2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

NAVMOE: Hybrid Model- and Learning-based Traversability Estimation for Local Navigation via Mixture of Experts

Botao He †[1], Amir Hossein Shahidzadeh †[1], Yu Chen†[2], Jiayi Wu[1], Tianrui Guan[1], Guofei Chen[2], Howie Choset[2], Dinesh Manocha[1], Glen Chou[3], Cornelia Fermuller[1], and Yiannis Aloimonos[1]

***Abstract*— This paper explores traversability estimation for robot navigation. A key bottleneck in traversability estimation lies in efficiently achieving reliable and robust predictions while accurately encoding both geometric and semantic information across diverse environments. We introduce Navigation via Mixture of Experts (NAVMOE), a hierarchical and modular approach for traversability estimation and local navigation. NAVMOE combines multiple specialized models for specific terrain types, each of which can be either a classical model-based or a learning-based approach that predicts traversability for specific terrain types. NAVMOE dynamically weights the contributions of different models based on the input environment through a gating network. Overall, our approach offers three advantages: First, NAVMOE enables traversability estimation to adaptively leverage specialized approaches for different terrains, which enhances generalization across diverse and unseen environments. Second, our approach significantly improves efficiency with negligible cost of solution quality by introducing a training-free lazy gating mechanism, which is designed to minimize the number of activated experts during inference. Third, our approach uses a two-stage training strategy that enables the training for the gating networks within the hybrid MoE method that contains nondifferentiable modules. Extensive experiments show that NAVMOE delivers a better efficiency and performance balance than any individual expert or full ensemble across domains, improving cross-domain generalization and reducing average computational cost by 81.2% via lazy gating, with less than a 2% loss in path quality.**

I. INTRODUCTION

We address traversability estimation for safe and efficient robot navigation—a core technique that evaluates how easily a robot can move through an environment, providing key heuristics for path planning [1]. Classical approaches typically rely on rule-based models based on geometric analysis [2], [3], [4]. These approaches extract geometric features such as slope, height variation, and obstacles from sensor inputs and estimate traversability using manually designed heuristics, including terrain roughness thresholds, elevation gradients, and obstacle proximity rules. Such model-based approaches are highly interpretable and allow for the straightforward incorporation of hard constraints on robot motion such as collision avoidance through physical and geometric rules. As a result, they offer provable safety guarantees and strong generalization across different environments. Additionally, model-based methods are typically lightweight and computationally efficient [5]. This makes them well-suited for resource-constrained mobile platforms. However, model-based approaches often struggle in complex or ambiguous environments, where geometric and physical features may be noisy or incomplete [6], [7]. They also struggle in understanding high-level features such as semantic information, which limits their capacity to anticipate hidden risks (e.g., recognize and follow traffic signs) or results in over-conservative behaviors.

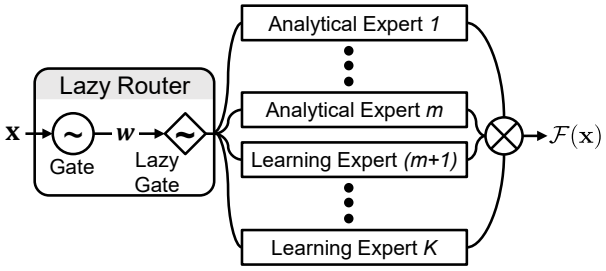

Fig. 1: A basic block of the NAVMOE. It fuses the output of different experts through a learned router.

With the rapid development of machine learning, learning-based approaches have become increasingly popular for traversability estimation [8], [9], [10], [11]. Compared to model-based approaches, learning-based methods offer significant advantages in unstructured environments. They also excel at interpreting high-level semantic information in the environment, such as recognizing terrain categories, obstacles, and contextual features that are critical for navigation decisions. Despite these advantages, learning-based approaches face challenges in providing formal safety guarantees. This is because they often lack interpretability, which makes it difficult to incorporate explicit safety constraints during deployment. Also, cross-domain generalization for a single model remains challenging.

Combining experts with different expertise is a effective approach in ML. Classical ensemble learning methods like [12] or Mixture of Experts (MoE) [13], [14] often leads to performance improvements. Recent works aim to address the above challenges by integrating model-based and learning-based methods into a unified framework. [15] encodes images and point clouds into a shared 3D map for geometric analysis and traversability prediction but depends heavily on high-quality multi-sensory data for training. [16] serially connect both paradigms using a human-designed probing strategy, but lacks cross-domain flexibility and is inefficient as both methods must always run. [17] utilizes physics-informed learning to improve generalization through sim-

1 Department of Computer Science, University of Maryland, MD 20742.
2 Robotics Institute, Carnegie Mellon University, PA 15213-3890.
3 Georgia Institute of Technology, GA 30332.
† Equal contribution. Email: {botao, jyaloimo}@umd.edu

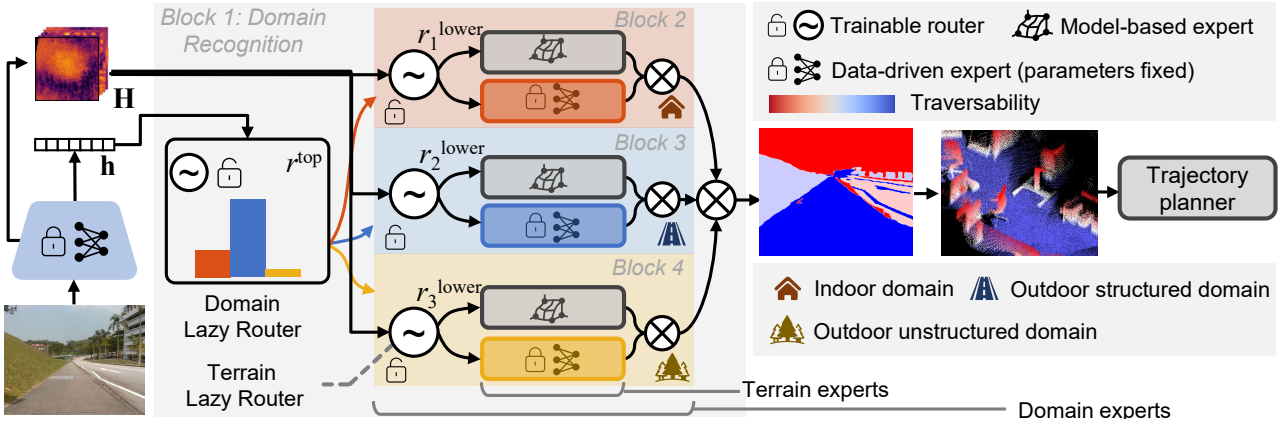

Fig. 2: **Overview of the hierarchical and modular design.** Given sensor input, the domain router first routes it to the most suitable terrain router. This terrain router then computes a pixel-wise weight map indicating each expert's confidence and queries the terrain experts iteratively. Each expert generates its traversability map, and these outputs are fused according to their weights to form the final estimation. All routers employ the proposed Lazy Gating mechanism. All blocks follow the same modular design illustrated in Fig. 1.

ulation, but the domain gap remains a challenge. However, these hybrid approaches still face significant challenges for traversability estimation: 1) It is difficult to determine the contribution of each method, as their performance can vary widely across domains. 2) Hybrid networks often require extensive expensive synchronized multi-sensory data for training. 3) It is often difficult to jointly update the components to enable better cooperation among them. This is because learning-based methods rely on gradient-based optimization, while model-based methods are usually nondifferentiable, as they often involve discrete decision rules or search-based operations that break gradient flow.

