LiteNav: Lightweight Map-free Outdoor Visual Navigation

Mingxu Zhu,* Chang Liu,* Xiao Zhao, Zheyuan Zhang, Linna Song
Qingliang Luo, Liming Zhang, Xiangyu Zhen, Chufan Guo, Kuifeng Su
ADLab, Tencent
{mingxuzhu, changeliu, shawzhao, lukezyzhang, linnaasong}@tencent.com
{fineluo, leolmzhang, xiangyuzhen, chufanguo, kuifengsu}@tencent.com

Abstract

Outdoor robot navigation in dynamic, unknown environments remains a formidable challenge, necessitating robust traversability assessment and collision-free planning. While traditional map-based approaches struggle with adaptability to novel scenes, existing map-free methods frequently incur high computational costs and depend heavily on LiDAR data. In response, we propose LiteNav, a vision-only for exteroceptive sensing and strictly map-free outdoor navigation system. Relying solely on a single off-the-shelf RGB camera and GPS for goal specification, LiteNav estimates traversability, encodes navigation goals, and generates candidate trajectories via a diffusion-based joint encoding model. These trajectories are then evaluated by a goal-oriented planner via high-frequency reactive replanning to balance goal progress and local obstacle avoidance. Remarkably, LiteNav outperforms LiDAR-equipped baselines on the open-loop GND benchmark without constructing or referencing explicit environmental maps. Deployed on a low-power NVIDIA Jetson Xavier NX, it operates in real time at 10 Hz, consuming only one-fifth of the computational resources required by prior approaches. Experiments demonstrate that LiteNav achieves state-of-the-art open-loop results on the GND dataset and outperforms existing vision-based methods in real-world closed-loop scenarios. In summary, LiteNav is a map-free outdoor navigation system that operates using monocular RGB perception and GPS goal cues on an embedded platform, demonstrating strong potential for scalable deployment.

1 Introduction

Robot navigation is a cornerstone of autonomous robotic systems, enabling agents to plan paths from start to goal using onboard sensors and execute autonomous locomotion. Outdoor robot navigation—a canonical yet highly challenging scenario—is characterized by dynamic, unknown, and highly complex environments. These conditions pose significant technical hurdles in trajectory generation and navigation, including traversability estimation, collision-free trajectory planning, and path optimality.

A representative class of outdoor robot navigation approaches relies on map-based methodologies, which involve pre-collecting and constructing a map of the operational environment [16, 4, 19, 3, 17, 20, 15, 21, 23, 24, 6, 13, 2, 5]. Such maps may take the form of high-precision LiDAR point-cloud maps [3, 4, 15] or purely image-based Image Topology Graphs (ITGs) [21, 23]. While map-based approaches inherently resolve traversability assessment and route selection priors—allowing the navigation algorithm to focus primarily on collision avoidance and optimal path planning—they are fundamentally constrained by their dependence on pre-built maps. This dependency renders them inapplicable in novel, previously unseen environments and incapable of achieving true "out-

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

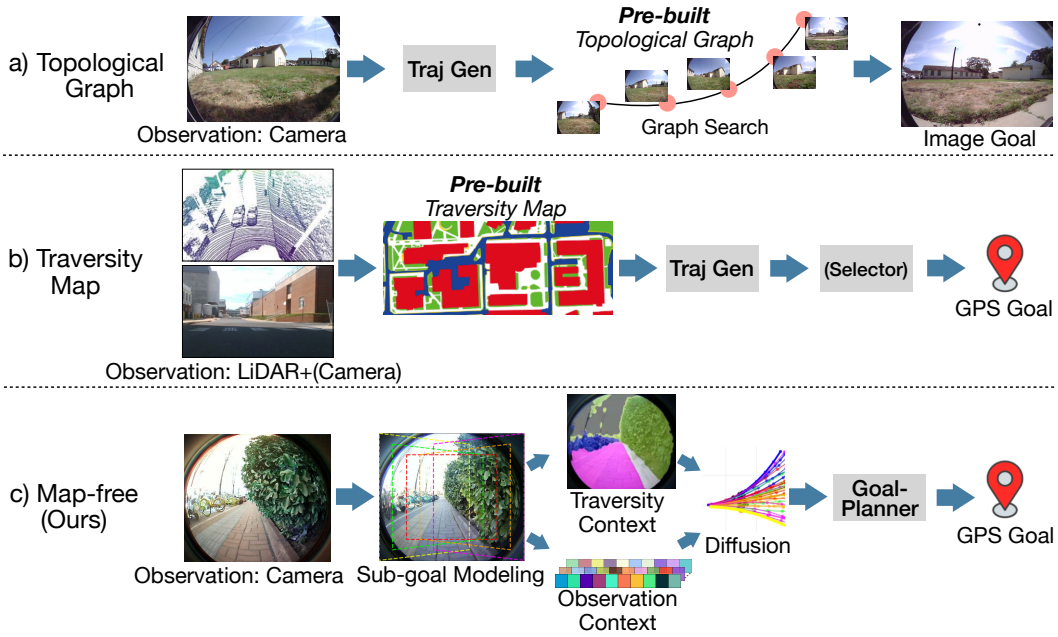

Figure 1: We propose LiteNav, a lightweight, map-free navigation framework. LiteNav achieves robust point-to-point navigation in unknown outdoor environments using monocular RGB for scene perception and GPS for global goal specification, without LiDAR or pre-built graphs/maps.

of-the-box" functionality. Moreover, issues related to map freshness and accuracy further limit their suitability for general-purpose outdoor robotic navigation [14].

Driven by the growing demand for general-purpose outdoor navigation capabilities, a new wave of "mapless" navigation methods has recently emerged [22, 10, 9, 27]. These approaches are designed specifically for operation in unknown environments without requiring any prior mapping. Consequently, beyond addressing collision avoidance and optimal trajectory generation, they must also tackle the additional challenge of traversable region identification. Although these mapless methods have made notable progress toward enhancing the generality of outdoor robot navigation, a considerable gap remains before practical deployment can be realized. This gap manifests in two primary aspects: (1) **Substantial computational overhead**, which typically necessitates high-end hardware configurations—such as equipping robots with LiDAR as the primary sensor and employing x86-based computing platforms with NVIDIA GTX 1080 [10] or even RTX 3060 GPUs [12]—thereby imposing significant constraints on robot payload capacity, power consumption, and operational endurance; and (2) **Reliance on hidden mapping**, as these methods are not entirely map-free in practice. Many still rely on LiDAR-based LIO mapping systems to provide ground truth during the training phase [10, 9, 22]. As such, their feasibility for mass-deployable, genuinely map-free outdoor navigation remains limited.

