 3–5 Hz and inertial data below 10 Hz, introducing substantial delays and packet loss. Missions consist of reaching a sequence of GPS-defined checkpoints located hundreds of meters to several kilometers apart, with a difficulty tier that determines the score awarded. Partial credit is granted for completing a subset of checkpoints, and teams may use a limited number of brief tele-operated interventions, which reduce the mission score.

These stringent constraints, including low-cost sensing, slow and unreliable communication, and the absence of onboard processing, render traditional SLAM-based mapping and planning pipelines difficult to apply effectively. Instead, systems must robustly identify traversable areas and plan under uncertainty with minimal information, while tolerating severe distribution shift across environments. The inaugural

2024 edition underscored this difficulty: the top-performing autonomous system achieved only 36% of the total possible score, far below the performance typically reported on standard academic benchmarks [7], [8].

To address these challenges, we propose GeNIE (Generalizable Navigation System for In-the-Wild Environments). An overview of GeNIE is shown in Fig. 3. At its core lies a robust traversability prediction model based on SAM2 [9], fine-tuned on a newly annotated dataset comprising 15,347 road scenes collected from diverse global environments. To efficiently produce training labels, we developed a semi-automatic annotation pipeline that combines SAM2-generated masks with lightweight human correction, significantly accelerating the annotation process while maintaining quality.

At inference time, the model outputs a pixel-wise traversability map, which is projected into a bird’s-eye-view cost map. From this map, random candidate paths are sampled and scored according to their traversability. However, naively selecting the path with the lowest cost often results in sub-optimal behavior due to the presence of multiple, similarly scored paths in noisy or ambiguous regions. To mitigate this, we introduce a simple yet effective path fusion strategy that merges nearby paths meeting traversability constraints. The final navigation path is then selected from this fused set, improving both stability and robustness.

Our system was deployed in the ERC ICRA 2025 competition and achieved **79% of the maximum score**, outperforming the second-best team by **17%**. In summary, our main contributions are as follows:

- We present **GeNIE**, a generalizable navigation system that demonstrates state-of-the-art performance across deployments in six countries spanning three continents, under diverse terrain types, road styles, weather conditions, and lighting. GeNIE achieves the highest performance to date in these challenging real-world settings, setting a new benchmark for generalizable navigation.
- We introduce a geographically diverse traversability annotation dataset comprising 15,347 road images from 20 countries. To the best of our knowledge, this is

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

the most geographically diverse dataset available for the traversability prediction task.

- We release a held-out benchmark for traversability prediction, consisting of data collected from 18 countries across 5 continents, featuring a wide variety of visual appearances and terrain styles. To the best of our knowledge, this is the first benchmark to evaluate traversability prediction under such diverse real-world scenarios.
- We have publicly released our codebase, pretrained model weights, annotated training dataset, and evaluation benchmark to support further research in real-world navigation.

II. RELATED WORK

A. Traversability Prediction

Recent work has advanced traversability prediction using self-supervised and vision-based methods. WVN [10] and WayFASTER [11] learn terrain affordances directly from RGB images using self-supervision and short-horizon motion cues, enabling fast adaptation in unlabeled wild environments. Jeon et al. [12] fuse geometry and appearance for vehicle-specific estimation, while Lambert et al. [13] rely on proprioception-derived slip to label off-road terrain without human supervision.

Danesh et al. [14] apply domain adaptation to transfer knowledge from simulation to real-world scenes. Probabilistic methods like [15] model uncertainty explicitly to improve safety on planetary rovers, and classical CNNs [16] remain strong baselines in structured off-road terrains.

However, most existing methods do not generalize well to unfamiliar terrains or degraded visual conditions. In contrast, our method leverages the pre-trained vision foundation model SAM2 [9] and a large, diverse dataset to achieve robust and generalizable traversability prediction, enabling deployment across drastically different settings without environment-specific retraining.

B. Trajectory Planning with Path Fusion

Sample-based planners like DWA [17] implicitly perform path fusion by simulating velocity-parameterized rollouts and selecting those optimizing safety and goal progress. Elastic Band [18] and TEB [19] refine a global path into smooth, tensioned corridors to mitigate local minima. Roll-out methods [20] simulate dynamics-aware trajectories in rough terrain, while BiC-MPPI [21] clusters bidirectional rollouts to reduce switching and enhance goal convergence.