Main Results: We propose Navigation via Mixture of Experts (NavMoE), a hybrid analytical–learning approach designed for traversability estimation. NavMoE unifies both model-based and learning-based approaches by incorporating them as modular components within a hierarchical Mixture of Experts (MoE) architecture. A basic building block is shown in Fig. 1 and the overall architecture is shown in Fig. 2. The novelties of our work include: **1)** A hierarchical MoE approach that performs expert selection at two levels: A high-level router coarsely identifies relevant domains, and a low-level router determines pixel-wise contribution of each expert to the final traversability prediction. **2)** An expert pruning approach based on the lazy gating mechanism to improve the efficiency of NavMoE. The core idea is to only activate an expert when it can lead to significant changes to the robot path solution. Instead of calling the best expert, lazy gating calls fast yet effective experts first and adds others only when needed, keeping the accuracy while improving efficiency. We theoretically prove that the path solution obtained using only a subset of experts differs from the solution using all experts by *at most a bounded margin* in terms of solution quality. **3)** A two-stage training approach that improves the performance of NavMoE with limited high quality multi-sensory data. The first stage involves pre-training on large-scale, weakly-labeled data to learn generalizable representations. It is then followed by a fine-tuning phase using smaller, accurately-labeled datasets to refine task-specific performance.

Extensive experiments in real world and diverse datasets across different domains demonstrate that NavMoE achieves 35.3% higher average accuracy in traversability map estimation compared to **any individual expert**, both purely geometric and learning-based methods. Moreover, NavMoE reduces computational overhead in average by 86.2% compared to our method without lazy gating, with path quality drop within 0.02. We will open source our code to facilitate future research. We believe the proposed hybrid modeling principle is generic and we expect its application in other robotics tasks.

II. Hierarchical and Modular Mixture-of-Experts

An overview of NavMoE is shown in Fig. 2. Given a RGB image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ and a depth image $\mathbf{D} \in \mathbb{R}^{H\times W}$, NavMoE predicts a traversability map $\mathcal{T} \in \mathbb{R}^{H\times W}$ and then computes a feasible robot trajectory $\tau \in \mathbb{R}^{3\times l}$, where l is the trajectory length. NavMoE introduces two levels of specialization: *domain experts* $\mathbf{E}_m^{\text{Domain}}$ with $m = \{1, 2, ..., M\}$ at the top level and *terrain experts* $\mathbf{E}_{mn}^{\text{Terrain}}$ with $n = \{1, 2, ..., N\}$ within each domain m. Notably, N is the number of terrain experts within each domain; in our setup, $M = 3$ and $N = 2$, as illustrated in Fig. 2. Domain experts route input images to relevant high-level domains for targeted optimization, while terrain experts further refine predictions based on distinct terrain features.

a) Top Level Router: Domain Specialization: At the top level, NavMoE performs coarse specialization by partitioning the input domain based on scenario types. Within the scope of this paper, we define three domains: 1) indoor, including hallways, rooms, and warehouses; 2) structured outdoor, featuring roads with lanes, sidewalks, and crosswalks; and 3) unstructured outdoor, encompassing off-road settings like forests, rocky terrain, and grassy fields.

We formulate domain partitioning as an image classification problem. NavMoE first extracts a dense feature map $\mathbf{H} \in \mathbb{R}^{H\times W\times d_1}$ and a feature vector $\mathbf{h} \in \mathbb{R}^{d_2}$ from the input images using a pre-trained encoder $\mathcal{E}$:

$$\mathbf{H}, \mathbf{h} = \mathcal{E}(\mathbf{I}) \tag{1}$$

where we use DINOv2 [18] as the encoder $\mathcal{E}$. After that, the top-level router $r^{\text{top}} : \mathbb{R}^{d_2} \to \mathbb{R}^M$ takes $\mathbf{h}$ as input for domain classification. The output of the top-level router is a normalized vector $\mathbf{w}_m \in \mathbb{R}^{M\times 1}$, which indicates the predicted possibilities for each type of domain: $\mathbf{w}_m = r^{\text{top}}(\mathbf{h})$. Then, based on the predicted weights $\mathbf{w}_m$, the domain experts $\mathbf{E}^{\text{Domain}}$ are queried using the proposed lazy gating algorithm (detailed in Section III).

b) Lower Level Router: Terrain Specialization: Within the selected domain, we implement a lower-level expert routing mechanism. Specifically, a router r^{lower} assigns terrain experts $\mathbf{E}^{\text{Terrain}}_{mn}$ to each pixel based on a pixel-wise terrain partition of the input image. Unlike the top-level coarse classification that assigns the whole input to several domain categories, $r^{\text{lower}} : \mathbb{R}^{H\times W\times d_1} \to \mathbb{R}^{H\times W\times N}$ performs fine-grained segmentation by predicting pixel-wise terrain expert preferences. It takes feature map $\mathbf{H}$ as input and predicts a weight map of expert weights $\mathbf{W} \in \mathbb{R}^{H\times W\times N}$:

$$\mathbf{W} = r^{\text{lower}}(\mathbf{H}) \tag{2}$$

Therefore, weight map for each expert can be expressed as $\mathbf{W}_n \in \mathbb{R}^{H\times W}$. Each terrain expert $\mathbf{E}^{\text{Terrain}}_{mn}$ is implemented as either a neural network or a model-based algorithm (details in Section V). These experts take RGB or depth images as input and produce pixel-aligned traversability maps $\mathcal{T}_{mn} \in \mathbb{R}^{H\times W}$. Formally, an expert can be expressed as a mapping $\mathbf{E}^{\text{Terrain}}_{mn} : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H\times W}$, where $C = 3$ if the input is an RGB image and $C = 1$ if the input is a depth image. The final $\mathcal{T}$ is computed as a two-level weighted combination of expert predictions:

$$\mathcal{T} = \sum_{m=1}^{M} \left(\mathbf{w}_m \cdot \sum_{n=1}^{N} \left(\mathbf{W}_n^{\text{lower}} \odot \mathcal{T}_{mn} \right) \right), \tag{3}$$

where $\odot$ refers to elementwise multiplication. The resulting $\mathcal{T}$ is then passed to a motion planner to compute an optimal trajectory τ^*. Here we adopt the primitive-based planner in [5]. Defining the candidate motion primitives set for the traversability map $\mathcal{T}$ as $S(\mathcal{T}) = \{\tau_1, \tau_2, ..., \tau_l\}$, we formulate the trajectory optimization problem as:

$$f(\mathcal{T}, p_{\text{goal}}) \triangleq \underset{\tau\in S(\mathcal{T})}{\arg\min} \sum_{i=1}^{\ell} \Big(J_{\text{trav}}(\tau_i, \mathcal{T}) + \lambda\, J_{\text{dis}}(\tau_i, p_{\text{goal}}) \Big). \tag{4}$$

