IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

GrASPE: Graph based Multimodal Fusion for Robot Navigation in Outdoor Environments

Kasun Weerakoon[1], Adarsh Jagan Sathyamoorthy[1], Jing Liang[2], Tianrui Guan[2], Utsav Patel[2], and Dinesh Manocha[2]

***Abstract*—We present a novel trajectory traversability estimation and planning algorithm for robot navigation in complex outdoor environments. We incorporate multimodal sensory inputs from an RGB camera, 3D LiDAR, and the robot's odometry sensor to train a prediction model to estimate candidate trajectories' success probabilities based on partially reliable multimodal sensor observations. We encode high-dimensional multimodal sensory inputs to low-dimensional feature vectors using encoder networks and represent them as a connected graph. The graph is then used to train an attention-based Graph Neural Network (GNN) to predict trajectory success probabilities. We further analyze the number of features in the image (corners) and point cloud data (edges and planes) separately to quantify their reliability to augment the weights of the feature graph representation used in our GNN. During runtime, our model utilizes multi-sensor inputs to predict the success probabilities of the trajectories generated by a local planner to avoid potential collisions and failures. Our algorithm demonstrates robust predictions when one or more sensor modalities are unreliable or unavailable in complex outdoor environments. We evaluate our algorithm's navigation performance using a Spot robot in real-world outdoor environments. We observe an increase of 10-30% in terms of navigation success rate and up to 15% increase in AU-ROC compared to the state-of-the-art navigation methods. Project details are available at http://gamma.umd.edu/graspe/.**

***Index Terms*—Motion and Path Planning, Field Robots, Sensor Fusion**

I. Introduction

Mobile robots have increasingly been utilized in numerous outdoor applications such as delivery [1], agriculture [2], exploration [3], rescue [4] etc. These applications need the ability of the robots to navigate in challenging outdoor environmental conditions such as low lighting, cluttered vegetation, etc. In this work, we consider such environments as *unstructured outdoor environments*.

The robots' perception could encounter noise, occlusions, or other error modes and failures when navigating in such environments. Especially, cameras undergo motion blur, low lighting, and occlusions [5], [6], while LiDAR point clouds experience heavy distortions/scattering in cluttered vegetation [7]. A key issue is developing methods that can perform reliable perception and planning computations by taking into account such sensor uncertainties.

In terms of robot perception and multi-modal sensor fusion, many recent Deep Learning (DL) techniques have demonstrated promising results. These methods combine distinct feature representations from different sensor modalities such as cameras, IMU, and LiDAR [10], [11] or perform only camera-LiDAR fusion [12]. Many of these works in sensor fusion incorporate Convolutional Neural Networks (CNN) or other linear operators to encode spatial information. However, such operators cannot capture the complex correlations between different modalities to fuse them effectively [13]. Multi-sensor data represented as graphs inherently overcome this limitation since graph edges can be defined to model the complex correlations in the data. Furthermore, recently developed Graph Neural Networks (GNN) [14] provide powerful learning and fusion capabilities on graph data as opposed to traditional neural networks such as CNNs. A key challenge is developing reliable methods for outdoor scenes since most existing DL and GNN methods [15] are formulated for indoor scenarios and do not consider the variable reliability of the cameras and LiDAR outdoors.

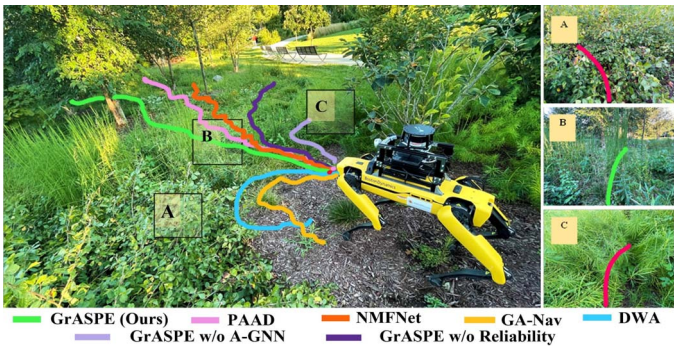

Fig. 1. Spot robot trajectories while navigating in an unstructured outdoor terrain using various methods. We consider that a trajectory is non-traversable/unsuccessful if it leads the robot to collisions, stumbling, or similar modes of failure. Our method GrASPE, utilizes multi-modal sensory inputs from RGB camera, 3D LiDAR, and the robot's odometry to estimate the future success probability of a candidate trajectory to avoid obstacles and other non-traversable regions such as *A* and *C*. Multi-modal fusion methods such as PAAD [6] and NMFNet [8] provide inconsistent perception under extremely cluttered settings. LiDAR-based methods such as DWA [9] recognize *A, B*, and *C* regions as obstacles due to the cluttered vegetation even though region B is actually traversable. GrASPE identifies trajectories that are traversable by intelligently fusing such partially reliable camera and LiDAR data.

Main Contributions: We present GrASPE(**Gr**aph **A**ttention based **S**ensor fusion for **P**ath **E**valuation), a novel trajectory traversability estimation and planning algorithm for legged robot navigation in unstructured outdoor environments. We incorporate multi-modal observations from an RGB camera, 3D LiDAR, and robot odometry to train a prediction model to estimate candidate trajectories' navigation success probabilities. Our model learns the correlation between multi-sensor data under unstructured outdoor conditions where the camera undergoes occlusions, motion-blur, and low-lighting, while the LiDAR point cloud

Manuscript received: May 16, 2023; Revised August 10, 2023; Accepted September 05, 2023. This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by ARO Grants W911NF2110026, and Army Cooperative Agreement W911NF2120076. We acknowledge the support of the Maryland Robotics Center.