Trajectory-level clustering has also been used for motion prediction [22] and socially compliant navigation [23], demonstrating that grouping geometrically or behaviorally similar paths improves planning robustness. Silhouette-guided clustering [24] is often used to tune the number of clusters, favoring precision in high-risk contexts.

Unlike these approaches, our method explicitly clusters candidate paths sampled from a BEV traversability map. We then fuse nearby paths and select the final trajectory based on heading alignment with the goal. This strategy mimics human

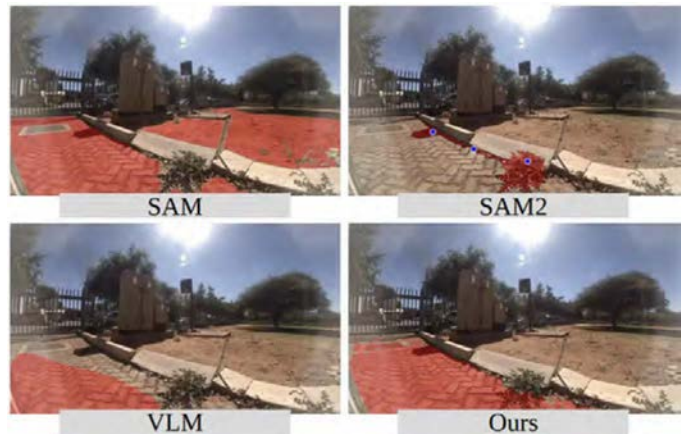


Fig. 4: Comparison of traversability predictions (red regions) from SAM, SAM2, VLM (gemini-2.5-flash), and our proposed approach. For SAM, a text prompt describing navigable regions is used directly. Since SAM2 does not support text prompts, we first query the VLM (gemini-2.5-flash) to identify relevant points in the image, which are then used as prompts for SAM2.

navigation heuristics and enhances stability and success in long-horizon outdoor planning.

III. METHOD

A. System Overview

Given an input RGB image, our system first predicts pixel-wise traversability scores in the image space. The predicted traversable regions are then projected into a local 2D bird’s-eye view (BEV) cost map. Multiple candidate paths are sampled from this BEV map. We then perform path fusion on the top- K paths with the lowest traversability costs to generate more stable and consistent trajectories. Finally, the best fused path is selected based on its alignment with the goal location. An overview of the system pipeline is illustrated in Figure 3.

B. Traversability Estimation

To robustly estimate traversability across diverse environments, we build upon the pretrained SAM2 model (sam2.1 hi-era tiny). However, naive application of SAM2 was inadequate for two main reasons. First, it tends to segment individual objects rather than the entire navigable surface, which often spans multiple visually similar regions. For example, in scenes where the flat ground is divided by markings or textures, SAM2 frequently outputs fragmented segments covering only parts of the traversable area. Second, SAM2 requires explicit point or bounding box prompts for inference and cannot directly generate masks based on the abstract concept of *traversable area*. While models like SAM support text prompts, they perform poorly on traversability segmentation tasks (see Figure 4).

Due to the lack of existing datasets for traversability segmentation across diverse terrains, we developed a semi-automatic annotation pipeline built on the FrodoBot dataset. Our data collection spans outdoor scenes from seven countries, covering a wide range of environments including grasslands,

urban parks, deserts, and rural roads. To support scalable annotation, we used SAM2-generated region proposals as initial masks, which human annotators refined using a lightweight graphical interface. This process enabled the creation of 15,347 high-quality annotated frames with minimal manual effort.

A key feature of our dataset is that each traversable mask includes only regions that are directly navigable from the robot’s current position, typically located near the bottom center of the image. This constraint ensures that the model learns to predict not just static free space, but the subset of the scene that is immediately reachable, thereby making the learned traversability more actionable for downstream planning (Figure 4)

To eliminate the reliance on manually provided prompts during inference, we removed the prompt encoder and instead directly optimized a learnable prompt token. Conceptually, this token encodes the notion of a “traversable” or “navigable” area. We refer to this modified version of SAM2 as **SAM-TP** (SAM for Traversability Prediction). All SAM-TP models are initialized with SAM2 pretrained weights [9] and trained on our annotated dataset using the Adam optimizer with a batch size of 20, a learning rate of 1×10^{-6} , and 10 epochs. Training was performed on a single A100 GPU and required less than 4 hours.