$J_{\text{trav}}(\tau_i, \mathcal{T})$ represents the traversability cost of the path τ_i on $\mathcal{T}$, $J_{\text{dis}}(\tau_i, p_{\text{goal}})$ represents the cost of τ_i with respect to reaching the goal p_{goal}. Finally, $\tau^* = f(\mathcal{T}, p_{\text{goal}})$.

Specifically, let $J_{\text{trav}}(\tau, \mathcal{T})$ denote the traversability cost of a 3D trajectory τ with respect to a 2D traversability map $\mathcal{T}$. Utilizing the depth information, we first project $\tau = \{q_1, q_2, \ldots, q_l\}$, where $q_i \in \mathbb{R}^3$, onto the 2D plane of $\mathcal{T}$, yielding $\hat{\tau} = \{p_1, p_2, \ldots, p_l\}$, where each $p_i = (x_i, y_i) \in \mathbb{R}^2$ corresponds to a waypoint on the map. The traversability cost is then computed by summing the map values at each projected waypoint: $J_{\text{trav}}(\tau, \mathcal{T}) = \sum_{p_i\in\hat{\tau}} (1 - \mathcal{T}(p_i)) \Big/ |\tau|$. Notably, since $J_{\text{trav}}(\tau, \mathcal{T})$ is defined as a cost, lower values indicate better traversability.

For $J_{\text{dis}}(\tau, p_{\text{goal}})$, let $\tau = \{q_1, q_2, \ldots, q_l\}$ denote a candidate 3D trajectory and $p_{\text{goal}} \in \mathbb{R}^3$ be the goal position. Let $p \in \mathbb{R}^3$ be the current robot position. Define the distance-based cost as: $J_{\text{dis}}(\tau, p_{\text{goal}}) = 1 - (\|p - p_{\text{goal}}\| - \min_{q_j\in\tau} \|q_j - p_{\text{goal}}\|) \big/ H$, where H denotes the maximum planning distance, which means that for all q_j in τ, $\|q_j - p\| \le H$. H is set as $2m$ for indoor, $4m$ for unstructured outdoor and $8m$ for unstructured outdoor environment. The result lies in the range of $[0, 1]$.

III. Lazy Gating Mechanism for Expert Pruning

A. Lazy Gating Mechanism

As the number of experts increases, activating all experts can cause severe computational overhead. To address this challenge, we introduce the lazy gating mechanism as detailed in Algorithm 1. The key idea is to selectively activate a subset of experts that could significantly impact the path solution. We implement the lazy gating mechanism to both lower and higher level routers.

Algorithm 1 Lazy Gating

function LazyGating
  Compute $\mathbf{W} \in \mathbb{R}^{H\times W\times N}$ using Eq. (2)
  Construct $\mathcal{Q}_E$ using Eq. (5)
  $\mathcal{T} \leftarrow$ None, $\mathbf{W}^{\text{prior}} \leftarrow 0$
  for $k = 1, 2, \cdots, \mathcal{K}$ **do**
    $E \leftarrow$ Dequeue($\mathcal{Q}_E$)
    $\hat{\mathcal{T}} \leftarrow E(\mathbf{I}, \mathbf{D})$
    if $k == 1$ **then**
      $\mathcal{T} \leftarrow \hat{\mathcal{T}}$, $\mathbf{W}^{\text{prior}} \leftarrow \mathbf{W}[n]$
    else
      $\mathcal{T} \leftarrow \frac{\mathcal{T}\cdot\mathbf{W}^{\text{prior}} + \hat{\mathcal{T}}\cdot\mathbf{W}[k]}{\mathbf{W}^{\text{prior}} + \mathbf{W}[k]}$ ▷ Eq. (6)
      $\mathbf{W}^{\text{prior}} \leftarrow \mathbf{W}^{\text{prior}} + \mathbf{W}[k]$
    Done $\leftarrow$ CheckTerm
    if *Done* **then**
      break
  return $\mathcal{T}$
end function
function CheckTerm
  Compute $\underline{\mathcal{T}}$ and $\overline{\mathcal{T}}$ using Eq. (7a) and Eq. (7b), respectively
  $\overline{C} \leftarrow$ Planner($\underline{\mathcal{T}}$), $\underline{C} \leftarrow$ Planner($\overline{\mathcal{T}}$)
  if $\overline{C} - \underline{C} \le \epsilon$ **then**
    return True
  else
    return False
end function

For the case of the lower router, we define the cost function $\Phi(\mathbf{E}^{\text{Terrain}}_{mi}) = (1 - ||\mathbf{W}_i||_1 / \sum_{n=1}^{N} ||\mathbf{W}_n||_1) \cdot \phi(\mathbf{E}^{\text{Terrain}}_{mi})$, where $||\cdot||_1$ is the pixel-wise sum of the weight map and $\phi(\cdot)$ indicates the computational cost of an expert, here measured with FLOPS (Floating Point Operations). We consider the lower router case here; however, the same logic can be applied to the top-level router as well. We begin by constructing an expert queue based on $\Phi(\mathbf{E}^{\text{Terrain}}_{mi})$:

$$\begin{aligned} \mathcal{Q}_E &= \{E_1, \ldots, E_k, \ldots, E_{\mathcal{K}}\} \\ \text{s.t.} \quad &\Phi(E_1) \le \Phi(E_2) \le \cdots \le \Phi(E_{\mathcal{K}}). \end{aligned} \tag{5}$$

Here, $\mathcal{K} = K$ defined in Fig. 1; however, we adopt a different notation to represent the reordering process used to construct