[1]Authors are with Dept. of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA. {kasunw, asathyam}@umd.edu

[2]Authors are with Dept. of Computer Science, University of Maryland, College Park, MD, USA. {jingl, rayguan, upatel22}@umd.edu, dm@cs.umd.edu

experiences scattering and distortions. We further explicitly evaluate the reliability of the camera and LiDAR data for achieving *reliability-aware* multi-modal fusion in our prediction model. The key contributions of our approach include:

- A novel trajectory success prediction model that estimates a candidate trajectory's probability of avoiding navigation failures such as collisions, getting stuck, etc. based on partially reliable multi-modal sensor observations in unstructured outdoor environments. We project the high-dimensional sensor inputs (RGB images and point clouds) into feature vectors and represent them as a *feature graph* to train an Attention-based GNN (A-GNN) for sensor fusion. Our model accurately estimates trajectories' success probabilities even in poorly lit, densely vegetated environments in real-time which results in upto 15% increase in AU-ROC compared to other methods.
- A novel reliability estimation method to quantify the usefulness of the image and 3D point cloud data for sensor fusion. Our formulation counts the number of features such as corners for RGB images, and edges and planes for point clouds to compute reliability and uses it to weigh the edges in the feature graph. Weighing based on reliability suppresses the correlations between unreliable sensor modalities leading to predictions only based on reliable sensors. Including reliability in feature graph improves the success rate by 20-40% compared to the predictions made without the reliability measures.
- We show that our feature graph's construction using the reliability metric is undirected and non-negatively weighted. We prove that its Laplacian is spectrally decomposable, thereby allowing graph convolution operations to be utilized in our GNN. This helps to learn complex node embeddings from our feature graph.
- A local planner that computes dynamically feasible, collision-free, and traversable velocities accounting for their success probabilities. We demonstrate our algorithm on a Boston Dynamics *Spot* robot to evaluate its performance in real-world outdoor scenarios with variable lighting and vegetation density. Our method results in an 10-30% improvement in terms of success rate compared to state-of-the-art methods.

II. Related Work

In this section, we discuss the existing work on multi-modal sensor fusion and anomaly detection in the context of robot navigation.

A. Multi-modal Sensor Fusion for Navigation

Robot perception in real-world environments can be challenging due to varying conditions such as lighting, noise, occlusions, motion blur, etc. To this end, prior algorithms incorporate uncertainty modeling and adaptation techniques for robot navigation using a single sensor modality [16]. However, such methods perform well only under controlled outdoor settings. Therefore, multi-sensor fusion methods are used to mitigate unreliable perception from individual sensors [17].

Deep learning methods have also been widely used for sensor fusion [18]. Particularly, the camera-lidar fusion is employed in navigation literature to combine visual and geometric features [19]. Several algorithms treat the multi-modal fusion-based navigation as a Traversability Estimation (TE) task where the robot's trajectories are evaluated during perception to identify candidate actions or regions for navigation [11], [20]. In [21], RGB and depth data along with navigation experience are used to estimate the traversability of the trajectories. Further, end-to-end control policies have been used in literature to generate navigation actions from multi-modal fusion [8], [22]. However, these methods typically assume that all the input sensory perceptions are reliable in terms of feature availability. In contrast, our proposed algorithm deals with real unstructured outdoor environments where this assumption is inapplicable.

B. Anomaly Detection based Prediction Models

Several recent works have incorporated anomaly detection algorithms to identify collisions or failures during navigation as anomalies based on multi-sensory observations [23], [24]. Such algorithms can be trained using simple positively (traversable) and negatively (non-traversable, collision, etc) labeled observations, which can be created trivially, as opposed to the extensive labeling required for training supervised learning methods [25], [26]. In [27], RGB and depth images are used to train a predictive model that generates an anomaly mask that reflects "known" and "unknown" regions. They assume that the observations have constant illumination and are feature-rich. Since the prediction from these methods do not account for potential future navigation failures and collisions, they could lead to catastrophic accidents during navigation.

C. Proactive Anomaly Detection

To deal with potential future navigation failures due to sensor uncertainties, *proactive anomaly detection* methods have been utilized to estimate the probability of such failures using predicted trajectories, and multi-sensor fusion [6]. A Supervised Variational AutoEncoder (SVAE) is used in [23] for failure identification in unstructured environments using 2D LiDAR data. This SVAE is used in PAAD [6] as a feature encoder to identify navigation anomalies proactively in crop fields using multi-modal fusion. However, the 2D LiDAR observations are heavily distorted in cluttered vegetation such as tall grass where 3D point clouds can provide richer information. Other navigation models [25] execute actions by avoiding undesirable maneuvers in cluttered environments based on the current visual sensory observations. However, such vision-based systems lead to erroneous predictions during camera occlusions and low light conditions. Our proposed method uses a camera, 3D LiDAR, and odometry sensor fusion strategy to achieve better perception in such conditions for robot navigation.

III. Background and Problem Formulation

In this section, we summarize the notations, problem formulation, and some background to the Graph Neural Networks (GNNs) used in our approach.

A. Symbols and Definitions

At any time t, our formulation's input observation space O_t uses four inputs: an RGB image I_t^{rgb}, 3D point clouds from a LiDAR P_t^{lidar}, a history of the robot's velocities V_t of length T, and a binary image of the extrapolated trajectory from the robot's current velocity I_t^{traj}. For our method's

training, we use a ground truth label vector E_t of a trajectory's success probability. Our method accounts for the reliability of its image and point cloud inputs using measures denoted as $r_{img} \in [0,1]$, and $r_{point} \in [0,1]$ respectively.

B. Problem Formulation

Our formulation for GrASPE can be stated as follows:

Formulation III.1. *To predict the success probability vector $\hat{E}_t = [e_t, ...e_{t+T-1}]$ with $e_t \in [0,1] \forall t$ of a robot trajectory using its local multi-modal fused observations O_t while accounting for the variable reliability (r_{img}, r_{point}) of the camera and LiDAR observations respectively.*

We define *success probability* as a trajectory's probability of not encountering collisions, stumbling, or similar modes of failure. The success probability is then used to evaluate trajectories that the robot can use for navigation. The overall system architecture of the prediction model is presented in Fig. 2.

C. Graphs and Graph Neural Networks

Real-world data modalities (e.g., images and point clouds) used for sensor fusion exhibit complex correlations between them. For instance, a rock captured in an image from a robot is represented by RGB values, as XYZ values in point clouds, and spikes in 6-DOF IMU readings when the robot runs over it. For accurate sensor fusion, these complex correlations cannot be represented by regular data structures in the Euclidian domain.

Representing the data streams as a graph offers the capability of modeling the complex correlations between them as weighted edge connections. Therefore, in our work, sensor data is processed and represented as a weighted graph $\mathcal{G} := (\mathcal{V}, \mathcal{W})$, where $\mathcal{V} = 1, 2, \ldots, N$ is a set of N nodes/vertices associated with elements of the sensors' feature vectors. $\mathcal{W}$ is a weighted adjacency matrix with entries $\mathcal{W}_{i,j}$ representing the strength of the connection (edge) between feature elements i and j.