- **SAM-TP (D1)**: Fine-tunes only the last block of the two blocks in the mask decoder.
- **SAM-TP (D2)**: Fine-tunes all blocks of the mask decoder.
- **SAM-TP (E6)**: Fine-tunes all layers of the mask decoder and the last 6 of the 12 blocks in the image encoder.
- **SAM-TP (Ours)**: Fine-tunes all layers of both the mask decoder and the image encoder.

To more thoroughly assess generalization capabilities, we collected an additional set of 484 images across 17 countries, encompassing a wide range of terrain types, weather conditions, and lighting scenarios. All images in this benchmark were excluded from training and annotated using our semi-automatic pipeline. This dataset provides a benchmark for evaluating model performance under realistic conditions. We organize the benchmark into three categories:

- **Normal (N)**: Contains 167 scenes captured under typical conditions, including mild lighting and clearly defined paths. This set serves as the baseline for model performance.
- **Difficult (D)**: Comprises 160 scenes featuring challenging conditions commonly observed in the Earth Rover Challenge, such as uneven terrain, heavy rain, strong shadows, overexposure, occlusions, and visually fragmented surfaces caused by markings or textures. This set is designed to assess the model’s robustness to real-world variability.
- **Cross-embodiment (CE)**: Includes 157 scenes collected using a different robot platform (FrodoBot Mini), which differs significantly in height, aspect ratio, camera

Method	IoU across three datasets \uparrow		
	Normal	Difficult	CE
WVN [10]	0.6157	0.4684	0.4551
SAM [25]	0.8759	0.7270	0.7610
VLM [26]	0.6528	0.5251	0.6206
SAM-TP (single scene)	0.8662	0.6867	0.8230
SAM-TP (20% dataset)	0.9120	0.7503	0.8889
SAM-TP (50% dataset)	0.9237	0.7758	0.9071
SAM-TP (D1)	0.8337	0.6572	0.8215
SAM-TP (D2)	0.8480	0.6843	0.8375
SAM-TP (E6)	0.9248	0.7974	0.9256
Ours	0.9289	0.7989	0.9262

TABLE I: Zero-shot segmentation accuracy on our test dataset. The table reports the average Intersection over Union (IoU) across three evaluation conditions: normal, difficult, and cross-embodiment (CE). It compares baseline methods, our proposed approach trained with different dataset subsets, and various fine-tuning strategies. “SAM-TP (D1)” and “SAM-TP (D2)” denote partial fine-tuning of the mask decoder; “SAM-TP (E6)” includes additional encoder layers; and “Ours” fine-tunes the entire image encoder and mask decoder.

placement, and optical characteristics compared to the FrodoBot Zero used during training. This set evaluates the model’s ability to generalize across embodiments.

In addition to evaluating variants of the SAM-TP model, we also benchmarked several baselines using our proposed test set:

- **WVN [10]**: The traversability prediction model introduced in [10].
- **SAM [25]**: The original SAM model prompted with a text description of the navigable region.
- **VLM gemini-2.5-flash [26]**: A vision-language model, gemini-flash-2.5, which directly generates segmentation masks in response to textual queries.
- **SAM-TP (single scene)**: SAM-TP fine-tuned on a dataset collected from a single environment.
- **SAM-TP (20%)**: SAM-TP fine-tuned using 20% of the full training dataset.
- **SAM-TP (50%)**: SAM-TP fine-tuned using 50% of the full training dataset.

The benchmark results are summarized in Table I. The **Difficult** test set presents significant challenges for all models, highlighting the complexity of real-world conditions. Among all methods, the proposed **SAM-TP** achieves the highest performance, demonstrating strong generalization across diverse environmental conditions and robotic embodiments.

C. Bird’s-Eye View Map Generation

Due to the lack of a depth camera on our robotic platform, we could not construct the bird’s-eye view (BEV) map by directly projecting point clouds, as is commonly done. While several monocular depth estimation models exist [27], [28], we

found that their predictions are insufficiently accurate for our application. Inaccurate depth estimates can result in critical failures such as collisions or the robot falling into a trench.

To overcome this, we adopt a simple yet effective approach that leverages the segmented traversable regions and a known camera height prior, which is a parameter typically available in robot systems. Instead of relying on the standard pinhole camera model, which fails under severe lens distortion, we use a generic camera model [29] to calibrate the camera. This calibration yields a directional ray vector \hat{d} for each pixel in the image.