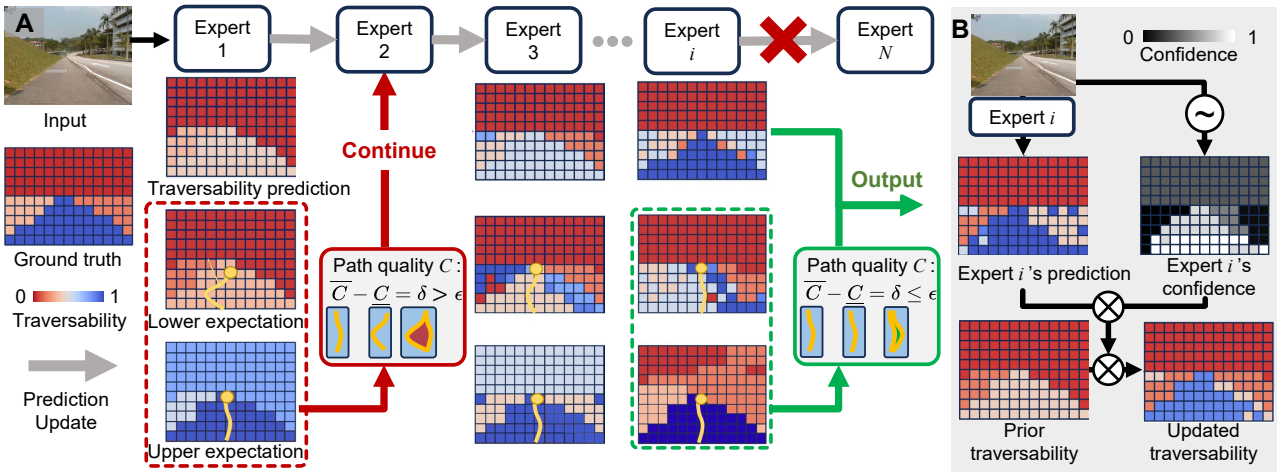

Fig. 3: **Lazy Gating Mechanism. (A)** Illustration of expert pruning during inference. **(B)** One iteration of the traversability map update.

the queue. Intuitively, if experts have similar weights, we prioritize those with higher computational efficiency; if one has a dominant weight, we prioritize even if it has lower efficiency. As shown in Fig. 3 A, the lazy gating algorithm processes experts sequentially, iterating from the front of the queue to the rear. During each iteration, the next expert is dequeued and queried to provide a traversability prediction. The internal processing of each expert is shown in Fig. 3 B: At iteration k, the algorithm maintains a prior traversability map $\mathcal{T}_{k-1}$, which aggregates the predictions from the previous $k-1$ experts. The kth expert predicts a traversability map $\hat{\mathcal{T}}_k$ while the gating network estimates the current expert's confidence as weight $\mathbf{W}_k$. The current expert's output $\hat{\mathcal{T}}_k$ is then fused into the overall map:

$$\mathcal{T}_k = \left(\mathcal{T}_{k-1} \odot \textstyle\sum_{j=1}^{k-1} \mathbf{W}_j + \mathcal{T}_k \odot \mathbf{W}_k\right) \Big/ \textstyle\sum_{j=1}^{k} \mathbf{W}_j, \quad (6)$$

where the division is performed elementwise. The updated traversability map $\mathcal{T}_k$ is then used as the new prior for the next iteration. The planner will take $\mathcal{T}_k$ as input to compute the path τ_k and path cost $C(\tau_k)$ using Eq. (4). Given $C(\tau_k)$ as prior, the expected cost of the final path $C(\tau_K)$ will be constrained to lie within an interval after any admissible update from expert $k+1$ to expert K; we discuss how to get the bounds of this interval in the following proposition. To reduce the computation overhead, we prune the rest of the experts if they do not significantly change the traversability output by the threshold ϵ, as detailed in lines 18-25 of Alg. 1.

Proposition 1 (Convergence Bound on Path Cost Under Incremental Traversability Updates)**.** *Let $\mathcal{T}_k$ be the traversability map after k aggregative updates following Eq.* (6)*, and let $C(\mathcal{T}_k) \doteq C(\tau^*)$ denote the path cost computed using $\mathcal{T}_k$. Define upper and lower expectation maps based on the remaining uncertainty in traversability ($[0,1]$) from step $k+1$ to K as:*

$$\begin{aligned}\underline{\mathcal{T}_k} &= \left(\mathcal{T}_k \odot \textstyle\sum_{j=1}^{k} \mathbf{W}_j + 0 \cdot \sum_{j=k+1}^{K} \mathbf{W}_j\right) \Big/ \textstyle\sum_{j=1}^{K} \mathbf{W}_j \\ &= \mathcal{T}_k \odot \textstyle\sum_{j=1}^{k} \mathbf{W}_j \Big/ \sum_{j=1}^{K} \mathbf{W}_j \ , \end{aligned} \quad (7a)$$

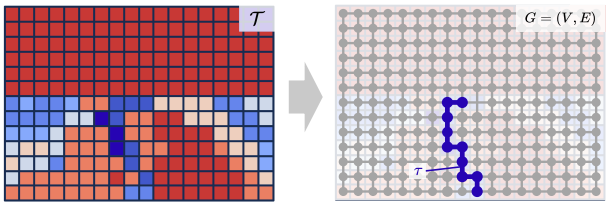

Fig. 4: $\mathcal{T}$ is modeled as an undirected graph $G = (V, E)$. A path is modeled as a sequence of vertices τ connected via G.

$$\overline{\mathcal{T}_k} = \left(\mathcal{T}_k \odot \textstyle\sum_{j=1}^{k} \mathbf{W}_j + 1 \cdot \sum_{j=k+1}^{K} \mathbf{W}_j\right) \Big/ \textstyle\sum_{j=1}^{K} \mathbf{W}_j, \quad (7b)$$

where $\mathbf{W}_j$ denotes the weight related to the j-th update predicted by the routers, and K is the total number of updates. Intuitively, the lower expectation map assumes that all remaining (unexecuted) experts label the environment as untraversable, while the upper expectation map assumes they label it as fully traversable. Then, the deviation of the current path cost from the final one is bounded as

$$|C(\mathcal{T}_k) - C(\mathcal{T}_K)| \leq \delta_k, \quad \textit{where} \quad \delta_k := C\left(\underline{\mathcal{T}_k}\right) - C\left(\overline{\mathcal{T}_k}\right), \quad (8)$$

and the bound δ_k is non-increasing with k, i.e.,

$$\delta_k \geq \delta_{k+1} \quad \forall k \in \{1, \ldots, K-1\}. \quad (9)$$

The full proof is provided in Section III-B.

The proposition implies that the path cost converges within a provably shrinking bound as more expert updates are aggregated. It guarantees that the path cost computed from a partially updated traversability map, obtained from only a subset of experts, deviates from the final cost computed using all experts by a bounded, computable margin.