Graph Neural Networks (GNNs) are a class of neural networks that operate on data represented as graphs. We incorporate Graph Convolution Network (GCN) [28] and Graph Attention Network (GAT) [29] operators in our GNN to encode and pay attention to the local graph structure and node embeddings. The convolution and attention mechanisms can encode node features and correlations between the graph nodes respectively. Especially, the GAT operator performs graph attention in four steps: Linear transformation, Leaky ReLU activation, Softmax normalization, and Multi-head attention which often outperforms traditional multi-head attention approaches on Euclidean data structures.

IV. Our Algorithm: GRASPE

In this section, we present the details of our proposed algorithm. We first discuss the components of the prediction model trained for the trajectory's success probability estimation: 1. Acquiring multi-modal data and reducing its dimensionality; 2. Estimating camera and LiDAR's reliability; 3. Graph-based prediction model. Finally, we explain the details of how robust perception is used for planning.

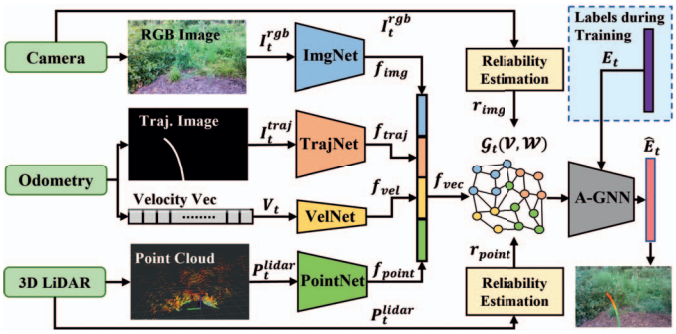

Fig. 2. GrASPE System Architecture for candidate trajectories' success probability estimation. We utilize RGB images, 3D point clouds, the robot's velocity history, and the predicted trajectory of the robot as an image to train a novel graph-based prediction model to estimate the given trajectory's probability of success in terms of navigation.

A. Multi-modal Observations

We design our success probability estimation model using the following multi-sensor inputs as observations: RGB image $I_t^{rgb} \in \mathbb{R}^{w \times h \times 3}$, 3D point cloud $P_t^{lidar} \in \mathbb{R}^{3 \times N_p}$, robot's velocity history $V_t = \{(v,\omega)_{t-T+1}, ..., (v,\omega)_t\}$ and predicted trajectory from the robot's current velocity as an image $I_t^{traj} \in \mathbb{R}^{w \times h \times 1}$ (see Fig. 2). To obtain I_t^{traj}, we first extrapolate the robot's trajectory for the next $T_{hor} = 10$ time steps based on the current velocity $(v,\omega)_t$. Then the resulting trajectory is projected to a blank image using homography projection. We restrict our robot's velocity space V_s such the extrapolated trajectories in I_t^{traj} lie within the field of view of the RGB image. Hence, the velocities that lead to sharp maneuvers are excluded from the velocity space while purely angular velocities are kept to allow the robot to rotate in place in complex scenarios. The RGB image and 3D point cloud capture visual and geometric features of the environment respectively. The robot's velocity history is used to account for the robot's recent behavior. The ground truth labels for predicting a trajectory's success are represented by a vector E_t of 0's (for portions of the trajectory with failures) and 1's (for portions with successes) of length T.

Finally, the prediction model uses the observations $O_t = \left[I_t^{rgb}, P_t^{lidar}, V_t, I_t^{traj}\right]$, and the ground truth labels to train to estimate the success probability vector of the predicted trajectory $\hat{E}_t = [e_t, ...e_{t+T-1}]$. Details of the observation data collection and ground truth labeling are described in Section V-B.

B. Multi-modal Feature Vector Generation

Different sensor modalities capture different environmental features (images capture visual features, point clouds capture edges and surfaces), and fusing them effectively is non-trivial. Moreover, raw image and point cloud data processing are computationally expensive due to the high dimensionality. Therefore, we pre-process each sensor input using separate feature encoding networks to obtain dimension-reduced feature vector representations denoted by $f_{img}, f_{point}, f_{vel}$, and f_{traj}.

The sizes of these feature vectors are chosen empirically considering the trade-off between the resulting graph size (effects on the computation complexity) and the feature encoding quality (smaller feature vectors may not properly encode the input data).

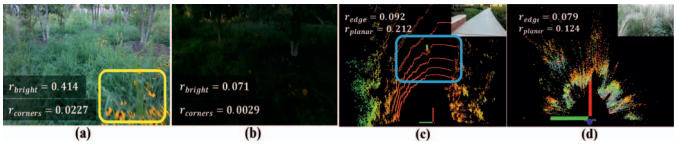

Fig. 3. **Image & Point cloud Reliability Estimation:** (a) and (b) demonstrate images of the same scene under two different lighting conditions. Regions with poor lighting and motion blur (yellow rectangle in (a)) have a significantly lower number of features. Similarly, the motion blur and camera occlusions result in images with low reliability values. (c) Top view of a point cloud with many edge and planar features (blue rectangle), (d) Distorted point cloud observed in cluttered vegetation such as tall grass, and the corresponding RGB image (top right).

1) Visual Feature Extraction: We utilize a ResNet[30] based pipeline to extract image features f_{img} from the camera RGB image I_t^{rgb}. We incorporate a CBAM [31] module between the network layers to perform spatial and channel attention toward the important visual features. The input image I_t^{rgb} is of size 320×240. The ImgNet branch in Fig. 2 includes a ResNet18 backbone with CBAM layers added similar to [31]. The output feature vector f_{img} of dimension 40×1 is obtained by passing through a $Sigmoid()$ activation layer.

2) LiDAR Feature Extraction: A Pointnet [32] based network is incorporated to extract point cloud features f_{point} from the LiDAR data P_t^{lidar}. To reduce the data size, we restrict LiDAR point cloud to $[-\pi/2, \pi/2]$ field of view w.r.t. the robot's heading direction (see Fig. 3). P_t^{lidar} is of dimensions $3 \times N_p$, where $N_p = 10000$ is the number of points captured from the LiDAR at each time t. The output f_{point} of dimensions 40×1 is obtained by passing through a $Sigmoid()$ activation layer at the end similar to ImgNet.