Motivated by the above observations, we pose the following question: Is it possible to devise a genuinely map-free, low-cost, and scalable solution tailored for outdoor robot navigation in unknown environments? To address this challenge, we propose **LiteNav**, a purely vision-based map-free outdoor navigation framework. LiteNav performs traversability estimation directly from images, encodes navigation goals to influence trajectory generation, and employs a joint-encoding based diffusion model to produce candidate trajectories. Furthermore, we design a goal-oriented planning module that leverages high-frequency reactive replanning over a short local horizon to promote goal-directed progress and collision avoidance.

LiteNav operates in a strictly map-free manner: it requires zero pre-computed physical or topological maps of the deployment environment, constructs no LiDAR-based traversability maps during training, and maintains no persistent spatial memory during inference. Notably, using only a single off-the-shelf RGB camera, LiteNav outperforms state-of-the-art (SOTA) LiDAR-equipped methods on the open-loop GND benchmark and surpasses vision-centric baselines in closed-loop field trials.

In summary, our contributions are as follows:

• We present LiteNav, to our knowledge the first lightweight and cost-effective map-free outdoor navigation system that enables GPS-guided point-to-point autonomous navigation using only a single wide-angle RGB camera, achieving genuine map-free operation. Crucially, no map—be it LIO-based, ITG-based, or SLAM-derived—is constructed at any stage of the pipeline. Moreover, LiteNav runs at 10 FPS on a low-power embedded NVIDIA Jetson Xavier NX platform (which provides about one-twelfth the computational performance and consumes approximately one-eleventh the power of an RTX 3060), demonstrating high real-time performance under resource-constrained conditions.

• We introduce a two-stage architecture: the first stage generates candidate trajectories by integrating semantic traversability constraints into a goal-conditioned diffusion model, and the second stage selects the optimal trajectory via a physics-aware planner.

• We propose a reactive planning module in the Frenet coordinate frame, which employs a sub-goal sampler to jointly evaluate short-horizon trajectories based on traversability and goal alignment, while providing oscillation recovery to handle local minima and dynamic disturbances reactively.

• We demonstrate strong empirical performance: LiteNav improves traversability by 17.9% and distance ratio by 7.1% over LiDAR and vision baselines on the public open-loop GND benchmark, and further validates its effectiveness against vision-based baselines (ViNT and NoMaD) in real-world closed-loop experiments.

2 Related work

2.1 Map-based outdoor navigation

Outdoor robot navigation faces technical challenges in generating navigable paths within constrained traversable areas [14]. For instance, in complex outdoor road scenarios, robot navigation paths must avoid non-traversable regions such as bushes and buildings, as well as hazardous or prohibited zones like motor vehicle lanes. Traditional mapping-based (map-based) approaches address this by pre-constructing spatial maps of target operational areas to identify traversable and non-traversable regions, thereby transforming the outdoor navigation problem into a path optimization task within constrained regions [3, 4, 15]. While SLAM-based mapping methods achieve highly accurate 3D spatial representations, not all outdoor applications have conditions suitable for high-precision map construction [9].

Image-based topology graphs, on the other hand, represent a newer class of mapping techniques with relatively lower precision requirements [21, 23, 24, 6]. This approach abstracts the robot's operational area into a topological graph expressed through image features, where nodes represent traversability relationships. During navigation, the robot plans trajectories by matching current observations with graph nodes. A similar concept employs Gaussian Splatting for topological map construction, as demonstrated in [13, 2]. ViKiNG [17] further proposes visual representations using imprecise satellite or road network maps. Regardless of representation format—whether SLAM point clouds, image-based topology graphs, Gaussian-based topology graphs, or satellite/road maps—these methods share a fundamental dependency on pre-mapping, limiting their practicality in novel scenes, unseen environments, or scenarios lacking pre-mapping infrastructure. This work proposes a map-free navigation framework that eliminates the need for pre-constructed maps of any representation format, enabling plug-and-play navigation capabilities in unknown, novel outdoor environments.

2.2 Map-free outdoor navigation

Conversely, map-free outdoor navigation methods explore approaches independent of pre-mapping. DTG [10] proposes an end-to-end map-agnostic trajectory generation method using diffusion models, employing conditional recurrent neural network diffusion units and adaptive traversability-constrained training strategies to generate global navigation trajectories directly from environmental observations. MTG [9] introduces an end-to-end trajectory generation framework based on conditional variational autoencoders (CVAE), incorporating traversability constraints to ensure comprehensive coverage of navigable directions. Building on MTG, VL-TGS [22] integrates a visual landmark module that projects and renders multiple candidate trajectories generated by CVAE onto images, then selects optimal trajectories using vision-language models (VLM) guided by task-oriented prompts. However, these map-free methods face two common limitations: (1) They are not truly map-free, as all require 3D traversability maps constructed using LIO-SAM [10, 9, 22] during training; (2) Their hard-

ware/software architectures are impractical for real-world deployment, requiring resource-intensive components like 16-channel LiDAR [10, 9, 22] and desktop-level GPUs (e.g., GTX 1080 [10], 2080 [22]) on x86 architectures[1]. Addressing practical mass-production requirements for robot outdoor navigation, this work proposes an ultra-lightweight solution using only a consumer-grade RGB camera and the robot's onboard NX computing platform, eliminating map construction requirements during both training and deployment.

3 Method

In this section, we present the LiteNav method. We first formally define the map-free navigation problem and provide an overview of the overall methodology in section 3.1. Subsequently, we detail our core algorithmic modules: Section 3.2 describes semantic-guided encoding of robot-traversable regions; Section 3.3 introduces diffusion-based trajectory generation; Section 3.4 presents goal-oriented trajectory planning via sub-goal sampling; Section 3.5 explains end-to-end ground truth construction; and Section 3.6 outlines the loss functions used during model training.