The proposition establishes that activating all experts is unnecessary to obtain a satisfactory path solution, which provides theoretical justification for our lazy gating approach. Specifically, if the convergence bound δ falls below a predefined threshold ϵ, the path cost is considered sufficiently stable, and the influence of remaining expert predictions is deemed negligible. This enables early termination of the update loop. This improves computational efficiency without significantly compromising path quality, as the traversability map has already converged within an acceptable tolerance.

B. Proof of Convergence Bound

To support the proof, we model a traversability map $\mathcal{T}$ as an undirected graph $G = (V, E)$, where each vertex $v \in V$ corresponds to a pixel in $\mathcal{T}$ that is assigned a scalar traversability value. An edge $(v_i, v_j) \in E$ exists if the pixels are adjacent, capturing local connectivity. A path through the environment is defined as a sequence of vertices $\tau = [v_1, v_2, \cdots, v_t, \cdots, v_T]$ connected via G. The path planner seeks an optimal path τ^* by minimizing an objective function subject to feasibility constraints. We formulate the path planning problem within graph G as a constrained optimization problem $\mathcal{P}$:

$$\begin{aligned} \mathcal{P}(\mathcal{T}): \quad & C(\mathcal{T}) = \min\{f(\tau;\mathcal{T}) : \tau \in S(\mathcal{T})\} \\ & \text{s.t.} \quad f(\tau;\mathcal{T}) = \sum_{t=1}^{T} (1 - \mathcal{T}(v_t))^2 + f'(\tau) \\ & S(\mathcal{T}) = \{\tau | \mathcal{T}(v_t) \geq \theta \ \forall t, v_1 = v_{\text{src}}\} \end{aligned} \tag{10}$$

where $f(\tau;\mathcal{T})$ is the cost of τ under traversability map $\mathcal{T}$, and $S(\mathcal{T})$ is the set of feasible paths. The first term in f penalizes low-traversability paths, and f' indicates auxiliary objectives such as path length, smoothness, or dynamics constraints. The definition of the feasible path set $S(\mathcal{T})$ indicates that the traversability of every vertex in the path should be at least of threshold θ.

We proceed to prove the proposition in two parts. First, we establish that the bound δ_k is well-defined. Then, we show that δ_k monotonically decreases with increasing k.

1) Validity and Bound Definition.: From the definitions of $\underline{\mathcal{T}_k}$ and $\overline{\mathcal{T}_k}$ in Eqs. (7a) and (7b), it follows that for any vertex v_t along a path τ, we have:

$$\underline{\mathcal{T}_k}(v_t) \leq \mathcal{T}_k(v_t) \leq \overline{\mathcal{T}_k}(v_t) \tag{11}$$

The elementwise inequality in Eq. (11) implies that, for any fixed path τ, the traversability component of the cost function in Eq. (10) is bounded:

$$f(\tau;\overline{\mathcal{T}_k}) \leq f(\tau;\mathcal{T}_k) \leq f(\tau;\underline{\mathcal{T}_k}) \tag{12}$$

Given the inequality in Eq. (11) and Eq. (10), it follows that $S(\underline{\mathcal{T}_k}) \subseteq S(\mathcal{T}_k) \subseteq S(\overline{\mathcal{T}_k})$. The nesting of these feasible sets, combined with the elementwise cost bounds in Eq. (12), implies that $\mathcal{P}(\mathcal{T}_k)$ is a relaxation of $\mathcal{P}(\overline{\mathcal{T}_k})$, and $\mathcal{P}(\underline{\mathcal{T}_k})$ is a further relaxation of $\mathcal{P}(\mathcal{T}_k)$.

From the definition of problem relaxation, we obtain a corresponding ordering of the optimal path costs:

$$C(\overline{\mathcal{T}_k}) \leq C(\mathcal{T}_k) \leq C(\underline{\mathcal{T}_k}) \tag{13}$$

To extend this to the true final $\mathcal{T}_K$, we observe that the elementwise value of $\mathcal{T}_K$ is bounded between $\underline{\mathcal{T}_k}$ and $\overline{\mathcal{T}_k}$:

$$\underline{\mathcal{T}_k} - \mathcal{T}_K = -\sum_{j=k+1}^{K} \mathcal{T}_j \cdot \mathbf{W}_j \Big/ \sum_{j=1}^{K} \mathbf{W}_j \leq 0 \Rightarrow \underline{\mathcal{T}_k} \leq \mathcal{T}_K \tag{14}$$

$$\mathcal{T}_K - \overline{\mathcal{T}_k} = -\sum_{j=k+1}^{K} (1 - \mathcal{T}_j) \cdot \mathbf{W}_j \Big/ \sum_{j=1}^{K} \mathbf{W}_j \leq 0 \Rightarrow \mathcal{T}_K \leq \overline{\mathcal{T}_k} \tag{15}$$

Therefore, applying the same logic as above, we obtain:

$$C(\overline{\mathcal{T}_k}) \leq C(\mathcal{T}_K) \leq C(\underline{\mathcal{T}_k}) \tag{16}$$

Combining Eq. (13) and Eq. (16), we conclude:

$$|C(\mathcal{T}_k) - \mathcal{T}_K| \leq C(\underline{\mathcal{T}_k}) - C(\overline{\mathcal{T}_k}) \tag{17}$$

This establishes that $\delta_k = C(\underline{\mathcal{T}_k}) - C(\overline{\mathcal{T}_k})$ is a valid upper bound on the absolute error between the current and final cost estimates as stated in Eq. (8).

2) Monotonicity of δ_k: We now show that the convergence bound $\delta_k = C(\underline{\mathcal{T}_k}) - C(\overline{\mathcal{T}_k})$ is non-increasing over iterations.

First, we examine the difference between $\overline{\mathcal{T}_{k+1}}$ and $\overline{\mathcal{T}_k}$:

$$\overline{\mathcal{T}_{k+1}} - \overline{\mathcal{T}_k} = \left(\hat{\mathcal{T}}_{k+1} - 1\right) \cdot \frac{\mathbf{W}_{k+1}}{\sum_{j=1}^{K} \mathbf{W}_j} \leq 0 \Rightarrow \overline{\mathcal{T}_{k+1}} \leq \overline{\mathcal{T}_k} \tag{18}$$

Similarly, for the pessimistic estimate we have $\underline{\mathcal{T}_{k+1}} \geq \underline{\mathcal{T}_k}$.

Therefore, applying the same logic as in Section III-B.1, we obtain:

$$C(\overline{\mathcal{T}_k}) \leq C(\overline{\mathcal{T}_{k+1}}) \leq C(\underline{\mathcal{T}_{k+1}}) \leq C(\underline{\mathcal{T}_k}) \tag{19}$$

Therefore, the convergence bound satisfies:

$$\delta_{k+1} = C(\underline{\mathcal{T}_{k+1}}) - C(\overline{\mathcal{T}_{k+1}}) \leq C(\underline{\mathcal{T}_k}) - C(\overline{\mathcal{T}_k}) = \delta_k \tag{20}$$

This establishes that δ_k is monotonically non-increasing with k.