For any pixel classified as traversable, we assume the corresponding 3D point lies on the ground plane. Given the known camera height h , we solve for the scale s in the equation: $s\hat{d}_y = -h$. This allows us to recover the 3D position of the pixel in the camera coordinate frame as $s\hat{d}$. These points are then projected into the BEV map.

D. Path Planning and Fusion

Given the BEV cost map, the robot must predict a safe and traversable path that ideally leads toward the goal. A common strategy is to use sample-based planning [10], where candidate paths are sampled from the robot’s current position toward the goal—or toward intermediate goals if the final goal is far away. The optimal path is then selected by minimizing a cost function of the form:

$$C(l) = \sum_{i=1}^n f(p_i) + \beta \sum_{i=1}^n g(p_i) \quad (1)$$

where p_i is the i -th waypoint along the sampled path, $f(\cdot)$ is the cost derived from the traversability map, and $g(\cdot)$ is a goal-directed cost that encourages progress toward the target.

However, this strategy often performs poorly when the goal lies hundreds of meters away. Two main issues arise:

- 1) **Difficulty balancing traversability and goal costs:** When the goal-directed term $g(\cdot)$ dominates the cost function, the robot tends to move greedily toward the goal while ignoring environmental structure. This often results in myopic behavior and causes the robot to fall into local minima. Conversely, if the traversability cost $f(\cdot)$ dominates, the robot may ignore the goal and choose suboptimal paths that appear easier to traverse but lead in the wrong direction. In practice, there is no single value of β that works robustly across all scenarios, and adaptively tuning β at runtime is not feasible. Representative failure cases are shown in Figure 5.
- 2) **Path switching due to cost equivalence:** In real-world deployments, robots typically employ a receding horizon strategy, continuously replanning as new observations become available. However, when many candidate paths have similar costs, the robot may frequently switch between them, leading to unstable behavior and slow overall progress.

Tuning the weight β between the goal-directed term and the traversability cost does not resolve these issues, as different

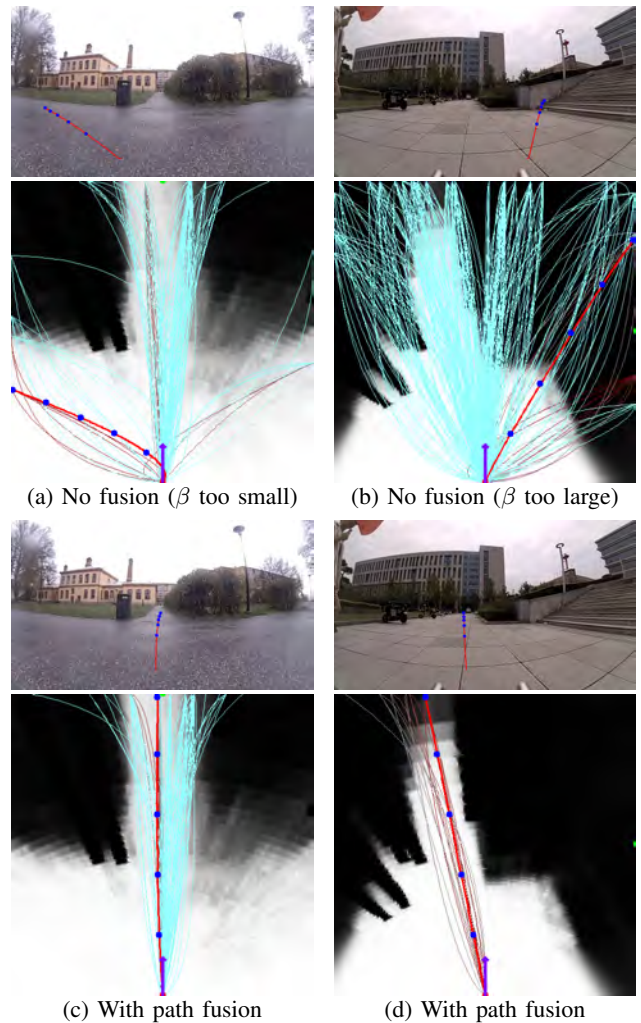


Fig. 5: Comparison of failure and success cases under different path planning strategies. The first and third rows show the robot’s current observation overlaid with the projected planned trajectory. The second and fourth rows display the BEV cost map with sampled paths; the selected path is shown in red, and the green dot at the edge of the map indicates the goal direction. Without path fusion, the planner tends to select paths with lower traversability when β is small, or gets stuck in local minima when β is large. In contrast, path fusion succeeds in both scenarios.

situations require different trade-offs. Ideally, the robot should adaptively balance these objectives based on the current context. As shown in Figure 6, the success rate of path selection under this naive strategy remains low across a wide range of β values.