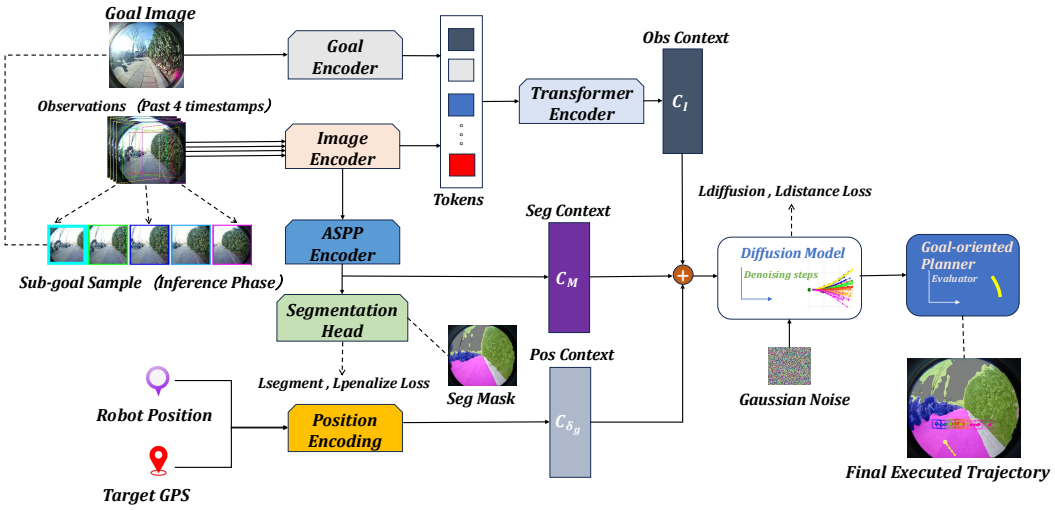

Figure 2: LiteNav architecture overview. LiteNav integrates semantic traversability estimation with a goal-conditioned diffusion model to generate diverse candidate paths, subsequently evaluated by a goal-oriented planner using sub-goal sampling to ensure safe and efficient outdoor navigation.

3.1 Problem formulation and method overview

The task of map-free outdoor robot navigation can be defined as follows: without relying on any prior map information of the environment, the robot must plan a path strategy based solely on its local observations to reach a target goal efficiently and safely in complex outdoor scenes. Our objective is to design a control policy π that maps the robot's current and historical observations ($\mathbf{o}_t := o_{t-T:t}$), its current and historical states ($\mathbf{s}_t := s_{t-T:t}$), and the pose goal residual $\delta_g = \text{diff}(g_t, g_o)$ to appropriate actions, enabling the robot to progressively move toward and eventually arrive at the target goal pose g_t. Here, g_o denotes the robot's current pose and g_t denotes the target goal pose, both expressed in a global frame (e.g., ENU); Algorithm 1 uses the same symbols. Importantly, LiteNav does not perform online state estimation (i.e., ego-pose, odometry, or precise yaw) from RGB images and GPS inputs; instead, the system directly ingests global ENU coordinates and coarse heading outputs from a standard consumer-grade GPS module (δ_g), encoding this relative pose differential purely as a conditioning vector to provide coarse global directional guidance. The design of π centers on two key components: (1) identification of locally traversable regions, and (2) trajectory generation under the joint constraints of target goals and local traversability. Additionally, to align with the practical goal of lightweight, mass-deployable outdoor navigation systems, we restrict the robot's exteroceptive

[1]To clarify, our emphasis on "real-time" capabilities excludes stacking x86 computers with desktop-grade NVIDIA GPUs on robots. Instead, we focus on embedded platforms like Orin or NX—low-power, compact, cost-efficient, and with reduced hardware weight burden—ensuring technical feasibility for mass-produced robots.

sensing modality to a monocular RGB camera, excluding more capable 3D sensors such as LiDAR or RGB-D cameras. Formally, the problem can be summarized as:

$$\hat{\theta} = \arg\max_{\theta} \pi(a_t \mid \mathbf{o}_t, g_t, g_o, \mathbf{s}_t) \tag{1}$$

where the a_t refers to waypoints in robot navigation settings.
To address the goal of traversability estimation, trajectory generation, and optimal trajectory selection in robot navigation, we propose a two-stage navigation framework built upon the indoor topological mapping method NoMaD. Unlike NoMaD, we eliminate explicit topology graph construction and replace image-based goals (used indoors) with pose-based goals suitable for outdoor environments. As shown in Figure 2, our method comprises three main components: (1) a vision-based temporal traversability estimation module, (2) a diffusion-based trajectory generation module, and (3) a goal-oriented planning module based on sub-goal sampling.

3.2 Traversability estimation with semantic guidance

The traversability estimation module aims to segment the robot's local sensory input into traversable and non-traversable regions—for example, sidewalks and roads are traversable, whereas bushes and obstacles are not.

Under the constraint of using only RGB imagery, a natural approach is to leverage pixel-wise semantic masks, which align well with the semantic definition of traversability.

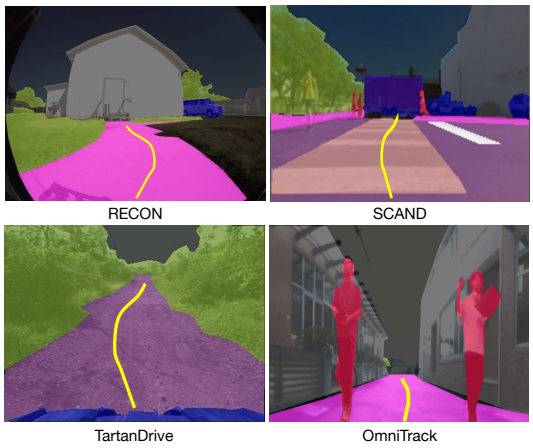

Figure 3: Automated labeling and semantic supervision. We transform raw robot logs into visual navigation clips. Each clip comprises: (1) an egocentric image sequence, (2) a goal-directed ground truth trajectory, and (3) corresponding semantic traversability masks generated by SAM3 to enhance scene generalization.

Specifically, given a sequence of image observations $\mathbf{I}_t := I_{t-4:t}$, we propose a dual-path image encoding scheme tailored respectively for semantic traversability recognition and diffusion-based local trajectory generation (see Section 3.3). We first extract high-dimensional visual features using an image encoder. For the semantic path, we adopt a segmentation head; for the trajectory generation path, we tokenize the features into Transformer-compatible input tokens. The token dimension is 5×256D, where the 5 tokens correspond to four past observations and one current observation.

However, under the egocentric (first-person) camera setting, ensuring sufficient spatial coverage of generated trajectories is crucial for effective planning. To address this, we introduce the concept of sub-goal images: the egocentric RGB view is partitioned into J sub-regions based on perspective-based sub-region attention, each of which undergoes affine transformation and subsequent tokenization. Consequently, the final token representation has dimension $5J \times 256$D.

