

GSAT: Geometric Traversability Estimation using Self-supervised Learning with Anomaly Detection for Diverse Terrains

Dongjin Cho¹, Miryeong Park¹, Juhui Lee¹, Geonmo Yang¹, and Younggun Cho^{1†}

Abstract—Safe autonomous navigation requires reliable estimation of environmental traversability. Traditional methods have relied on semantic or geometry-based approaches with human-defined thresholds, but these methods often yield unreliable predictions due to the inherent subjectivity of human supervision. While self-supervised approaches enable robots to learn from their own experience, they still face a fundamental challenge: the positive-only learning problem. To address these limitations, recent studies have employed Positive-Unlabeled (PU) learning, where the core challenge is identifying positive samples without explicit negative supervision. In this work, we propose GSAT, which addresses these limitations by constructing a positive hypersphere in latent space to classify traversable regions through anomaly detection without requiring additional prototypes (e.g., unlabeled or negative). Furthermore, our approach employs joint learning of anomaly classification and traversability prediction to more efficiently utilize robot experience. We comprehensively evaluate the proposed framework through ablation studies, validation on heterogeneous real-world robotic platforms, and autonomous navigation demonstrations in simulation environments. Our method is available at <https://sparolab.github.io/research/gsat/>.

I. INTRODUCTION

Traversability estimation is an essential ability to determine whether a robot can safely traverse a given terrain for autonomous navigation in unstructured environments.

Traditionally, this task has followed two main paradigms: semantic methods [1, 2] employ predefined classes (e.g., road, rock) to determine traversability categories (e.g., safe, risky), whereas geometric methods [3, 4] extract terrain features (e.g., slope, roughness) from elevation maps. However, both methods depend on human-defined thresholds, which often yield inaccurate predictions, leading to unreliable navigation.

To address these limitations, many studies [5–14] utilize self-supervised learning, enabling robots to acquire traversability knowledge from their own experience without human supervision. In these approaches, supervision signals are automatically generated from traversal experiences. Such signals allow robots to learn models that capture platform-specific traversal patterns, thereby reducing dependence on manually designed heuristics.

This work was supported by National Research Foundation of Korea (NRF) grant (RS-2025-02217000) funded by the Korea government (MSIT) & This work was supported by National Research Foundation of Korea (NRF) grant (RS-2025-24803365) funded by the Korea government (MSIT) & This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00448/RS-2022-II220448) and Inha University.

¹Dongjin Cho, ¹Miryeong Park, ¹Juhui Lee, ¹Geonmo Yang and ^{1†}Younggun Cho are with the Electrical and Computer Engineering and INHA Future Mobility IPCC, Inha University, Incheon, South Korea. [e-mail: d22g66, bark9757, dlwngml6635, ygm7422]@inha.edu, yg.cho@inha.ac.kr

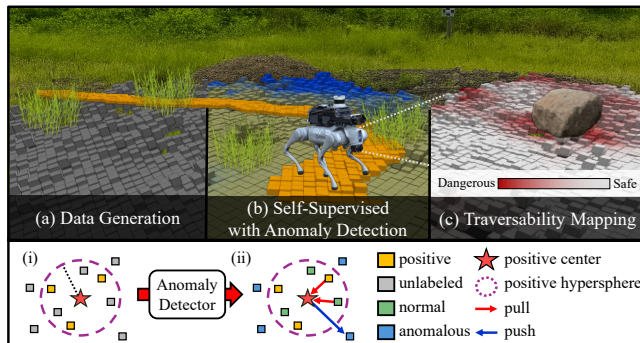


Fig. 1. Overall process of the proposed GSAT framework: (a) Automated data generation using positive and unlabeled samples to enable self-supervised learning, corresponding to the initial latent space (i); (b) Self-supervised anomaly detection and hypersphere refinement to optimize the decision boundary within the latent space (ii); (c) Final traversability mapping for autonomous navigation.

Despite this progress, self-supervised methods face the positive-only learning problem. The absence of contrastive samples (i.e., negative samples) causes the positive hypersphere representation to become unstable, leading to feature collapse, such as a trivial solution. Therefore, robots struggle to distinguish between normal samples (similar to experienced areas) and anomalous samples (unexperienced areas).