In contrast, humans rarely plan paths at the pixel level. Instead, we abstract navigable structure from visual inputs and reason at a higher level: along paths or tracks [30]. Whether following an urban road or an informal dirt path flanked by grass, we tend to follow the path until a junction appears, and only then do we make directional decisions. This behavior reflects a strong prior: if a path leads generally toward the

goal, it is likely to be successful. This prior holds in many real-world outdoor settings.

To incorporate this prior, we propose a simple but effective planning method that operates at the path level. Given the BEV cost map, we first sample candidate paths from the robot’s current position to intermediate goals within its current field of view. These paths are parameterized as first- and second-order polynomials connecting the start and end points. We then filter these candidates using the BEV cost map and retain the top N lowest-cost paths.

To detect the traversable paths, we apply **path fusion** to the top N candidate paths. We first cluster these paths using adaptive k -means, where clustering is based on the Euclidean distance between corresponding waypoints across different paths. The number of clusters k is determined by optimizing the silhouette loss, which balances intra-cluster compactness and inter-cluster separation.

The silhouette loss is defined as the negative of the average silhouette coefficient across all data points:

$$\mathcal{L}_{\text{silhouette}} = -\frac{1}{N} \sum_{i=1}^N \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (2)$$

where $a(i)$ is the average distance between sample i and all other points in the same cluster, and $b(i)$ is the minimum average distance between sample i and points in the nearest neighboring cluster. A higher silhouette coefficient indicates better clustering quality, and minimizing the negative value guides the model toward an optimal number of clusters.

Since minimizing silhouette loss tends to favor finer-grained clusters, we perform a post-processing step to merge any clusters whose centroids lie within a predefined distance threshold. The centroid of each merged group is used as the final representative path.

We define a *false positive (FP) merge* as the erroneous merging of distinct paths that should remain separate, e.g., merging two paths split by an obstacle, potentially leading to unsafe planning. Conversely, a *false negative (FN) merge* occurs when two similar paths are not merged despite being part of the same structure. While FPs can lead to critical failures (e.g., collisions), FNs simply preserve redundant safe paths. Hence, we prefer **precision over recall** in determining the merging threshold. A Precision–Recall (PR) curve illustrating this trade-off is shown in Figure 6 (a).

After path fusion, if only a single representative path remains, we select it as the final plan. If multiple paths remain, we choose the one most aligned with the goal direction.

To evaluate the effectiveness of our path fusion and selection approach, we curated a test set consisting of road images with varying numbers of path branches, annotated by human raters. We use the following metrics in our empirical study:

- **Path Identification Precision (PIP):** The proportion of correctly merged path branches (true positives) out of all predicted path merges.

Method	PIP (%) ↑	PIR (%) ↑	PSA (%) ↑
No Fusion ($\beta = 0.1$)	-	-	27.3
No Fusion ($\beta = 1$)	-	-	42.9
No Fusion ($\beta = 10$)	-	-	36.4
k-means Only	75.0	75.0	61.9
Euclidean Selection	84.6	57.9	71.4
Ours	84.6	57.9	85.7

TABLE II: Evaluation of path fusion and selection strategies with Path Identification Precision (PIP), Path Identification Recall (PIR), and Path Selection Accuracy (PSA).

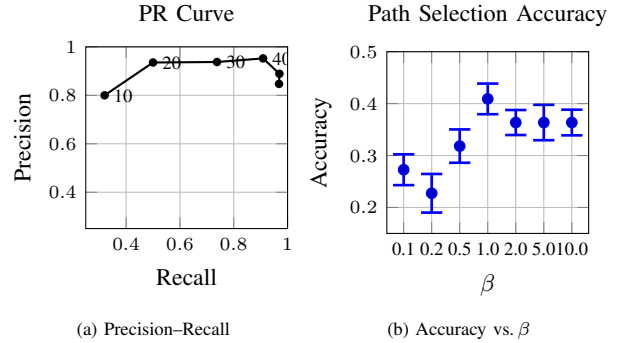


Fig. 6: (a) Precision–Recall curve for different merging threshold for path identification (b) Path selection accuracy for No Fusion, as a function of β .