The total loss for traversability estimation combines a semantic segmentation loss and a penalty for trajectories violating traversability constraints:

$$\mathcal{L}_{\text{trav}} = \lambda_1 \cdot \mathcal{L}_{\text{segment}} + \lambda_2 \cdot \mathcal{L}_{\text{penalize}} \tag{2}$$

Here, $\mathcal{L}_{\text{segment}}$ is the standard cross-entropy loss for semantic segmentation, and $\mathcal{L}_{\text{penalize}}$ measures the proportion of trajectory points—when projected onto the image plane—that fall outside traversable pixels. λ_1 and λ_2 are balancing coefficients.

3.3 Diffusion-based trajectory generation

In the context of map-free outdoor navigation, trajectory generation aims to produce an optimal path that moves the robot toward the goal while satisfying two criteria: (1) safety—the trajectory remains within traversable regions and avoids obstacles; and (2) efficiency—the path minimizes travel distance toward the goal. To this end, we propose a diffusion-based trajectory generation module, wherein safety and efficiency are encoded as conditional inputs to the diffusion process.

Formally, one reverse update of the trajectory diffusion model is expressed as:

$$a_t^{k-1} = \alpha_k \cdot \Big(a_t^k - \gamma_k \, \epsilon_\theta \left(\mathrm{Enc}(C_I, C_M, C_{\delta_g}),\, a_t^k,\, k\right)\Big) + \mathcal{N}(0, \sigma_k^2 I) \tag{3}$$

where t indexes the planning time step for trajectory a_t; k indexes the reverse-diffusion iteration with $k \in \{N, N-1, \ldots, 1\}$, where we use $N = 10$ denoising steps for both training and inference; a_t^k is the noisy trajectory at reverse step k; α_k, γ_k, and σ_k set the noise schedule at step k, with grid search yielding optimal values of $\alpha_k = 0.8$ (state decay), $\gamma_k = 1.0$ (network correction weight), and $\sigma_k = 0.05$ (Gaussian noise standard deviation); $\mathrm{Enc}(\cdot)$ fuses environmental context into a conditioning vector comprising: image observation features (C_I), traversability mask features (C_M), and interpolated pose-difference features (C_{δ_g}); $\mathcal{N}(0, \sigma_k^2 I)$ is Gaussian noise; and ϵ_θ is the noise prediction network.

The noise prediction network ϵ_θ is conditioned on C_I, C_M, and C_{δ_g}, and progressively denoises a Gaussian trajectory distribution. Thus, the dual objectives of traversability compliance and goal-directed efficiency are implicitly encoded into the model and optimized through learning. Note that LiteNav does not incorporate an explicit temporal forecasting module for dynamic agents; instead, safety in dynamic scenes is achieved via high-frequency reactive replanning (10 Hz) operating on a short-horizon local path (~ 2 m), effectively treating moving agents (e.g., pedestrians and vehicles) as instantaneous static obstacles within real-time semantic masks at each time step.

3.4 Goal-oriented planning with sub-goal sampling

Goal-oriented planning operates after trajectory generation and serves to select the best trajectory among the top-K candidates produced by the diffusion model, guided by goal awareness and enhanced with robustness mechanisms.

Specifically, we sample sub-goals from the current observation image. For each sub-goal, candidate trajectories are transformed into the Frenet coordinate frame, and scored based on their lateral and longitudinal distances to the final goal. Additionally, a pose history queue is maintained to detect oscillatory behavior within a sliding window; if oscillation is detected, a replanning trigger is activated. The procedure is formalized in Algorithm 1; if no candidate satisfies the traversability mask, the algorithm returns a REPLAN signal so the outer pipeline can regenerate trajectories.

Algorithm 1. Oscillation-aware goal planner

Require: Current pose g_o, candidates $\mathcal{T}$, goal pose g_t, image I_t, history $\mathcal{H}$ ▷ both in ENU / global frame
Ensure: Track point q^*, or REPLAN if no candidate satisfies $\mathbf{M}$
// Phase 1: Trajectory Pre-processing
1: $\mathbf{M} \leftarrow \textsc{Safety}(\mathcal{T}, I_t)$ ▷ Traversability mask
2: $\mathcal{T}_{\text{enu}} \leftarrow \textsc{Transform}(\mathcal{T}, g_o)$ ▷ To global frame
3: best $\leftarrow (\infty, 0, 0, \emptyset)$ ▷ Init: (score, s, d, index)
// Phase 2: Optimal Trajectory Selection
4: **for** each candidate $\tau_{i,j} \in \mathcal{T}_{\text{enu}}$ **do**
5: **if** $\mathbf{M}[i,j]$ is *True* **then**
6: $(s, d) \leftarrow \textsc{Frenet}(\tau_{i,j}[-1], g_t)$; cost $\leftarrow w_s|s| + w_d|d|$ ▷ Frenet w.r.t. straight-line ref. from g_o to g_t in ENU xy
7: **if** cost $<$ best.score **then**
8: best $\leftarrow$ (cost, s, d, (i,j))
9: **if** best.score $= \infty$ **then**
10: **return** REPLAN ▷ no (i,j) with $\mathbf{M}[i,j] =$ true; signal upper layer
// Phase 3: Oscillation Detection & Recovery
11: $\textsc{UpdateHistory}(\mathcal{H}, \text{best}.s, \text{best}.d)$
12: **if** $\textsc{IsStuck}(\mathcal{H}, W, \theta_{\text{osc}})$ **then** ▷ Check progress window
13: $\Delta\psi \leftarrow \textsc{ComputeRotateAngle}(g_o.\text{yaw}, g_t)$
14: $\textsc{ExecuteRotation}(\Delta\psi)$; $\textsc{ResetHistory}(\mathcal{H})$
15: **return** $\mathcal{T}$[best.idx][track_point_idx] ▷ Control point

In Algorithm 1 (Line 8), the reliance on endpoint deviation ($\tau_{i,j}[-1]$) for Frenet scoring is dictated by our local planning formulation, which operates over a short spatiotemporal horizon (~ 2 m per frame). In this constrained window, prioritizing the terminal heading alignment toward the global goal is critical, and long-horizon trajectory oscillation within a single local frame is negligible; furthermore, we adopt this formulation to maintain strict consistency with the established GND benchmark protocols.