Existing methods [10–12] address this problem by constructing prototypes from positive and unlabeled sets via Positive-Unlabeled (PU) learning, but the unlabeled data inherently contain normal samples. Consequently, the positive prototype becomes sensitive to prototype configurations (e.g., their number), resulting in inconsistent classification.

Recently, other methods [13, 14] utilize visual foundation models such as Segment Anything (SAM) [15] to mask regions similar to positive samples, treating the remaining areas as explicit negative samples. However, unlike semantic data, geometric data lacks corresponding foundation models, limiting the applicability of such foundation-based approaches.

To address this problem, we present **GSAT** (Geometric traversability estimation using Self-supervised learning with Anomaly detection for diverse Terrain), as illustrated in Fig. 1. The proposed framework introduces four key contributions:

- 1) **Experience-aware Anomaly Detection:** We propose a self-supervised anomaly detection framework using a single positive hypersphere. Unlike prototype-based PU methods, our approach defines a positive hypersphere as a decision boundary, enabling the identification of anomalies in unlabeled data. This formulation facilitates anomaly-aware metric learning, addressing the inherent limitations of positive-only learning.

- 2) **Joint Traversability Learning Framework:** We introduce a framework that jointly learns anomaly detection and traversability prediction. This joint optimization enables efficient terrain assessment through shared representations.
- 3) **Geometric Data Augmentation for Diversity:** Human-operated data often lacks diversity due to its consistent and safe operations. To address this, we introduce geometric augmentations. These generate diverse experiences, allowing the robot to adapt to various unseen regions.
- 4) **Comprehensive Evaluation:** We evaluate our model across multiple datasets for anomaly classification. Extensive experiments on downstream tasks demonstrate the effectiveness of our approach.

II. RELATED WORKS

A. Traversability Estimation with Human Supervision

Traditional traversability estimation relies on either semantic or geometric features. Semantic approaches [1, 2] classify terrain into predefined categories (e.g., road, bush, rock) to determine traversability levels (e.g., safe, risky). Geometric methods [3, 4] extract terrain characteristics (e.g., slope, roughness, step height) from elevation maps generated through various probabilistic frameworks. However, these approaches depend on manually defined classes or constraints, often leading to inaccurate predictions as they fail to capture vehicle-specific traversal experiences.

B. Self-Supervised Traversability Estimation

To mitigate the reliance on human supervision, recent self-supervised approaches project driving trajectories into network features to generate automated supervision signals. Various sensor modalities have been explored for this task, including RGB [6, 7], RGB-D [5], and LiDAR [8, 16], with some methods combining modalities to leverage both geometric and semantic information [17].

However, these methods rely primarily on reconstruction-based uncertainty derived solely from positive data. Due to the absence of contrastive samples during training, their generalization performance becomes highly sensitive to threshold configurations, often failing to establish a robust decision boundary between normal and anomalous regions.

C. Traversability from Anomalies

To address the lack of negative samples, several works adopt the PU learning framework, treating the unlabeled set as contrastive samples [10–12]. These approaches often rely on unlabeled prototypes to distinguish between regions. However, since unlabeled sets inherently contain positive-like samples, the decision boundary becomes highly sensitive to prototype configurations, leading to inconsistent classification.

Alternative approaches [13, 14] incorporate visual foundation models, such as SAM, to explicitly generate negative samples for refining positive prototypes. Although effective for semantic scene understanding, these methods face significant challenges in geometric domains due to the absence of corresponding foundation models.

In contrast, our method focuses solely on constructing a reliable positive prototype to define a decision boundary. This formulation enables the separation of normal and anomalous samples within unlabeled data without requiring additional prototypes or relying on foundation models.

III. METHOD

In this section, we describe how geometric information can be utilized for traversability estimation through self-supervised learning. First, we introduce an automated data generation pipeline (Sec. III-A). We then describe the traversability network architecture (Sec. III-B), followed by our proposed experience-aware anomaly detection methodology and corresponding loss functions for constructing a positive hypersphere through joint learning (Sec. III-C). These core components are illustrated in Fig. 2. Finally, we present a targeted geometric augmentation strategy to address the limited diversity of positive samples (Sec. III-D).