3) Velocity Feature Extraction: A velocity vector V_t of size 100×1 includes linear and angular velocities of the previous 50 time steps. To obtain this, (v_t, ω_t) data of size 50×2 is reshaped to a vector V_t of size 100×1. This V_t is fed into 4 linear convolutional layers with dilation size $\{3, 2, 2, 2\}$ in VelNet to obtain the velocity features f_{vel} of length 20.

4) Trajectory Feature Extraction: The predicted trajectory image $I_t^{traj} \in \mathbb{R}^{320\times240\times1}$ passes through 3 pooling and 3 convolutional layers with kernel size $\{3, 2, 2\}$, a flatten layer and $Sigmoid()$ activation layer to obtain the trajectory feature vector f_{traj} of length 20. Hence, the final feature vector f_{vec} is of dimensions 120×1 after concatenating $f_{img}, f_{point}, f_{vel}$ and f_{traj}.

C. Sensor Reliability Estimation

RGB images and the point cloud data become unreliable in certain outdoor conditions (e.g. low luminance, scattering of laser rays, etc.). Even though the feature encoding networks proposed in the literature and in Section IV-B1 are capable of feature extraction from images and point clouds, they do not account for their reliability.

Therefore, we estimate the reliability of the image and point cloud inputs at each time step quantitatively using classical image and point processing methods that execute in real-time [33], [34]. We further assume that the instantaneous robot velocities obtained from its odometry are not significantly affected by the environment and are reliable.

1) Image Reliability Estimation: We consider that the images captured by the camera have high reliability if they have: 1. High image brightness, and 2. Availability of visual features such as corners, edges, etc [35]. See Fig. 3 for sample

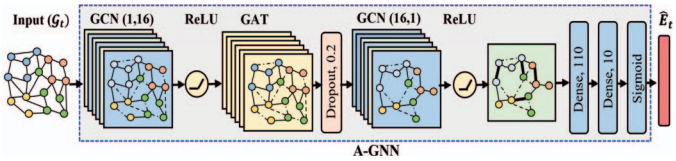

Fig. 4. Architecture of the attention-based graph neural network (A-GNN) used in our GrASPE system. The *reliability-aware* feature graph $\mathcal{G}_t$ is fed into this A-GNN to encode and attend to the useful multi-modal sensory feature interactions.

reliability comparisons. To estimate the image brightness, we first convert the input RGB image I_t^{rgb} to a gray-scale image I_t^{gray}. Then we calculate the Root Mean Square(RMS) value of the histogram distribution of I_t^{gray} as follows to obtain brightness estimation,

$$r_{bright} = RMS(hist(I_t^{gray})) \tag{1}$$

Here, $hist$ is the histogram operator and RMS is the root mean square operator where $RMS(x) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$.

To estimate the availability of useful features in the image, we calculate the number of corner features n_c in the input image I_t^{rgb} using FAST (Features from Accelerated Segment Test) algorithm [36]. These corner features are fast to compute and inherently reflect the non-blurriness and well-lit condition, therefore a good measure of the image's reliability. We consider the image is feature rich if the n_c is higher than a threshold. Hence, $r_{corners} = \frac{n_c}{(w\times h)}$. Here, w and h are the height and width of the input RGB image. The final image reliability measure is obtained as a normalized scalar value $r_{img} \in [0, 1]$ using the weighted sum of r_{bright} and $r_{corners}$:

$$r_{img} = \alpha_b r_{bright} + \alpha_c r_{corners} \tag{2}$$

2) Point Cloud Reliability: We observe that the LiDAR point cloud is heavily distorted in the presence of unstructured vegetation leading to poor estimation of the surrounding objects' geometries. Therefore, to evaluate the point cloud reliability, we perform edge and planar feature extraction to calculate the number of 3D features available in the point cloud P_t^{lidar} at a given time t. High numbers of edges and planes in the point cloud denote low scattering, and therefore a structured environment.

Let X_l be the l^{th} 3D point in the input point cloud P_t^{lidar} and let $\mathcal{M}$ be the set of points in the neighborhood of X_l acquired from an instance of the point cloud. Then, we can obtain the local surface smoothness evaluation factor c_l of the point l as,

$$c_l = \frac{1}{||X_l||.|\mathcal{M}|}\Big\| \sum_{k\in\mathcal{M}, k\neq l} \big(X_l - X_k\big)\Big\|. \tag{3}$$

Zhang et al. [34] demonstrate that the points with higher and lower c_l values obtained from equation 3 belong to edge (non-smooth) and planar (smooth) features respectively. Hence, we define two threshold values c_{max} and c_{min} as the minimum and maximum smoothness thresholds to consider a given point on a surface as an edge or a plane respectively. The resulting edge and planar feature point sets can be denoted as:

$\mathcal{S}_{edge} = \{l | c_l \geq c_{max}, l \in P_t^{lidar}\}$ and $\mathcal{S}_{planar} = \{l | c_l \leq c_{min}, l \in P_t^{lidar}\}$ respectively.

$$r_{edge} = \frac{|\mathcal{S}_{edge}|}{|P_t^{lidar}|}, \quad r_{planar} = \frac{|\mathcal{S}_{planar}|}{|P_t^{lidar}|}. \tag{4}$$

Finally, the point cloud reliability measure r_{point} is derived as a weighted combination of r_{edge} and r_{planar} using two tunable parameters β_e and β_p.

$$r_{point} = \beta_e r_{edge} + \beta_p r_{planar}. \tag{5}$$

Fig. 3(c) and (d) present an example comparison of r_{edge} and r_{planar} in real outdoor scenarios.

D. Reliability-aware Graph Construction

Once the sensor inputs are encoded into feature vectors (Section IV-B), we concatenate them to obtain a combined vector $f_{vec} = [f_{img}, f_{point}, f_{traj}, f_{vel}]$. The i^{th} element of $f_{vec} \in [0,1]^N$ ($N = 120$) becomes the i^{th} node of the graph representation for fusion.

The weighted adjacency matrix $\mathcal{W}$ represents the edge weights between the graph nodes. In our context, they represent the similarity between the encoded feature elements in f_{vec}. In related literature, edge weights are calculated using a distance measure (e.g., absolute difference or L_2 norm) since the distance between the values residing on two nodes is inversely proportional to the similarity between the node values [37], [14]. We modify these edge weights between the nodes corresponding to the image and point cloud features based on the reliability measures (Section IV-C) as follows.