- **Path Identification Recall (PIR):** The proportion of correctly identified path branches (true positives) out of all ground truth two-branch cases.
- **Path Selection Accuracy (PSA):** The percentage of cases where the selected path matches the human-annotated correct branch.

We evaluate several variants of our pipeline:

- **No Fusion (NF):** Directly selects the lowest-cost path from the sampled set using the cost function in Equation 1.
- **k-Means Only (KM):** Clusters paths using k-means, without any post-processing or merging.
- **Euclidean Selection (ES):** Uses k-means + merge, then selects the path whose endpoint is closest (Euclidean distance) to the goal.
- **Angular Selection (Ours):** Uses k-means + merge, then selects the path with heading direction most aligned with the goal direction.

The results are summarized in Table II. The **no-fusion** strategy yields the lowest path selection accuracy, and tuning the weight β does not improve performance, as shown in Figure 6(b). While using k-means alone results in the highest path identification recall due to the adaptive k-means assigning more clusters, it suffers from low path identification precision and lower path selection accuracy compared to strategies that include a merging step.

Algorithm 1: Path Planning with Path Fusion

Input: Random paths $\mathcal{T} = \{l_1, l_2, \dots, l_M\}$, GPS goal \mathbf{g} , RGB image I , traversability model ϕ

Output: Selected trajectory l^*

- 1 $m = \text{ProjectToBEV}(\phi(I))$; // BEV cost map
- 2 $\mathcal{T}_K \leftarrow \text{TopK}_{l_j \in \mathcal{T}} [\sum_i m(p_i)]$; // Top-K paths
- 3 $\mathcal{C} \leftarrow \text{AdaptiveKMeans}(\mathcal{T}_K)$; // Clustering
- 4 $\mathcal{C}_{\text{merged}} \leftarrow \text{MergeCloseCentroids}(\mathcal{C})$
- 5 **if** $|\mathcal{C}_{\text{merged}}| = 1$ **then**
- 6 $l^* \leftarrow \text{Centroid}(\mathcal{C}_{\text{merged}})$
- 7 **else**
- 8 $l^* \leftarrow \arg \min_{l \in \mathcal{C}_{\text{merged}}} \mathcal{L}(l, \mathbf{g})$
- 9 **return** l^*

Among the merging-based strategies, selecting the path whose heading direction is most aligned with the goal direction achieves the highest path selection accuracy. This result aligns with intuitive human path planning behavior, where a balance between feasibility and goal alignment is naturally considered. Some of the visualized path fusion results are shown in Figure 5

Putting everything together, our planning algorithm is summarized in Algorithm 1. In our implementation, the random path set is generated only once at initialization, and only cost evaluations are updated during each planning cycle—making the approach computationally efficient.

E. Closed-loop Control

The control module implements a simple waypoint-tracking policy that prioritizes robustness over efficiency. Our system operates in a receding-horizon loop: at each timestep, the latest observation is used to replan a path via Algorithm 1, and the robot then executes the trajectory toward the first waypoint before repeating the process with updated input. A collision-avoidance mechanism runs continuously in parallel, monitoring the predicted traversability map from each incoming frame to halt motion if a nearby hazard is detected. Concretely, after each planning step the robot aligns its heading with the next waypoint, and then advances by 1 m, unless the collision-avoidance module signals low traversability in the near-front region. Since our SAM-TP model runs at 10 Hz on an RTX 3090 GPU while frames arrive at only 3–5 Hz, collision checks can be executed in real time without adding latency. The planning module itself runs at approximately 5 Hz. Although this synchronous replan–align–move cycle is less efficient than continuous controllers such as DWA, it was deliberately chosen for robustness under the high-latency conditions of ERC.