A. Automated Data Generation

As the initial step, training data is automatically generated by aligning SLAM-derived robot trajectories with the point cloud \mathcal{P}^t captured at discrete time intervals. Specifically, the system constructs this supervision by associating sampled trajectory points with traversability scores. For each time step t , the supervision set \mathcal{B}^t is defined as:

$$\mathcal{B}^t = \{(\mathbf{b}^s, \tau^s) \mid s \in [t, t + n]\}, \quad (1)$$

where $\mathbf{b}^s = (x^s, y^s)$ represents the robot’s position coordinates, τ^s denotes the traversability score at sampling step s , and n denotes the window size. Traversability scores are derived from velocity tracking performance as follows [7]:

$$v_{\text{error}} = \frac{1}{2} \left((v_x - v_{x^*})^2 + (v_y - v_{y^*})^2 \right), \quad (2)$$

$$\tau = \sigma(-\eta(v_{\text{error}} - v_{\text{th}})), \quad (3)$$

where (v_x, v_y) and (v_{x^*}, v_{y^*}) are the actual and commanded velocities, respectively. The parameter η controls the sharpness of the sigmoid function $\sigma(\cdot)$, and v_{th} denotes the midpoint where the traversability score is assigned as 0.5.

1) *BEV Representation:* Direct processing of 3D point clouds limits performance on resource-constrained robotic platforms [18]. Therefore, we adopt a BEV representation to achieve computational efficiency for real-time applications. To align \mathcal{P}^t and \mathcal{B}^t , both are projected onto their respective 2D grids of size $H_\rho \times W_\rho$ with resolution ρ .

Pillar Voxelization: \mathcal{P}^t is voxelized into a 4D tensor $\mathcal{V} \in \mathbb{R}^{H_\rho \times W_\rho \times M \times 7}$, where each grid cell (i, j) stores a vertical pillar $\mathbf{V}_{i,j} = \{\mathbf{v}_{i,j}^{(m)}\}_{m=1}^M \in \mathbb{R}^{M \times 7}$. M denotes the maximum number of points per pillar, with the remaining entries zero-padded. Each per-point feature $\mathbf{v}_{i,j}^{(m)} \in \mathbb{R}^7$ is defined as:

$$\mathbf{v}_{i,j}^{(m)} = [x_l, y_l, z, x_c, y_c, z_c, \sigma_z]^\top, \quad (4)$$

where z is the absolute height of the point, (x_l, y_l) denote the local offsets from the pillar center, (x_c, y_c, z_c) are the offsets relative to the mean of all points in the pillar, and σ_z is the standard deviation of z within the pillar.