Let $\mathcal{W}_{i,j}$ be the edge weight between the feature elements i and j in f_{vec}.

$$\mathcal{W}_{i,j} = \begin{cases} exp^{-\{\lambda \cdot |f_{vec}(i) - f_{vec}(j)| \cdot (2 - r_{i,j})\}}, & \text{if } i \neq j \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where, $\lambda > 0$ is a tunable scalar and $r_{i,j} \in [0,1]$ represents the reliability measure between the node i and j as follows,

$$r_{i,j} = \begin{cases} r_{img}, & \text{if } \{i,j | i \in f_{img}, j \notin f_{point}\} \\ r_{point}, & \text{if } \{i,j | i \notin f_{img}, j \in f_{point}\} \\ \frac{1}{2}\big(r_{point} + r_{img}\big), & \text{if } \{i,j | i \in f_{img}, j \in f_{point}\} \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

We can observe that $r_{i,j} = r_{j,i} \forall i,j$. Our formulation leads to higher edge weights when the values residing on the nodes are similar and belong to the same sensor modality. Finally, we refer to the derived graph representation $\mathcal{G}_t$ of the feature vector f_{vec} as a *reliability-aware* feature graph.

E. A-GNN Architecture

We design a light-weight A-GNN presented in Fig.4 to predict the success probability vector $\hat{E}_t$ for a given trajectory. We first pass our reliability-aware feature graph $\mathcal{G}_t$ through a 16-channel Graph Convolutional layer (GCN) to encode the node features. Secondly, a Graph Attention network (GAT) is utilized to identify the important/strongly correlating neighbors of each node of the 16-channel graphs. Next, a dropout layer is incorporated for regularization to minimize the risk of network over-fitting. The output graph is obtained after passing through another GCN layer. We use the $ReLU$ activation layer after each GCN layer. The output graph is concatenated into a vector and passed through two dense layers and $sigmoid$ activation to obtain the output prediction vector $\hat{E}_t$. Hence, the overall prediction pipeline can be treated as a mapping function $\psi_{GrASPE} : O_t \rightarrow \hat{E}_t$.

F. Theoretical Validation of Graph Construction

Lemma 1. *GrASPE's reliability aware feature graph $\mathcal{G}_t$ generated using the weight matrix $\mathcal{W}$ ensures that $\mathcal{G}_t$ is an undirected graph with non-negative weights.*

Proof. Let $w_{i,j} \in \mathcal{W}$. By construction, $r_{i,j} = r_{j,i} \, \forall \, i,j,$ and $i \neq j \implies w_{i,j} = w_{j,i} \, \forall \, i,j,$ and $i \neq j$ and $w_{i,j} = 0 \, \forall \, i,j,$ and $i = j$. This implies that $\mathcal{W}$ is symmetric and $\mathcal{G}_t$ is undirected. Let $m_{i,j} = \lambda |f_{vec}(i) - f_{vec}(j)| \cdot (2 - r_{i,j})$ from the Eq.6. Then, $m_{i,j} \geq 0, \forall i,j$ because $\lambda > 0$ and $r_{i,j} \in [0,1]$. Therefore, $exp^{-m_{i,j}} > 0 \, \forall \, i,j, i \neq j \implies w_{i,j} \geq 0 \, \forall \, i,j \implies \mathcal{W}$ has non-negative weights. □

GCN layers in our A-GNN perform spectral graph convolution which requires the input graph's Laplacian $\mathcal{L}$ to be spectrally decomposable [28]. Please refer to [38] for the proof that our graph's construction leads to a spectrally decomposable Laplacian.

G. Reliability-aware Planning

Our local planner adapts the Dynamic Window Approach (DWA) [9] to perform navigation while evaluating the success probabilities of the generated trajectories using GrASPE.

We represent the robot's actions as linear and angular velocity pairs (v, ω). Let V_s be the space of all the possible robot velocities. DWA considers two velocity space constraints: (1) The dynamic window V_d contains the dynamically feasible velocities during the next Δt time window; (2) The admissible velocity space V_a includes the collision-free velocities. Hence, the restricted velocity space $V_r = V_s \cap V_d \cap V_a$.

The collision estimations for V_a are generally identified using 2D LiDAR scans. However, the LiDAR scan is highly distorted and unreliable in cluttered vegetation. Hence, the traditional DWA will identify traversable cluttered regions as obstacles. To avoid such misclassifications, we omit the admissible velocity space constraint when the point cloud is highly unreliable. That is,

$$V_r = \begin{cases} V_s \cap V_d, & \text{if } r_{point} \leq r_{th} \\ V_s \cap V_d \cap V_a & \text{otherwise,} \end{cases} \tag{8}$$

where r_{th} is a reliability threshold that indicates the point cloud data is highly distorted. The resulting velocity space V_r is used to calculate the optimal velocity pair (v^*, ω^*) by maximizing the objective function below,

$$Q(v,\omega) = \sigma\big(\gamma_1.heading(v,\omega) + \gamma_2.dist(v,\omega) + \gamma_3.vel(v,\omega)\big), \tag{9}$$

where $heading(.), dist(.)$, and $vel(.)$ are the cost functions in [9] to quantify the robot's heading towards the goal, distance to the obstacles, and the robot's velocity, respectively. σ is a smoothing function and $\gamma_i, i = 1,2,3$ are adjustable weights. The resulting optimal velocity pair can be denoted as $(v^*, \omega^*) = argmax(Q(v,\omega))$.

We calculate the predicted trajectory for the next T time steps considering the optimal (v^*, ω^*) as the robot's immediate next velocity. Then, the obtained trajectory is fed into GrASPE model to estimate the trajectories' success probabilities $\hat{E}_t$. If $min(\hat{E}_t) \geq e_{th}$, the velocity pair (v^*, ω^*) corresponding to that trajectory is considered navigable. Otherwise, the (v^*, ω^*) is removed from velocity space V_r, and a new optimal velocity pair is calculated using the objective function defined in equation 9. Here, e_{th} is an adjustable threshold. Please refer to [38] for the algorithm summary and default threshold values.

V. Results and Analysis

We detail our method's implementation and experiments conducted on a Spot robot. Then, we perform ablation studies and comparisons to highlight the benefits of our algorithm.