In some cases, the planner may guide the robot into dead ends such as dense grass, enclosed areas, or regions near obstacles, where it fails to find a valid path. To handle these situations, we introduce a vision-language model (VLM)-assisted recovery module. Given a short history of recent observations, the VLM is prompted with a natural language

Team	Total Score \uparrow	Norm. Score \uparrow	Interventions \downarrow
Human Player	30.00	1.00	-
GeNIE (ours)	23.78	0.79	0
Team A	18.74	0.62	1
Team B	8.07	0.27	11
Team C	7.39	0.25	2
Team D	5.89	0.20	6

TABLE III: Leaderboard summary of the top 5 teams from the challenge. The normalized score is computed relative to the maximum possible score of 30.

query on how to return to the previously navigable route. This allows the robot to recover from local failures using high-level semantic reasoning. Please see the online supplementary material for further details.

IV. ERC COMPETITION RESULTS

The GeNIE system was deployed in the 2025 Earth Rover Challenge (ERC) across Africa, South America, and North America. In each mission, the robot, equipped with front and rear RGB cameras, IMU, wheel encoders, and GPS, was tasked with reaching a sequence of GPS-defined checkpoints over distances of several hundred meters to a few kilometers. Scoring was based on mission difficulty and the number of checkpoints completed, with penalties applied for human interventions, and a score of zero if the robot flipped over. The overall competition setup and scoring rules are described in Section I and in [7].

We report the following evaluation metrics:

- **Total Score:** The cumulative score earned by the team, calculated according to the scoring rules described above.
- **Normalized Score:** The total score divided by the maximum possible score (30), providing a percentage-based measure of performance.
- **Interventions:** The total number of human interventions used across all missions in which the team score was non-zero¹

Our system achieved **79%** of the maximum possible score, outperforming the second-place team by 17%. Full results are provided in Table III. Notably, our system required zero human interventions, demonstrating its robustness and adaptability across challenging and varied environments.

V. CONCLUSION AND LIMITATIONS

In this paper, we present **GeNIE**, a navigation system capable of generalizing across a wide range of real-world environments. The key to this generalization lies in two components: (1) a robust traversability prediction model, and (2) an effective path planning strategy based on path fusion. Our system demonstrated strong performance in the Earth Rover

¹This differs from the official count which is total number of interventions across all missions, regardless of score. This better reflects the model’s ability to navigate since zero scores indicate a non-recoverable failure (e.g., the robot flipping over).

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

Challenge (ERC), achieving first place across deployments in six countries.

GeNIE is not without limitations. The system failed to complete three missions during ERC: one mission resulted in a zero score due to a local minimum problem, and two missions yielded only partial scores due to inefficient control under high latency.

The local minimum problem arises because the system relies on single-frame traversability predictions without maintaining a local map, which limits its ability to reason over longer horizons. In Mission 9, for example, the robot refused to enter a passage with low predicted traversability, not realizing it was the only viable route to the target. Such cases suggest that effective traversability prediction may require persistent memory and long-term reasoning, enabling the robot to integrate past observations and replan accordingly.

A second limitation is the conservative control strategy adopted to ensure robustness under high-latency conditions. The robot executes commands in a sequential “observe–plan–execute–wait” cycle rather than continuously re-planning. While this avoids overshooting behaviors, it reduces efficiency and caused timeouts in two missions. Incorporating more latency-aware or hybrid control strategies could improve efficiency without compromising robustness.

Finally, our BEV projection assumes a locally flat ground plane, which is generally valid within the 8 m planning horizon but may introduce errors on steep slopes. Although this did not cause failures in ERC, integrating depth sensing or monocular depth estimation could further improve reliability. Another limitation is that the current representation uses a single traversability token, which may be insufficient for robots with fundamentally different embodiments (e.g., legged versus wheeled).

Addressing these limitations by incorporating spatial memory for long-horizon reasoning, improving control strategies under latency, and relaxing the flat-ground assumption represents an important direction for future work.

ACKNOWLEDGEMENTS

This research / project is supported by A*STAR under its National Robotics Programme (NRP) (Award M23NBK0053).

REFERENCES

- [1] Earth Rover Challenge, “ERC’25 Recording,” <https://www.youtube.com/watch?v=LN99BnqhIdg>, 2025.
- [2] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, “Habitat: A platform for embodied ai research,” *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9339–9347, 2019.
- [3] F. Xia, A. R. Martin, A. Naveen, and et al., “Gibson env: Real-world perception for embodied agents,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9068–9079, 2018.
- [4] S. K. Ramakrishnan, U. Jain, E. Wijmans, Y. Li, P. Anderson, and D. Batra, “Habitat-matterport 3d dataset (hm3d): 1000+ large-scale 3d environments for embodied ai,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [5] F. Xia, R. Mottaghi, S. Savarese et al., “Generalization in embodied navigation: A survey,” *Foundations and Trends in Robotics*, vol. 11, no. 3, pp. 162–195, 2023.
- [6] J. Gao, S. Belkhale, S. Dasari, A. Balakrishna, D. Shah, and D. Sadigh, “A taxonomy for evaluating generalist robot policies,” *arXiv preprint arXiv:2503.01238*, 2025.