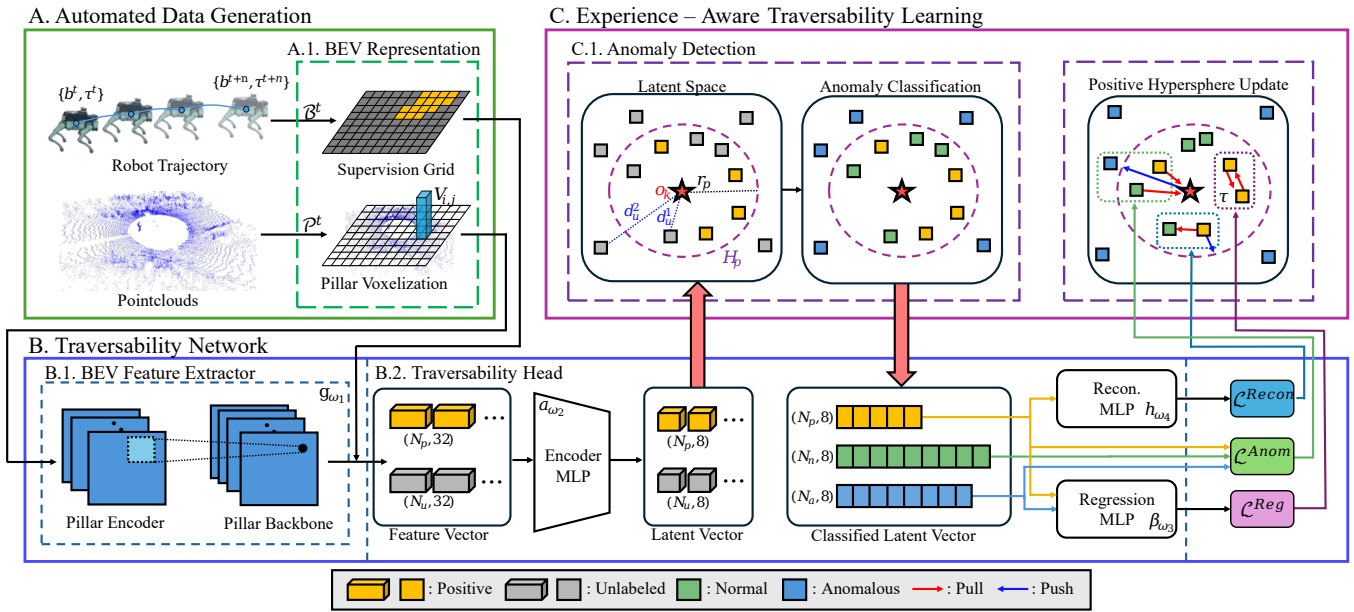


Fig. 2. **Overview of the proposed GSAT framework.** The framework consists of (A) automated data generation leveraging robot traversal supervision, and (B) a traversability network that extracts BEV features to produce latent representations. The core innovation lies in (C) experience-aware traversability learning, where self-supervised anomaly detection in the latent space partitions unlabeled data into normal and anomalous samples, enabling joint traversability estimation and positive hypersphere refinement through the proposed loss formulations.

Supervision Grid: The supervision set \mathcal{B}^t is represented as a 2D grid $\mathbf{G} \in \mathbb{R}^{H_\rho \times W_\rho}$. Each cell $\mathbf{G}_{i,j}$ containing a robot position \mathbf{b}^s stores the corresponding traversability score τ^s . When multiple positions occupy the same cell, the mean score is assigned. Unvisited cells are set to -1 .

B. Traversability Network

1) *BEV Feature Extractor:* We design our feature extractor based on the PointPillars [18] architecture. Pillar features $\mathbf{V}_{i,j}$ are encoded through a 1D-CNN followed by a 2D-CNN backbone, producing a spatial feature vector $\mathbf{q}_{i,j} \in \mathbb{R}^{32}$ for each grid cell (i, j) . The BEV feature map \mathbf{Q} is composed of per-cell feature vectors $\mathbf{q}_{i,j}$, which are computed as:

$$\mathbf{q}_{i,j} = g_{\omega_1}(\mathbf{V}_{i,j}), \quad (5)$$

where $g_{\omega_1}(\cdot)$ denotes the BEV feature extraction module with learnable parameters ω_1 . The feature vectors are separated into positive and unlabeled sets based on the supervision grid \mathbf{G} :

$$\begin{aligned} \mathcal{Q}_p &= \{\mathbf{q}_{i,j} \mid \mathbf{G}_{i,j} \neq -1\}, \\ \mathcal{Q}_u &= \{\mathbf{q}_{i,j} \mid \mathbf{G}_{i,j} = -1\}, \end{aligned} \quad (6)$$

where the subscripts p and u indicate the positive and unlabeled sets, respectively.

2) *Traversability Head:* The Traversability Head processes the BEV feature set $\mathcal{Q} = \mathcal{Q}_p \cup \mathcal{Q}_u$ through three MLPs:

$$\mathcal{Z} = \alpha_{\omega_2}(\mathcal{Q}), \quad \mathcal{T} = \beta_{\omega_3}(\mathcal{Z}), \quad \mathcal{U} = h_{\omega_4}(\mathcal{Z}), \quad (7)$$

where $\alpha_{\omega_2}(\cdot)$ denotes the encoder, $\beta_{\omega_3}(\cdot)$ the regression head, and $h_{\omega_4}(\cdot)$ the reconstruction head. Specifically, for $N = N_p + N_u$ total samples where $N_p = |\mathcal{Q}_p|$ and $N_u = |\mathcal{Q}_u|$, $\mathcal{Z} \in \mathbb{R}^{N \times 8}$ denotes the latent representations, $\mathcal{T} \in \mathbb{R}^{N \times 1}$ the predicted traversability scores, and $\mathcal{U} \in \mathbb{R}^{N \times 32}$ the reconstructed features. The encoder and reconstruction heads consist

of two layers each, while the regression head has a single layer. All hidden layers employ batch normalization, dropout, and ReLU, whereas the final layer of the regression head applies a sigmoid activation.