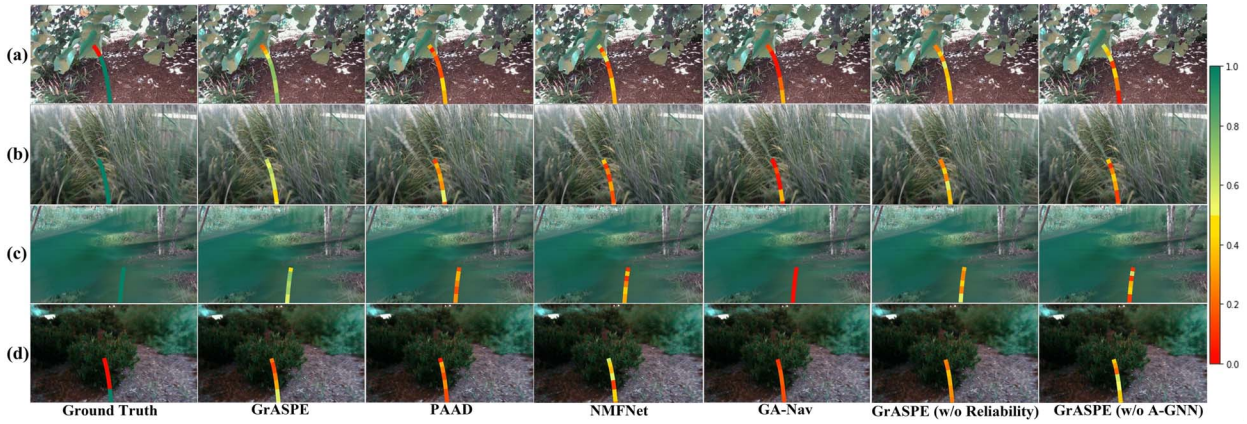

Fig. 5. Navigability predictions from GrASPE(Ours), PAAD[6], NMFNet [8], GA-Nav[39], GrASPE w/o reliability, and GrASPE w/o A-GNN under different environmental conditions compared to the Ground Truth. (a) Hanging leaves; (b) Pliable tall grass; (c) Camera occlusion; (d) Low-light condition. GrASPE predictions outperform the other methods under varying conditions that are critical for robot navigation in unstructured outdoor environments.

A. Implementation

GrASPE is implemented using PyTorch and PyTorch Geometric (PyG). The prediction model is trained in a workstation with an Intel Xeon 3.6 GHz processor and an Nvidia Titan GPU using real-world data collected from a Spot robot. The robot is equipped with an Intel NUC 11 (a mini-PC with Intel i7 CPU and NVIDIA RTX 2060 GPU), a VLP16 LiDAR, and an Intel RealSense L515 camera.

B. Dataset

The multi-modal dataset used in this work is collected by operating the Spot robot under different lighting conditions in an outdoor field that includes bushes, small trees, hanging leaves, and grass regions of different heights and densities. We incorporate a randomized planner to collect I_t^{rgb}, P_t^{lidar} and V_t from the robot to minimize human effort. The ground truth labels are created part manually and part automatically. Successful navigation tasks from the randomized planner during the data collection are automatically labeled with 1's in the E_t vector. The scenarios with failures or oscillations are manually checked to generate ground truth labels E_t. Since our ground truth label vector E_t contains binary elements, the manual labeling process is relatively simple and requires minimal effort. The final dataset includes a set of observations and label pairs (O_t, E_t) for each time step t. The training set includes 28721 positive samples and 8463 negative samples whereas the test set contains 4537 positive samples and 1096 negative samples.

C. Evaluations

We use the following evaluation metrics to compare our method's performance against: PAAD [6], NMFNet [8], GA-Nav [39], and DWA [9] algorithms. PAAD is a proactive anomaly detection-based navigation method that fuses images, 2D LiDAR scans, and the robot's future trajectory using Multi-head attention. NMFNet is a deep multi-modal fusion network that has three branches: 2D laser, RGB images, and point cloud data for action prediction. We use an existing pointcloud-to-laserscan ROS package to derive 2D LiDAR scan data required for PAAD and NMNet from the 3D point cloud. GA-Nav is a multi-modal navigation framework that combines image semantic segmentation with elevation maps for traversability estimation. DWA is a local planner that utilizes a 2D LiDAR scan for obstacle avoidance.

We further compare the GrASPE model without reliability estimations (GrASPE w/o reliability) and without the A-GNN model (GrASPE w/o A-GNN) for ablation studies. PAAD, NMFNet, and the two ablation models are trained using the same dataset (Section V-B) for evaluations. GA-Nav is trained on publicly available outdoor datasets (RUGD [40] and RELLIS-3D [41] that contains trees and grass) since it requires pixel-level segmentation labels.

Success Rate - The number of successful goal-reaching attempts (while avoiding non-pliable vegetation and collisions) over the total number of trials.

Normalized Trajectory Length - The ratio between the robot's trajectory length and the straight-line distance to the goal in both successful and unsuccessful trajectories.

AU-ROC Score - A threshold-independent metric indicating the Area Under the Receiver Operating Characteristics. It quantifies the model's ability to discriminate between positive and negative examples. A greater value of AU-ROC denotes better model performance.

D. Test Scenarios

We evaluate our navigation method's performance in the following four outdoor test scenarios that differ from the training environments. At least 10 trials are conducted in each scenario.

Scenario 1 - Contains trees, pliable and non-pliable plants and bushes (Fig. 1).

Scenario 2 - Contains hanging leaves and trees where the sensors undergo partial and complete occlusions (Fig. 6(a)).

Scenario 3 - Contains open ground and pliable tall grass regions (Fig. 6(b)).

Scenario 4 - Contains trees and vegetation under low light conditions (Fig. 6(c)).