- [7] “The EarthRover Challenge - 2025 — earth-rover-challenge.github.io,” <https://earth-rover-challenge.github.io/>, [Accessed 29-05-2025].
- [8] X. Xiao, J. Tan, M. Cho, D. Hsu, D. Shah, J. Truong, T. Xiao, N. Yokoyama, W. Yu, T. Zhang et al., “Autonomous mario kart in the wild: Lessons learned from the earth rover challenge at iros 2024 [competitions],” *IEEE Robotics & Automation Magazine*, vol. 32, no. 2, pp. 189–196, 2025.
- [9] A. Kirillov, K. He, and P. Dollár, “Segment anything model 2 (sam2),” <https://segment-anything.com>, 2024, technical report, Meta AI.
- [10] J. Frey, M. Mattamala, N. Chebrolo, C. Cadena, M. Fallon, and M. Hutter, “Fast Traversability Estimation for Wild Visual Navigation,” in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.
- [11] M. V. Gasparino, A. N. Sivakumar, and G. Chowdhary, “Wayfaster: A self-supervised traversability prediction for increased navigation awareness,” in *ICRA*, 2024.
- [12] Y. Jeon, E. I. Son, and S.-W. Seo, “Follow the footprints: Self-supervised traversability estimation for off-road vehicle navigation based on geometric and visual cues,” *RA-L*, 2024.
- [13] A. Lambert, B. Boots, and J. Lee, “Learning off-road terrain traversability with self-supervisions only,” in *CoRL*, 2023.
- [14] M. Danesh et al., “Terrain traversability prediction through self-supervised learning and unsupervised domain adaptation,” *Autonomous Robots*, 2024.
- [15] M. Endo, T. Taniai, and G. Ishigami, “Deep probabilistic traversability with test-time adaptation for uncertainty-aware planetary rover navigation,” *RA-L*, 2024.
- [16] D. Lee et al., “Deep learning traversability estimator for mobile robots in unstructured environments,” in *IROS*, 2021.
- [17] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [18] S. Quinlan and O. Khatib, “Elastic bands: Connecting path planning and control,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1993, pp. 802–807.
- [19] C. Rösmann, F. Hoffmann, and T. Bertram, “Integrated online trajectory planning and optimization in distinctive topologies,” *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.
- [20] G. Sten, L. Feng, and B. Möller, “An efficient trajectory roll-out algorithm for autonomous articulated vehicles in forest terrain,” *Preprint*, 2025.
- [21] M. Jung and K. Kim, “Bic-mppi: Goal-pursuing, sampling-based bidirectional rollout clustering path integral for trajectory optimization,” *arXiv preprint arXiv:2410.06493*, 2024.
- [22] C. Sung, D. Feldman, and D. Rus, “Trajectory clustering for motion prediction,” in *IEEE International Conference on Robotics and Automation*, 2012.
- [23] E. Pruc, S. Zilberstein, and J. Biswas, “Dense crowd flow-informed path planning,” *arXiv preprint arXiv:2206.00705*, 2022.
- [24] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
- [25] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo et al., “Segment anything,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.
- [26] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican et al., “Gemini: a family of highly capable multimodal models,” *arXiv preprint arXiv:2312.11805*, 2023.
- [27] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, “Depth anything v2,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 21 875–21 911, 2024.
- [28] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun, “Depth pro: Sharp monocular metric depth in less than a second,” *arXiv preprint arXiv:2410.02073*, 2024.
- [29] T. Schops, V. Larsson, M. Pollefeys, and T. Sattler, “Why having 10,000 parameters in your camera model is better than twelve,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2535–2544.
- [30] J. Balaguer, C. Summerfield, D. Hassabis, and H. J. Spiers, “Neural mechanisms of hierarchical planning in a virtual subway network,” *Neuron*, vol. 90, no. 4, pp. 893–903, 2016.