3.5 Vision navigation ground truth generation

To support "out-of-the-box" map-free outdoor navigation, strong scene generalization is essential. We construct a large-scale, knowledge-based pretraining dataset by aggregating multiple outdoor mobile robot datasets [18, 7, 25, 8]. Using sequential frames and associated poses from public datasets, we form visual navigation clips, treating the clip endpoint as the goal and the robot's actual path as the ground truth trajectory, with action-space trajectory distributions normalized and inverse-frequency weighted sampling applied across datasets. We emphasize that LiteNav's training objective uses

Table 1: Quantitative comparison of LiteNav against SOTA navigation methods on the GND dataset. We report the mean and standard deviation ($\pm$) of traversability and distance ratio across 3 independent training runs. LiteNav achieves the highest mean performance with the lowest variance while maintaining the lowest inference latency (tested on an NVIDIA Jetson Xavier NX platform).

Method	**Modality**	**Prior Map**	**Trav. (%)↑**	**Dist. Ratio (%)↑**	**Latency (ms)↓**
ViNT [21]	Camera	Topology	57 ± 1.8	62 ± 2.1	8124
NoMaD [23]	Camera	Topology	59 ± 1.6	61 ± 2.3	2976
MTG [9]	LiDAR	Trav. Map	61 ± 1.5	64 ± 1.9	228
DTG [10]	LiDAR	Trav. Map	67 ± 1.7	66 ± 2.0	1596
VL-TGS [22]	L + C	Trav. Map	65 ± 1.9	70 ± 2.2	2772
LiteNav (Ours)	**Camera**	**None**	$\mathbf{79 \pm 1.2}$	$\mathbf{75 \pm 1.4}$	**87**

publicly released trajectories and poses for imitation-style supervision only; we do not build LiDAR-based traversability maps (e.g., via LIO-SAM) as an intermediate training artifact, which is the key distinction from prior work criticized in Section 2.2. Furthermore, we employ SAM3 [1] to distill pixel-wise semantic masks into 11 discrete categories as supervision for traversability: *Background*, *Curb*, *Fence*, *Road*, *Crosswalk Marking*, *Sidewalk*, *Building*, *Person*, *General Marking*, *Vegetation*, and *Car*. This multi-dataset fusion significantly enhances the model's generalization capability.

3.6 Losses

Overall, the training objective of LiteNav is formulated as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{segment}} + \lambda_2 \mathcal{L}_{\text{penalize}} + \lambda_3 \mathcal{L}_{\text{diffusion}} + \lambda_4 \mathcal{L}_{\text{distance}} \tag{4}$$

where: $\mathcal{L}_{\text{segment}}$ is the cross-entropy loss for semantic mask prediction on observed images; $\mathcal{L}_{\text{penalize}}$ penalizes trajectory points projected onto non-traversable image regions; $\mathcal{L}_{\text{diffusion}}$ is the mean squared error (MSE) between the generated trajectory and the ground truth trajectory; $\mathcal{L}_{\text{distance}}$ is the MSE between the final point of the generated trajectory and the target goal pose. For model training, we utilize an $8 \times$ NVIDIA L40 (40 GB) GPU cluster for approximately 5 days, comprising 4 days of pretraining on the aggregated open datasets followed by 1 day of fine-tuning for real-world deployment.

4 Experiment

4.1 Benchmarking analysis

To evaluate the performance of LiteNav against state-of-the-art (SOTA) robotic navigation methods, we conduct both quantitative and qualitative comparative analyses on publicly available outdoor robot navigation datasets.

4.1.1 Dataset

We conduct experiments on the GND dataset [11], which is proposed for the field of map-free outdoor robot navigation: The GND dataset is a large-scale outdoor navigation benchmark covering 2.7 square kilometers of urban and rural environments. Data were collected using a Clearpath Jackal wheeled mobile robot equipped with a Velodyne VLP-16 16-beam LiDAR, an RGB camera, a 360-degree panoramic camera, and a u-blox GNSS module. LiteNav primarily utilizes the forward-facing RGB camera and the localization module.

4.1.2 Metrics

Since public datasets only support open-loop evaluation—where the planned trajectories generated by the algorithm do not influence the actual robot trajectory (as data are pre-collected)—we adopt the following metrics for quantitative comparison across methods: traversability, distance ratio, and inference time. While open-loop evaluations lack the compounding temporal dynamics of closed-loop execution, traversability and distance ratio serve as high-fidelity proxies for the core requirements of a short-horizon planner: immediate obstacle avoidance and geometric progress. Closed-loop validation is presented in Section 4.2.

• **Traversability**: This metric evaluates the navigability of a trajectory by assessing whether each point along trajectory $\mathbf{t}$ lies within a traversable region, computed as:

$$\mathrm{tr}(\mathbf{A}, \mathbf{t}) = \prod_{m=1}^{M} c(\mathbf{A}, w_m), \tag{5}$$

where $c(\mathbf{A}, w_m)$ indicates whether waypoint w_m is traversable given the traversability map $\mathbf{A}$.

• **Distance Ratio**: This metric measures the similarity between the generated trajectory and the ground-truth trajectory, defined as $1 - \frac{d_t - d_o}{|t|}$, where d_t denotes the distance from the current position to the target along the generated trajectory, and d_o represents the optimal path distance from the current position to the target.

• **Inference Time**: This metric measures the computational latency of the model during inference, reflecting its efficiency and suitability for real-time embedded deployment.

Latency is *not* intended to reflect per-method upper bounds after deployment-specific optimization (e.g., TensorRT); it documents relative cost under a shared, out-of-the-box PyTorch-on-Jetson recipe, with navigation quality as the main comparison axis.

4.1.3 Baseline methods

• **ViNT** [21]: Image-goal navigation on an image topology graph; an EfficientNet–Transformer policy predicts temporal distances and waypoints from graph-selected subgoal images, while an optional image-to-image diffusion model can propose exploratory subgoals for long-horizon graph expansion.

• **NoMaD** [23]: Like ViNT, NoMaD uses an image topology graph and image goals, but replaces the feedforward policy with a diffusion model for multimodal action prediction.

• **MTG** [9]: MTG eliminates the need for a topology graph by encoding LiDAR point clouds and using a conditional variational autoencoder (CVAE) to generate candidate trajectories.

• **DTG** [10]: Building upon MTG, DTG introduces a goal-oriented shortest-trajectory generation mechanism using conditional recurrent neural networks.