E. Analysis and Discussion

The qualitative and quantitative results are presented in Fig. 5, 6, and Table I respectively. Qualitative results in Fig. 5 demonstrate that our method is able to differentiate trajectories with high and low success probabilities even during camera occlusions and low light conditions (see Fig. 5(c), (d)). GrASPE also provides accurate success prediction in cluttered

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

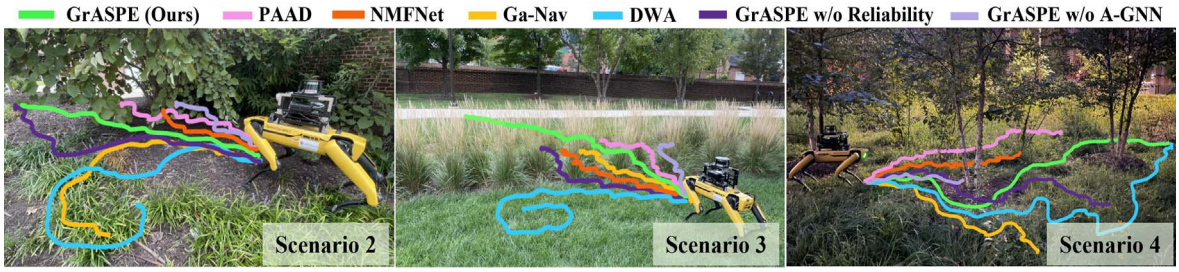

Fig. 6. Spot robot navigation in challenging outdoor scenarios using GrASPE (ours), PAAD [6], NMFNet [8], GA-Nav [39], DWA [9], GrASPE w/o reliability, and GrASPE w/o A-GNN. The test environments are different from the training scenarios. GrASPE is able to successfully continue navigation in varying environmental conditions.

regions in Fig. 5(a), (b) where point cloud encounters heavy distortions.

We further observe that GrASPE outperforms all the comparison methods in terms of success rate, and AU-ROC score when navigating in highly unstructured outdoor regions. This is primarily due to the improved trajectory success probability predictions generated by our algorithm using *reliability-aware* sensor fusion. PAAD and NMFNet demonstrate comparative perception and navigation performance in moderately complex environments since they are designed for multi-modal fusion. Especially, PAAD is capable of predicting potential navigation failures under slightly distorted sensor observations which result in reasonably high AU-ROC scores for Scenarios 1 and 3. However, PAAD fails to continue with successful navigation when the robot enters to extremely challenging regions such as cluttered vegetation such as tall grass in Scenario 3, and hanging leaves in Scenario 2. NMFNet shows severe performance degradation in all scenarios due to the lack of potential failure prediction capabilities.

2D LiDAR based methods such as DWA perform well in less cluttered regions such as scenario 4. However, thin, cluttered, and pliable vegetation are also detected as obstacles by DWA. Hence, it could not succeed in Scenarios 1 and 3 even once during testing. In scenario 4, DWA generates longer trajectories to avoid thin pliable grass regions while GrASPE traverses through them to generate relatively shorter trajectories. However, we observe that GrAPSE sometimes gets stuck in Scenario 4 and rotates in place until it finds a traversable region. Further, PAAD, NMFNet, and GA-Nav result in incomplete trajectories mostly under visually challenging scenarios 2 and 4, which leads to normalized trajectory length values of less than 1. We report them since they measure the progress toward the goal.

GA-Nav uses a point cloud based elevation map along with RGB images to estimate terrain traversability. We observe that the distorted point clouds generate misleading estimations of terrain elevation. This results in a significantly lower success rate for GA-Nav in scenario 2. However, GrASPE-based navigation is robust even when the robot's camera is 100% occluded, and the LiDAR is partially occluded. It uses the partially reliable LiDAR and robot's odometry to avoid collisions and navigate to its goal.

Failure cases: There are trials in scenarios 1 and 2 where both the camera and LiDAR are highly occluded and unreliable. In such cases, GrASPE-based navigation does not have any feedback to avoid obstacles and non-traversable regions and could cause collisions or get stuck. Particularly, when Eq. 8 omits the admissible velocity space V_a considering unreliable point cloud scenarios, misclassification from the GrASPE model could lead to collisions. We note, however, that our method still significantly outperforms the most recent and related works such as PAAD and NMFNet.

TABLE I
NAVIGATION PERFORMANCE OF OUR METHOD COMPARED TO OTHER METHODS ON VARIOUS METRICS.

Metrics	Method	Scn. 1	Scn. 2	Scn. 3	Scn. 4
Success Rate (%)	DWA [9]	0	20	0	70
	GA-Nav [39]	10	10	30	10
	NMFNet [8]	30	10	40	20
	PAAD [6]	40	40	50	40
	GrASPE without A-GNN	10	20	10	0
	GrASPE without Reliability	50	40	40	60
	GrASPE(ours)	**70**	**70**	**80**	**80**
Norm. Traj. Length	DWA [9]	0.38	0.56	0.33	1.32
	GA-Nav [39]	0.35	0.59	0.52	0.49
	NMFNet [8]	0.69	0.58	0.53	0.72
	PAAD [6]	0.81	0.95	0.76	0.93
	GrASPE without A-GNN	0.42	0.63	0.38	0.43
	GrASPE without Reliability	0.89	1.25	0.96	0.77
	GrASPE(ours)	1.11	1.09	1.15	1.23
AU-ROC Score	DWA [9]	-	-	-	-
	GA-Nav [39]	0.578	0.531	0.544	0.529
	NMFNet [8]	0.625	0.596	0.652	0.584
	PAAD [6]	0.756	0.652	0.713	0.623
	GrASPE without A-GNN	0.557	0.548	0.537	0.538
	GrASPE without Reliability	0.772	0.674	0.743	0.588
	GrASPE(ours)	**0.803**	**0.725**	**0.821**	**0.694**

TABLE II
INFERENCE RATES OF THE METHODS IN COMPARISON.

Method	Inference Rate (Hz)
GA-Nav [39]	14.832
NMFNet [8]	10.725
PAAD [6]	13.578
GrASPE without A-GNN	19.415
GrASPE without Reliability	9.167
GrASPE (ours)	8.873

Benefits of Attention-based GNN (A-GNN) : In this ablation, we feed f_{vec} to a set of 1D convolution layers, instead of using the graph representation and our A-GNN. We observe this GrASPE w/o A-GNN pipeline cannot outperform our GrASPE system qualitatively or quantitatively after training and testing with our dataset. This is primarily due to the strong feature correlation learning capabilities of our A-GNN.

Ablation Study on Reliability Measures: We compare the performance of GrASPE without the reliability terms in equation 6. GrASPE w/o reliability leads to erroneous predictions when one or more sensory inputs are heavily distorted. Hence, even though GrASPE w/o reliability can navigate reasonably well compared to other methods, it still could not complete the navigation tasks consistently. This results in a lower success rate and higher AU-ROC score.

Inference Rate: Table II compares GrASPE's inference rate with the other methods. GrASPE demonstrates a low inference rate due to the computationally heavy GNN backbone. However, the design choices in our GNN (Section IV-E) such as dropout and GAT layers (instead of heavy deep GCN backbones) help keep the rates high enough for navigation.