IV. Training Strategy

We first pretrain the gating networks on large-scale partially-aligned semantic segmentation datasets using human-prior-based label mappings to encourage generalization. Then, we fine-tune the router and experts end-to-end on small-scale multi-modal data with weak supervision derived from expert-label consistency.

A. Pre-Training on Large-Scale Partially-Aligned Data

The gating networks are first pretrained on large-scale data that contains various datasets with a wide domain distribution to learn the broad and general features. These data include 32,000 images from three single-sensory datasets [20], [21], [22], with each dataset corresponding to a distinct domain. However, it is impractical to directly use these datasets for training, as they contain only 2D images without synchronized depth information. This limitation makes it difficult to implement certain experts, such as model-based approaches that rely on geometric data. We solve this problem by mapping semantic or instance segmentation labels to expert

Classes	Trav. Cost	Ass. Exp.
sidewalk, floor, crosswalk, bike lane, etc	c_{free}	M
gravel, sand, snow, low grass, etc	c_{mid1}	NN
road, parking area	c_{mid2}	NN
terrain (tall grass, bush, dirt)	c_{mid3}	NN
person, curb, wall, other obstacles	c_{obs}	M

TABLE I: Semantics to assigned expert. M: model-based method; NN: Neural Networks. The definition of traversability cost c follows the prior work [19].

selection labels based on human knowledge. The label mapping encourages model-based approaches to handle terrains with clear geometric features, while assigning terrains where semantic information is more critical to traversability analysis to various neural networks. An example of the label mapping is shown in Table I. We encourage the gating networks to assign higher confidence to model-based experts with terrains such as roads or sidewalks. Meanwhile, higher confidence is encouraged to be assigned to experts based on neural networks on terrains such as grass or water surfaces.

Although human priors do not perfectly capture the performance of each expert across domains or terrains, our experiments demonstrate that this pre-training strategy effectively paves the way for the subsequent fine-tuning stage.

B. End-to-End Fine-Tunning for Task-Specific Optimization

Leveraging human priors eliminates the need for expensive multi-sensory data but introduces bias from human priors. Therefore, in the fine-tuning process, we train the model with a weakly supervised method on small-scale multi-modality data. We use 3,000 multi-sensory samples from both indoor and outdoor environments, including structured datasets [21], [23] and unstructured outdoor data [23], all featuring 2D images paired with synchronized depth maps as inputs.

Given the set of expert traversability predictions $\mathcal{T}_k$ and the ground-truth traversability map $\mathcal{T}_{GT}$, we generate a routing supervision label $\mathcal{W}_{\text{sup}}$ by selecting, at each pixel, the prediction from the expert whose output most closely matches the ground truth. Specifically, for each expert k, the pixel-wise squared error at location (i, j) is computed as $\varepsilon_k(i, j) = (\mathcal{T}_k(i, j) - \mathcal{T}_{GT}(i, j))^2$. The routing supervising label at pixel (i, j) is then determined by selecting the expert with the minimum error:

$$\mathcal{W}_{\text{sup}}(i, j) = \mathcal{T}_{k^*}(i, j), \quad \text{where} \quad k^* = \underset{k \in \{1,2,\ldots,K\}}{\arg\min} \; \varepsilon_k(i, j) \tag{21}$$

Applying this procedure across all pixel locations constructs the full supervision map for training the router. Guided by the router label $\mathcal{W}_{\text{sup}}$, we unfreeze and jointly fine-tune both the routers and experts to promote better alignment and co-adaptation. The domain router is trained to prioritize the expert whose prediction best matches the ground truth at each input, while each domain expert is fine-tuned to improve its performance on the terrain regions where it is selected as the best predictor.

V. Experiment Results

To evaluate the cross-domain traversability estimation ability and computational efficiency of NavMoE, we compare our approach with 4 state-of-the-art baselines that can run in real-time: Falco [5], Mask2Former [24] for indoor, MaskFormer [25] for structured outdoor, and GANav [11] for unstructured outdoor environment. Additionally, we include four ablation studies, NavMoE without fine-tuning and lazy gating (Ours w/o FT/LG), without lazy gating (Ours w/o LG), without lower-level lazy gating (Ours w/o LLG) and without fine-tuning (Ours w/o FT), to evaluate the individual contributions of these components to overall performance. Scene overviews for all three domains are shown in Fig. 5.A. The indoor experiment is conducted in the real world, while outdoor experiments use test sequences from [23], [26].

We evaluate performance using four metrics: **Motion Primitives Quality:** We introduce aligned-path traversability estimation mean square error (e_P) to measure the mean squared error between estimated $\mathcal{T}$ and ground-truth traversability $\mathcal{T}_{\text{gt}}$ along candidate motion primitives, which directly reflects planning quality with arbitrary goals. It can be formulated as: $e_P = \|S(\mathcal{T}) - S(\mathcal{T}_{\text{gt}})\|_2^2 \,/\, |S(\mathcal{T}_{\text{gt}})|$. **Traversability Map Quality:** We introduce traversability map estimation mean square error (e_M) to assess the accuracy of the predicted traversability map; **Computational Efficiency:** is quantified in GFLOPS; and **Normalized path quality (Q_p):** Given any path τ, goal p_{goal}, and $\mathcal{T}$, we define the path quality as $\eta(\tau) = 1 - f(\mathcal{T}, p_{\text{goal}})$, where $f(\mathcal{T}, p_{\text{goal}}) \in [0, 1]$ is defined in Eq. (4)). The path quality ratio Q_p is then formulated as: $Q_p = \eta(\hat{\tau})/\eta(\tau_{\text{gt}})$, where τ_{gt} represents the optimal ground truth path and $\hat{\tau}$ means the estimated path. This metric ranges between 0 and 1, with higher values indicating better path quality. Since our approach involves the lazy gating mechanism that doesn't need to produce a full $\mathcal{T}$ for path planning, we do not record e_P and e_M for our approach.

We report GFLOPs instead of runtime because some experts in our MoE framework are implemented in C++ while others in Python. Thus, we believe GFLOPs provide a fairer metric for comparison. We here report the average runtimes estimated in outdoor scenario to demonstrate our time efficiency: Gating network 6.0ms; Model-based expert 6.2ms; Learning-based expert 64.3ms; and path calculation 119.8ms, and the average processing time is 163.8ms. We believe this is sufficient for common navigation tasks, although the code implementation is not optimized yet.