• **VL-TGS** [22]: VL-TGS extends MTG by incorporating visual landmarks and leverages a vision-language model (VLM) to select among CVAE-generated candidate trajectories.

Notably, in contrast to the aforementioned SOTA methods, LiteNav is the only truly map-free, vision-only approach. ViNT and NoMaD require pre-built image topology graphs, while MTG, DTG, and VL-TGS rely on LiDAR and necessitate traversability maps constructed during training. As shown in Table 1, we conduct three independent training runs on the fixed train/validation split of the GND benchmark and report the mean and standard deviation for traversability and distance ratio; LiteNav consistently achieves higher mean performance with lower variance while operating at 87 ms latency.

4.1.4 Qualitative analysis

We visually compare the projected trajectories of SOTA methods against those of LiteNav, as shown in Figure 4. For each method, we display its top-K candidate trajectories. The figure shows that LiteNav outperforms SOTA approaches across multiple scenarios: when encountering obstacles, all of its trajectories clearly circumvent them, whereas other methods produce a non-negligible fraction that collide with obstacles, compromising safety. Moreover, at multi-directional intersections with multiple traversable paths, LiteNav accurately covers distinct traversable directions, while other methods either cover only a subset of these regions or exhibit large directional errors.

4.2 Real-world robot experiment

To further validate the effectiveness of LiteNav in map-free outdoor navigation, we conduct closed-loop navigation experiments on a real-world robotic platform. We employ a quadruped robot, the Unitree Go2 [26], equipped with an onboard NVIDIA Jetson NX compute unit and outfitted solely with a monocular RGB camera and a GPS module—no additional sensors such as LiDAR or RGB-D cameras are used, as illustrated in Figure 5. The camera operates at 10 Hz, and LiteNav achieves matching real-time output at 10 Hz. For environmental observation sequences, we use the most recent four frames as historical context.

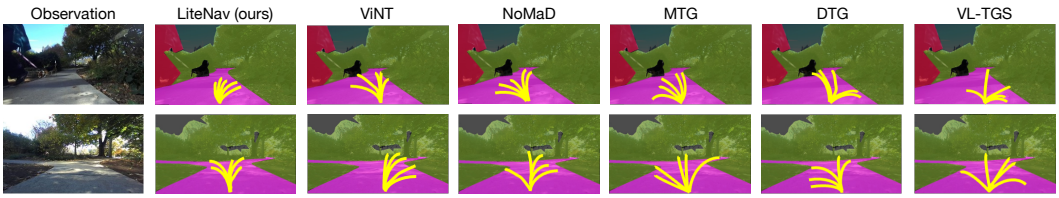

Figure 4: Qualitative comparison against SOTA methods on the GND dataset.

Inference Implementation Details: During training, the next goal image serves as the image goal; during inference, however, the next goal image is unavailable. To address this, we propose partitioning the current frontal-view image into five sub-goal images according to potential navigation directions. These correspond to projections of five candidate trajectories onto the frontal image plane, centered at azimuth angles of $-60°$, $-30°$, $0°$, $30°$, and $60°$ within the robot's $180°$ forward field of view. After affine transformation, these sub-goal images are batch-fed into LiteNav's image encoder, and the diffusion model subsequently generates $K \times 5$ candidate trajectories. This design ensures effective local planning guided by directional goals while enhancing trajectory coverage.

Among the SOTA methods, DTG, MTG, and VL-TGS rely on LiDAR and are thus incompatible with our sensor-limited platform. Therefore, we compare LiteNav only against the vision-based methods ViNT and NoMaD. To enable their operation—which requires an image-based topology graph—we pre-collect data in the test environment to construct the necessary graph for both methods. The collection area and test routes are shown in Figure 5. Specifically, in our real-world closed-loop setups, global targets span 200 m to 2 km, and environments encompass enclosed campuses and urban municipal roads, including paved lanes, sidewalks, crosswalks, curbs, and unpaved vegetated areas. The routes include standard pedestrian interaction zones as well as high-density transit areas, evaluating the system against dynamic cross-traffic and close-proximity obstacle avoidance; data collection was conducted under daytime clear weather, with a subset of light-rain scenarios. We strictly controlled hardware, route selection, and time-of-day variables. To account for the stochasticity of dynamic real-world environments, each method executed 10 independent trials per route, and we report the expected performance metrics in Table 2 (quantifying exact statistical variance in dynamic real-world environments remains challenging due to inherently non-reproducible pedestrian and traffic states).

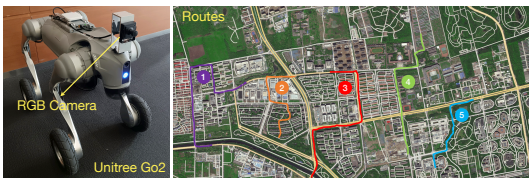

Figure 5: Illustration of the robot platform (Unitree Go2) and open-world test routes.

Table 2: Real-world performance comparison.

Method	Success Rate ↑	Collisions ↓
ViNT [21]	77%	11
NoMaD [23]	83%	9
LiteNav (Ours)	**92%**	**2**

Success rate indicates whether the robot reaches the designated goal, while collisions measure unintended contacts with obstacles. We evaluate closed-loop navigation performance using success rate and collision count. Despite ViNT/NoMaD having access to a prior map, LiteNav still outperforms them in map-free settings.

Across the three compared methods (ViNT, NoMaD, and LiteNav), the relative rankings on open-loop GND metrics (Table 1) align with the real-world closed-loop success rate rankings (Spearman's rank correlation $\rho = 1.00$), serving as a preliminary consistency check between our open-loop and closed-loop evaluations.

Safety–Efficiency Trade-off: Figure 6 illustrates the behavioral divergence between methods under identical visual inputs (e.g., curb-following and crosswalks). While the baselines (ViNT and NoMaD) optimize purely for distance-to-goal (geometric efficiency), their generated paths often exhibit unacceptably small safety margins relative to obstacles. In contrast, LiteNav is explicitly optimized to trade minor geometric detours for expanded safety margins, representing a deliberate safety–efficiency trade-off.