VI. Conclusions, Limitations and Future Work

We present a novel multi-modal fusion algorithm to navigate a legged robot in unstructured outdoor environments where the sensors experience distortions. We utilize a graph attention-based prediction model along with a sensor reliability estimator to obtain a given trajectory's navigation success probabilities. This prediction model is combined with a local planner to generate navigation actions. We validate our method's performance in challenging environments and compare it with the other methods qualitatively and quantitatively. Our algorithm has a few limitations. We assume non-holonomic robot dynamics and further investigation is required to extend our fusion strategy for the holonomic cases. Our method could cause collisions in extreme scenarios where all the perception sensors become unreliable. Adding a haptic sensor modality to navigate more cautiously could help reduce the impact of such failures.

References

[1] R. Limosani, R. Esposito, A. Manzi, G. Teti, F. Cavallo, and P. Dario, "Robotic delivery service in combined outdoor–indoor environments: technical analysis and user evaluation," *Robotics and Autonomous Systems*, vol. 103, pp. 56–67, 2018.

[2] J. J. Roldán, J. del Cerro, D. Garzón-Ramos, P. Garcia-Aunon, M. Garzón, J. De León, and A. Barrientos, "Robots in agriculture: State of art and practical experiences," *Service robots*, pp. 67–90, 2018.

[3] Y. Gu, J. Strader, N. Ohi, S. Harper, K. Lassak, C. Yang, L. Kogan, B. Hu, M. Gramlich, R. Kavi, *et al.*, "Robot foraging: Autonomous sample return in a large outdoor environment," *IEEE Robotics & Automation Magazine*, vol. 25, no. 3, pp. 93–101, 2018.

[4] B. Choi, W. Lee, G. Park, Y. Lee, J. Min, and S. Hong, "Development and control of a military rescue robot for casualty extraction task," *Journal of Field Robotics*, vol. 36, no. 4, pp. 656–676, 2019.

[5] M. Aladem, S. Baek, and S. A. Rawashdeh, "Evaluation of image enhancement techniques for vision-based navigation under low illumination," *Journal of Robotics*, vol. 2019, 2019.

[6] T. Ji, A. N. Sivakumar, G. Chowdhary, and K. Driggs-Campbell, "Proactive anomaly detection for robot navigation with multi-sensor fusion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4975–4982, 2022.

[7] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2020, pp. 11 682–11 692.

[8] A. Nguyen, N. Nguyen, K. Tran, E. Tjiputra, and Q. D. Tran, "Autonomous navigation in complex environments with deep multimodal fusion network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5824–5830.

[9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[10] A. J. Sathyamoorthy, K. Weerakoon, T. Guan, J. Liang, and D. Manocha, "Terrapn: Unstructured terrain navigation using online self-supervised learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7197–7204.

[11] K. Weerakoon, A. J. Sathyamoorthy, U. Patel, and D. Manocha, "Terp: Reliable planning in uneven outdoor environments using deep reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9447–9453.

[12] C. Debeunne and D. Vivet, "A review of visual-lidar fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, 2020.

[13] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[14] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[15] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, and L. Carlone, "Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9272–9279.

[16] K. Katyal, K. Popek, C. Paxton, P. Burlina, and G. D. Hager, "Uncertainty-aware occupancy map prediction using generative networks for robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5453–5459.

[17] F. Liu, X. Li, S. Yuan, and W. Lan, "Slip-aware motion estimation for off-road mobile robots via multi-innovation unscented kalman filter," *IEEE Access*, vol. 8, pp. 43 482–43 496, 2020.

[18] Y. Su, T. Wang, S. Shao, C. Yao, and Z. Wang, "Gr-loam: Lidar-based sensor fusion slam for ground robots on complex terrain," *Robotics and Autonomous Systems*, vol. 140, p. 103759, 2021.

[19] S. S. Mansouri, C. Kanellakis, D. Kominiak, and G. Nikolakopoulos, "Deploying mavs for autonomous navigation in dark underground mine environments," *Robotics and Autonomous Systems*, vol. 126, p. 103472, 2020.

[20] D. C. Guastella and G. Muscato, "Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review," *Sensors*, vol. 21, no. 1, p. 73, 2020.

[21] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. Velasquez, V. A. Higuti, J. Rogers, H. Tran, and G. Chowdhary, "Wayfast: Navigation with predictive traversability in the field," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 651–10 658, 2022.

[22] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus, "Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 568–575.

[23] T. Ji, S. T. Vuppala, G. Chowdhary, and K. Driggs-Campbell, "Multi-modal anomaly detection for unstructured and uncertain environments," in *Conference on Robot Learning*. PMLR, 2021, pp. 1443–1455.

[24] D. Park, H. Kim, and C. C. Kemp, "Multimodal anomaly detection for assistive robots," *Autonomous Robots*, vol. 43, no. 3, pp. 611–629, 2019.

[25] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[26] K. Jin, H. Wang, C. Liu, Y. Zhai, and L. Tang, "Graph neural network based relation learning for abnormal perception information detection in self-driving scenarios," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8943–8949.

[27] L. Wellhausen, R. Ranftl, and M. Hutter, "Safe robot navigation via multi-modal anomaly detection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1326–1333, 2020.

[28] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[29] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based graph neural network for semi-supervised learning," *arXiv preprint arXiv:1803.03735*, 2018.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[31] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[32] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 652–660.

[33] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2008.

[34] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[35] A. Mojsilovic, J. Gomes, and B. E. Rogowitz, "Isee: Perceptual features for image library navigation," in *Human Vision and Electronic Imaging VII*, vol. 4662. SPIE, 2002, pp. 266–277.

[36] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European conference on computer vision (ECCV)*. Springer, 2006, pp. 430–443.

[37] J. Gu, Z. Jia, T. Cai, X. Song, and A. Mahmood, "Dynamic correlation adjacency-matrix-based graph neural networks for traffic flow prediction," *Sensors*, vol. 23, no. 6, p. 2897, 2023.

[38] K. Weerakoon, A. Jagan Sathyamoorthy, J. Liang, T. Guan, U. Patel, and D. Manocha, "GrASPE: Graph based Multimodal Fusion for Robot Navigation in Unstructured Outdoor Environments," *arXiv e-prints*, p. arXiv:2209.05722, Sept. 2022.

[39] 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.

[40] 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)*. IEEE, 2019.

[41] 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.