Quantitative Results. As illustrated in Fig. 5.B, our method navigates complex environments, such as Fig. 5.B①, safely by primarily leveraging model-based experts while querying learning-based experts only when semantic cues are critical, achieving robust performance across all domains and terrain types. In Fig. 5.B③, learning-based expert [25] fail to strictly adhere to safety constraints, potentially causing the robot falling off a curb into traffic, while NavMoE utilizes geometric perception from model-based methods to maintain safety. Conversely, in Fig. 5.B② and ④, the learning-based experts' semantic perception correctly identifies non- or low-traversable areas (e.g., stop sign, grass), whereas the model-based method [5] mistakenly classify grass as traversable due to its geometric similarity to planar surfaces.

Cross-Domain Traversability Estimation Accuracy. As shown in Table II, NavMoE is the only method that can generalize well across all three domains, achieving best or second best performance among e_P, e_M and Q_p. In structured outdoor environment where learning-based expert [25] and Model-based one [5] have complementary expertise, the advantage of NavMoE becomes significant and we achieve the best performance among e_P, e_M and Q_p. In indoor

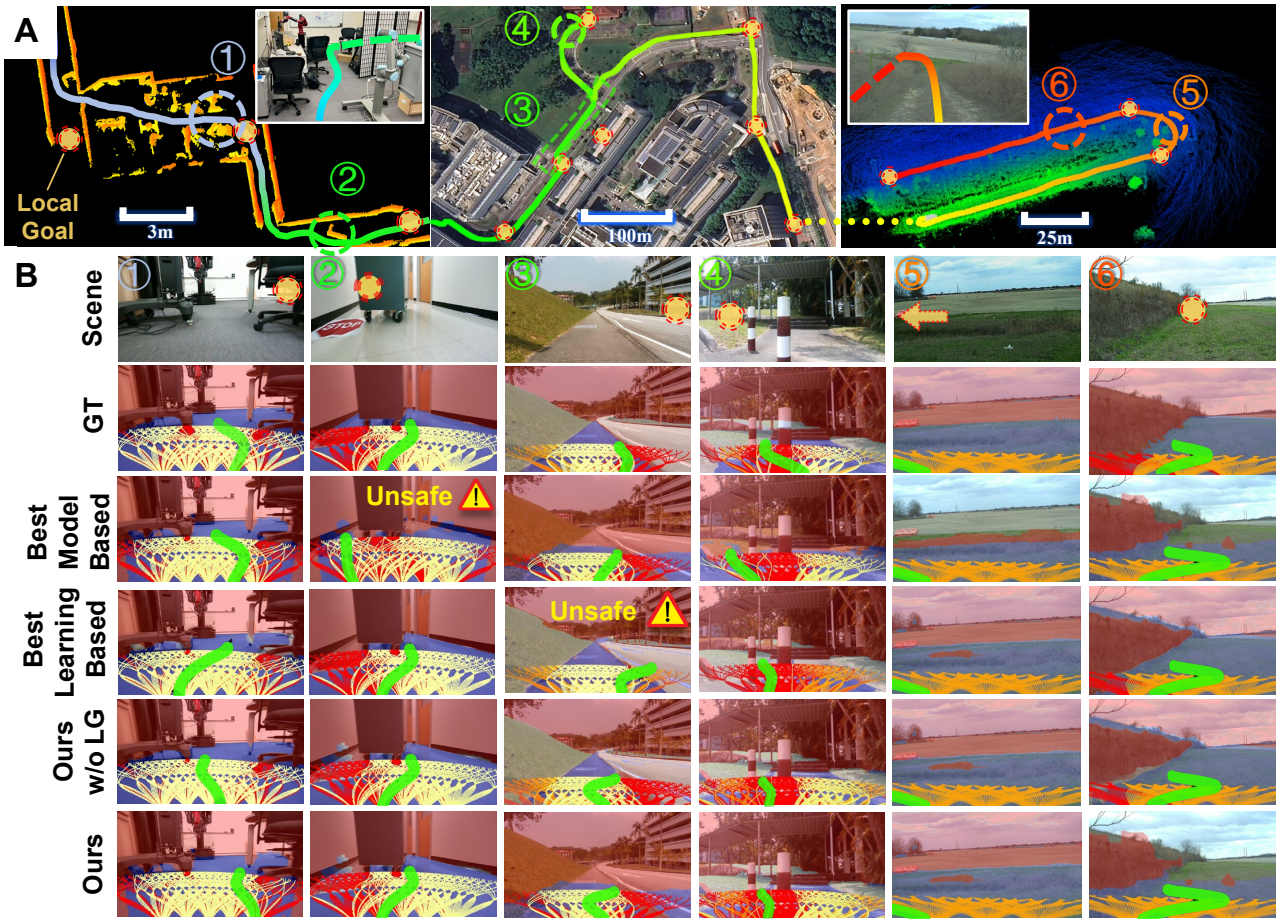

Fig. 5: **Sample frames from three domains. (A)** Overview of the test scenes. **(B)** Representative frame with results: a semi-transparent traversability map is overlaid on the RGB image, red-to-blue means non-to-fully traversable; candidate motion primitives are drawn on the map and colored by their average traversability from red to gold; the green curve marks the final path.

environment, our approach achieves comparable accuracy compared to domain-specific baseline [24] with only a 0.03 drop in Q_p. The same phenomenon extends to unstructured outdoor environment with a 0.01 drop in Q_p. This trend aligns with findings reported in other MoE works [14], [27], which indicates that MoE methods tend to deliver similar or occasionally slightly lower performance in domains where a single expert already excels. Additionally, our results are impacted by hardware misalignment and synchronization issues between the RGB camera and lidar or depth camera, which can be observed in the model-based method part in Fig. 5. **Effectiveness of FT.** Comparing NAVMOE w/o FT/LG to NAVMOE w/o LG, fine-tuning improves estimation accuracy up to 41% in e_P and 54% in e_M. When lazy gating is applied (Ours w/o FT v.s. Ours in Table II), fine-tuning balances Q_p and GFLOPS. Specifically, in the indoor scenario, fine-tuning shifts weights toward model-based methods, improving efficiency with small Q_p loss. Conversely, in outdoor environments, fine-tuning allocates more weights to learning-based methods, enhancing Q_p.

Efficiency and Effectiveness for Path Planning. Across all domains, NAVMOE has much higher efficiency than the best performing expert, marked in blue in Table II, by up to 76%. With the guaranteed solution quality introduced by the lazy gating mechanism, the resulting Q_p are within 0.03 compared to NAVMOE without LG, which is within the bound set as 0.05.

Effectiveness of LG. Comparing NAVMOE to our method without LG, lazy gating can decrease the computational overhead up to 96.3%, with only up to 0.02 loss in Q_p. Interestingly, we found the upper level lazy router achieves over 95% confidence and over 98% in accuracy across all domains, meaning that keeping the domain lazy router will not affect the estimation accuracy but significantly improve the computational efficiency, shown in Table II: Ours w/o LG and Ours w/o LLG. In this case, our method can still improve the computational efficiency by up to 82.6%, thereby reducing the average inference time to 164 milliseconds.