Robustness to Degraded GPS Conditions: Architecturally, GNSS acts strictly as a global directional prior, while the local traversability manifold is constrained by visual perception. We evaluate GPS error sensitivity along a 600-meter park route with four multi-directional intersections (5 independent runs per grade), superimposing zero-mean Gaussian noise ($\sigma = 10\,\text{m}$ and $20\,\text{m}$) onto the baseline ($\sigma \leq 5\,\text{m}$). As shown in Table 3, under $\sigma = 10\,\text{m}$ noise, isolated directional misjudgments are corrected via reactive replanning, with a single failure caused by cumulative detours across intersections. Under severe drift ($\sigma = 20\,\text{m}$), large localization deviations induce consecutive incorrect route selections across the four intersections, causing prolonged looping in three trials.

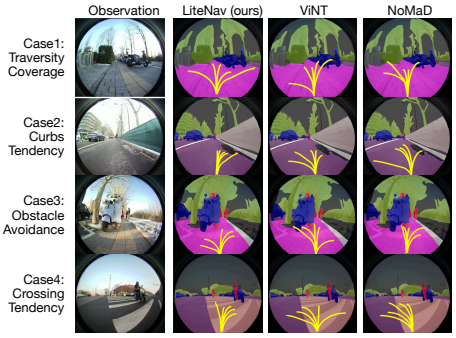

Figure 6: Real-world comparison of LiteNav with vision-based navigation methods ViNT and NoMaD under identical visual inputs, illustrating the safety–efficiency trade-off: LiteNav trades minor geometric detours for expanded safety margins around obstacles and curbs. (ViNT and NoMaD do not output masks; we project their trajectories onto our mask images for visualization.)

Table 3: Quantitative evaluation of LiteNav under degraded GPS positioning conditions on a 600 m real-world park route with four multi-directional intersections (5 independent trials per grade).

GPS Position Error Grade	Successful Trials	Success Rate ↑	Total Collisions ↓	Avg. Completion Time ↓
Baseline ($\sigma < 5$ m native error)	5 / 5	**100%**	**0**	**612 s**
Medium Drift (Extra $\sigma = 10$ m Gaussian noise)	4 / 5	80%	**0**	688 s
Severe Drift (Extra $\sigma = 20$ m Gaussian noise)	2 / 5	40%	1	840 s

4.3 Ablation study

In this section, we conduct an ablation study on key components of LiteNav: semantic-guided traversability estimation, sub-goal image modeling, diffusion-based trajectory generation, and goal-oriented planning: (a) Remove semantic supervision for traversability estimation; (b) Remove sub-goal modeling and use only the original full-frame image; (c) Replace the diffusion model with a variational autoencoder (VAE); (d) Remove the goal-oriented planner and directly select the trajectory with the highest probability; (e) Full LiteNav model with all components enabled.

Table 4: Ablation on modular contributions.

	Semantic Traversability	Sub-goal Model	Diffusion Model	Goal Planner	Success Rate	Collisions
(a)	–	✓	✓	✓	79%	12
(b)	✓	–	✓	✓	82%	2
(c)	✓	✓	–	✓	89%	3
(d)	✓	✓	✓	–	75%	3
(e)	✓	✓	✓	✓	**92%**	**2**

The ablation study reveals that: (a) Semantic supervision is crucial for traversability estimation; removing it significantly increases collision rates. (b) Sub-goal modeling is essential, as its removal drastically reduces navigation success. (c) Compared to VAEs, diffusion models can better capture the multi-modal distribution of trajectories, thereby providing a richer set of candidate paths at complex intersections. (d) The goal-oriented planner plays a critical role—its removal leads to a sharp drop in success rate, highlighting the importance of goal-directed trajectory selection and the oscillation recovery mechanism.

5 Conclusion

We proposed **LiteNav**, a lightweight framework for truly map-free outdoor navigation. Using only a monocular camera, LiteNav operates without pre-built maps or LiDAR-derived priors. By integrating semantic traversability with a goal-conditioned diffusion model and high-frequency reactive replanning, it generates collision-free, goal-directed local trajectories from RGB observations. Our dual-path encoding and sub-goal sampling provide broad trajectory coverage and goal-directedness across diverse outdoor terrains. Evaluations on the open-loop GND benchmark and real-world closed-loop tests demonstrate that LiteNav outperforms existing baselines in traversability and navigation success. Future work includes enhancing LiteNav with 3D semantic occupancy predictions for more explicit spatial understanding and finer-grained collision avoidance in complex scenarios.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their helpful comments and our colleagues for discussions.

References

[1] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

[2] Samarth Chopra, Jing Liang, Gershom Seneviratne, Yonghan Lee, Jaehoon Choi, Jianyu An, Stephen Cheng, and Dinesh Manocha. SplatBlox: Traversability-aware Gaussian splatting for outdoor robot navigation. *arXiv preprint arXiv:2511.18525*, 2025.

[3] Sivasankar Ganesan, Senthil Kumar Natarajan, and Jeevaanand Srinivasan. A global path planning algorithm for mobile robot in cluttered environments with an improved initial cost solution and convergence rate. *Arabian Journal for Science and Engineering*, 47(3):3633–3647, 2022.

[4] Penggang Gao, Zihan Liu, Zongkai Wu, and Donglin Wang. A global path planning algorithm for robots using reinforcement learning. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1693–1698. IEEE, 2019.

[5] Sourav Garg, Dustin Craggs, Vineeth Bhat, Lachlan Mares, Stefan Podgorski, Madhava Krishna, Feras Dayoub, and Ian Reid. ObjectReact: Learning object-relative control for visual navigation. In *Conference on Robot Learning (CoRL)*, 2025.

[6] Samiran Gode, Abhijeet Nayak, Débora NP Oliveira, Michael Krawez, Cordelia Schmid, and Wolfram Burgard. FlowNav: Combining flow matching and depth priors for efficient navigation. *arXiv preprint arXiv:2411.09524*, 2024.

[7] Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (SCAND): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7(4):11807–11814, 2022.

[8] Young-Ho Kim, Jae Ho Jeon, Bongshin Lee, Eun Kyoung Choe, and Jinwook Seo. OmniTrack: A flexible self-tracking approach leveraging semi-automated tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–28, 2017.

[9] Jing Liang, Peng Gao, Xuesu Xiao, Adarsh Jagan Sathyamoorthy, Mohamed Elnoor, Ming C Lin, and Dinesh Manocha. MTG: Mapless trajectory generator with traversability coverage for outdoor navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2396–2402. IEEE, 2024.

[10] Jing Liang, Amirreza Payandeh, Daeun Song, Xuesu Xiao, and Dinesh Manocha. DTG: Diffusion-based trajectory generation for mapless global navigation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5340–5347. IEEE, 2024.