Overall, domain-specific approaches show satisfying performance within their area of expertise, they exhibit significantly weaker performance in out-of-distribution scenarios. NAVMOE demonstrates more robust and effective cross-domain traversability estimation accuracy. This reinforces our intuition that a robust and effective navigation model can be achieved by hierarchically combining experts with different expertise, paired with an efficient routing mechanism. Furthermore, our domain-gating mechanism allows the model to seamlessly scale with new learning-based navigation models across any domain, requiring only minimal fine-tuning of the gating networks.

Methods	Environments															
	Indoor				*Out. Struc.*				*Out. Unstruc.*				*Avg*			
	$e_P \downarrow$	$e_M \downarrow$	$Q_p \uparrow$	GFLOPS $\downarrow$	$e_P \downarrow$	$e_M \downarrow$	$Q_p \uparrow$	GFLOPS $\downarrow$	$e_P \downarrow$	$e_M \downarrow$	$Q_p \uparrow$	GFLOPS $\downarrow$	$e_P \downarrow$	$e_M \downarrow$	$Q_p \uparrow$	GFLOPS $\downarrow$
Falco [5]	0.213	0.100	0.87	0.13	0.157	0.098	0.9	0.13	0.104	0.048	0.64	0.13	0.158	0.082	0.74	**0.13**
Mask2Former [25]	**0.021**	**0.012**	**0.98**	246.31	0.091	0.033	0.87	246.31	0.083	0.057	0.80	246.31	0.065	0.034	0.88	246.31
MaskFormer[24]	0.470	0.187	0.74	48.3	0.070	0.026	0.92	48.3	0.069	0.033	0.81	48.3	0.203	0.082	0.83	48.3
GaNav [11]	0.811	0.403	0.33	18.69	0.729	0.265	0.17	18.69	**0.034**	**0.031**	**0.87**	18.69	0.525	0.233	0.56	18.69
Ours w/o FT/LG	0.039	0.028	**0.98**	313.18	0.064	0.034	0.96	313.18	0.048	0.035	0.84	313.18	0.051	0.032	0.92	313.18
Ours w/o LG	0.023	0.013	0.97	313.18	**0.055**	**0.021**	**0.98**	313.18	0.043	0.033	**0.87**	313.18	**0.040**	**0.022**	**0.94**	313.18
Ours w/o LLG	0.023	0.013	0.97	252.58	**0.055**	**0.021**	**0.98**	54.57	0.043	0.033	**0.87**	24.96	**0.040**	**0.022**	**0.94**	110.7
Ours w/o FT	-	-	0.95	122.04	-	-	0.96	9.47	-	-	0.84	10.28	-	-	0.92	52.16
Ours	-	-	0.95	94.83	-	-	**0.98**	11.58	-	-	0.86	16.25	-	-	0.93	40.22

TABLE II: **Quantitative results** of traversability estimation accuracy, path quality, and computational efficiency over multiple datasets.

VI. Conclusion

We presented NavMoE, a hierarchical and modular approach that combines the fast, reliable model-based methods with the semantic reasoning of learning-based approaches. By developing a hierarchical Mixture of Experts (MoE) architecture, we adaptively leverage specialized experts for different terrains, enhancing generalization across both seen and unseen environments. Key innovations include a lazy gating mechanism for improved computational efficiency and a two-stage training strategy that reduces reliance on high-quality multi-sensory data. Extensive experiments show that NavMoE reduces computational cost by 81.2% while maintaining path planning quality, outperforming existing methods in both efficiency and generalization beyond training domain. Our approach is highly extensible, enabling future breakthroughs in robotics by unifying non-differentiable and differentiable approaches across a variety of tasks.

References

[1] J. Ahtiainen, T. Stoyanov, and J. Saarinen, "Normal distributions transform traversability maps: Lidar-only approach for traversability mapping in outdoor environments," *Journal of Field Robotics*, vol. 34, no. 3, pp. 600–621, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21657

[2] A. Chilian and H. Hirschmüller, "Stereo camera based navigation of mobile robots on rough terrain," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4571–4576.

[3] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1184–1189.

[4] Y. Zhou, Y. Huang, and Z. Xiong, "3d traversability map generation for mobile robots based on point cloud," in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2021, pp. 836–841.

[5] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, "Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1300–1313, 2020.

[6] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[7] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler, faster, stronger," in *CVPR*, 2024.

[8] T. Guan, Z. He, R. Song, D. Manocha, and L. Zhang, "TNS: Terrain Traversability Mapping and Navigation System for Autonomous Excavators," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.

[9] T. Guan, R. Song, Z. Ye, and L. Zhang, "Vinet: Visual and inertial-based terrain classification and adaptive navigation over unknown terrain," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4106–4112.

[10] K. Weerakoon, A. J. Sathyamoorthy, J. Liang, T. Guan, U. Patel, and D. Manocha, "Graspe: Graph based multimodal fusion for robot navigation in outdoor environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8090–8097, 2023.

[11] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.

[12] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105151, 2022.

[13] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[14] H. Wang, F. M. Polo, Y. Sun, S. Kundu, E. Xing, and M. Yurochkin, "Fusing models with complementary expertise," *arXiv preprint arXiv:2310.01542*, 2023.

[15] Q. Zhu, Z. Sun, S. Xia, G. Liu, K. Ma, L. Pei, Z. Gong, and C. Jin, "Learning-based traversability costmap for autonomous off-road navigation," *arXiv preprint arXiv:2406.08187*, 2024.

[16] G. Haddeler, M. Y. Chuah, Y. You, J. Chan, A. H. Adiwahono, W. Y. Yau, and C.-M. Chew, "Traversability analysis with vision and terrain probing for safe legged robot navigation," *Frontiers in Robotics and AI*, vol. 9, p. 887910, 2022.

[17] B. Sebastian, "Traversability estimation techniques for improved navigation of tracked mobile robots," 2019.

[18] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[19] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter, "Viplanner: Visual semantic imperative learning for local navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5243–5249.

[20] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[21] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4990–4999.

[22] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[23] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 1110–1116.

[24] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[25] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Advances in neural information processing systems*, vol. 34, pp. 17 864–17 875, 2021.

[26] J. Zhang, H. Li, G. Peng, L. Zheng, T. Chia, S. T. Pan, M. Wen, Y. Ma, D. Wang *et al.*, "Continental-ntu delivery robot dataset for perception and navigation on footpath," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023.

[27] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A survey on mixture of experts," *arXiv preprint arXiv:2407.06204*, 2024.