[11] Jing Liang, Dibyendu Das, Daeun Song, Md Nahid Hasan Shuvo, Mohammad Durrani, Karthik Taranath, Ivan Penskiy, Dinesh Manocha, and Xuesu Xiao. GND: Global navigation dataset with multi-modal perception and multi-category traversability in outdoor campus environments. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2383–2390. IEEE, 2025.

[12] Jing Liang, Kasun Weerakoon, Daeun Song, Senthurbavan Kirubaharan, Xuesu Xiao, and Dinesh Manocha. MOSU: Autonomous long-range robot navigation with multi-modal scene understanding. *arXiv preprint arXiv:2507.04686*, 2025.

[13] Ryan Meegan, Adam D'Souza, Bryan Bo Cao, Shubham Jain, and Kristin Dana. YOPO-Nav: Visual navigation using 3DGS graphs from one-pass videos. *arXiv preprint arXiv:2512.09903*, 2025.

[14] Ülkü Öztürk, Melih Akdağ, and Tarık Ayabakan. A review of path planning algorithms in maritime autonomous surface ships: Navigation safety perspective. *Ocean Engineering*, 251:111010, 2022.

[15] Martin Psotka, František Duchoň, Mykhailyshyn Roman, Tölgyessy Michal, and Dobiš Michal. Global path planning method based on a modification of the wavefront algorithm for ground mobile robots. *Robotics*, 12(1):25, 2023.

[16] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*, 2018.

[17] Dhruv Shah and Sergey Levine. ViKiNG: Vision-based kilometer-scale navigation with geographic hints. In *Robotics: Science and Systems (RSS)*, 2022.

[18] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. In *Conference on Robot Learning (CoRL)*, pages 674–684. PMLR, 2021.

[19] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. ViNG: Learning open-world navigation with visual goals. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13215–13222. IEEE, 2021.

[20] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[21] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A foundation model for visual navigation. In *Conference on Robot Learning (CoRL)*, pages 711–733. PMLR, 2023.

[22] Daeun Song, Jing Liang, Xuesu Xiao, and Dinesh Manocha. VL-TGS: Trajectory generation and selection using vision language models in mapless outdoor environments. *IEEE Robotics and Automation Letters*, 2025.

[23] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. NoMaD: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024.

[24] Lauri Suomela, Jussi Kalliola, Harry Edelman, and Joni-Kristian Kämäräinen. PlaceNav: Topological navigation through place recognition. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5205–5213. IEEE, 2024.

[25] Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. TartanDrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2546–2552. IEEE, 2022.

[26] Unitree. Unitree Go2-W product page. https://www.unitree.com/go2-w, 2026.

[27] Kasun Weerakoon, Mohamed Elnoor, Gershom Seneviratne, Vignesh Rajagopal, Senthil Hariharan Arul, Jing Liang, Mohamed Khalid M Jaffar, and Dinesh Manocha. BehAV: Behavioral rule guided autonomy using VLMs for robot navigation in outdoor scenes. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7044–7051. IEEE, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 accurately frame LiteNav's map-free monocular setting (requiring no prior environmental maps or LiDAR LIO training maps), its two-stage reactive local planning design (10 Hz short-horizon replanning without explicit dynamic trajectory forecasting), Jetson NX deployment, and empirical scope (open-loop evaluation on GND against LiDAR/vision baselines and closed-loop real-world evaluation against vision-only baselines).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4.2 quantitatively evaluates sensitivity to GPS drift ($\sigma = 10\,\text{m}$ and 20 m) and characterizes failure cases under severe localization degradation at multi-directional intersections; Section 5 outlines future directions (e.g., richer 3D semantic occupancy) that bound the current system's spatial reasoning.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 3.1 states the map-free observation and goal formalism; Sections 3.2–3.6 specify traversability losses, the conditioned diffusion update, planner logic, supervision construction, and the composite objective with explicit terms and coefficients—giving a complete mathematical specification of the method rather than separate numbered theorems.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 describes the full LiteNav pipeline (traversability path, sub-goal tokenization, goal-conditioned diffusion, Frenet planner, SAM-supervised pretraining, and losses), while Section 4 documents the GND benchmark, metric definitions, baselines, open-loop protocol, Jetson latency measurement, real-world setup, and ablations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Evaluations rely on the public GND benchmark and other cited public robot datasets for pretraining.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3 specifies the map-free formulation, traversability path, sub-goal tokenization, diffusion conditioning ($N = 10$ denoising steps and grid-searched schedule parameters $\alpha_k = 0.8$, $\gamma_k = 1.0$, $\sigma_k = 0.05$), planner, and losses; Section 4.1 covers the GND benchmark (sensors used, metrics, baselines, open-loop setting) and Section 4.2 documents the Unitree Go2 / Jetson deployment (monocular RGB, GPS module, four-frame history at 10 Hz camera rate, five azimuth sub-goals, $K \times 5$ candidate trajectories, 200 m–2 km routes across diverse terrains and weather, and 10 independent trials per route)—sufficient context to interpret Tables 1, 2, 3, and 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 reports the mean and standard deviation ($\pm 1\sigma$) of traversability and distance ratio on the GND benchmark across 3 independent training runs; real-world closed-loop results in Table 2 aggregate 10 independent trials per route across methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 3.6 specifies training compute ($8\times$ NVIDIA L40 40 GB GPUs for ~ 5 days: 4 days of open-dataset pretraining and 1 day of real-world fine-tuning), while Section 4 and Table 1 document embedded inference efficiency and latency on the NVIDIA Jetson Xavier NX platform.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work uses published outdoor navigation datasets and laboratory field tests on a robot platform without human-subject protocols or high-risk model release described in the manuscript.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Sections 1–2 emphasize practical benefits of lightweight, map-free outdoor navigation (lower sensor and compute cost for deployable robots), but the paper does not explicitly analyze potential negative societal impacts such as misuse of autonomous platforms in public spaces.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped web-scale datasets, or other assets that would require special misuse safeguards beyond standard academic use of public robot corpora.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: GND, upstream pretraining datasets, SAM3, baselines, and robot platforms are credited through bibliographic citations to their originating works (Sections 2–4 and Section 3.5), and usage follows publicly released academic resources.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper introduces the LiteNav method and evaluates it on public benchmarks, but it does not release a new public dataset, model checkpoint, or code asset as part of the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this paper does not use LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.