C. Experience-Aware Traversability Learning

1) *Anomaly Detection:* Unlike methods [10–14] that rely on unlabeled or negative prototypes, we employ a metric-based approach using positive hypersphere boundaries to separate normal and anomalous samples in unlabeled data.

The positive hypersphere H_p is defined by its center $\mathbf{o}_k \in \mathbb{R}^8$ and radius r_p . The center \mathbf{o}_k is defined as the mean feature vector of positive latent features $\mathbf{z}_p^{(k)}$, computed over the full batch and updated every k epochs for stable convergence:

$$\mathbf{o}_k = \frac{1}{N_p} \sum_{\mathcal{Z}_p} \mathbf{z}_p^{(k)}. \quad (8)$$

The radius r_p follows an exponential moving average:

$$r_p^+ = \varepsilon r_p^- + (1 - \varepsilon) \bar{d}_p, \quad (9)$$

where $d_p = \|\mathbf{z}_p - \mathbf{o}_k\|$ is the distance between each positive latent vector $\mathbf{z}_p \in \mathcal{Z}_p$ and the center, \bar{d}_p is their mean, and ε is the momentum parameter. Superscripts $-$ and $+$ denote pre- and post-update values at each epoch. Using H_p as a boundary, unlabeled samples $\mathbf{z}_u \in \mathcal{Z}_u$ are split into normal \mathcal{Z}_n and anomalous \mathcal{Z}_a sets:

$$\begin{aligned} \mathcal{Z}_n &= \{\mathbf{z}_u \in \mathcal{Z}_u \mid d_u \leq r_p^+\}, \\ \mathcal{Z}_a &= \{\mathbf{z}_u \in \mathcal{Z}_u \mid d_u > r_p^+\}, \end{aligned} \quad (10)$$

where $d_u = \|\mathbf{z}_u - \mathbf{o}_k\|$ represents the distance from each unlabeled sample to the hypersphere center, with $N_n = |\mathcal{Z}_n|$ and $N_a = |\mathcal{Z}_a|$ denoting the sizes of each set, satisfying $N_u = N_n + N_a$.

2) *End-to-End Training Objective*: To address the positive-only problem in traversability estimation, we jointly optimize traversability prediction and anomaly-aware representation learning. Specifically, the overall training objective consists of the loss functions detailed below.

Anomaly Loss. We focus on optimizing a positive prototype without negative or unlabeled ones. Therefore, we adapt an anomaly loss inspired by Deep-SAD [19], which refines the positive center by explicitly pushing anomalous samples away from it. While Deep-SAD requires pre-labeled normal and anomalous data, our approach leverages automated anomaly detection (Sec. III-C.1) to classify unlabeled samples, enabling the application of this loss formulation. The anomaly loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{Anom}} = & \underbrace{\frac{1}{N_p} \sum_{\mathcal{Z}_p} \|\mathbf{z}_p - \mathbf{o}_k\|^2}_{\text{positive}} + \underbrace{\frac{1}{N_n} \sum_{\mathcal{Z}_n} \|\mathbf{z}_n - \mathbf{o}_k\|^2}_{\text{normal}} \\ & + \underbrace{\frac{1}{N_a} \sum_{\mathcal{Z}_a} (\|\mathbf{z}_a - \mathbf{o}_k\|^2 + \zeta)^{-1}}_{\text{anomalous}}, \end{aligned} \quad (11)$$

where $\mathbf{z}_p \in \mathcal{Z}_p$, $\mathbf{z}_n \in \mathcal{Z}_n$, and $\mathbf{z}_a \in \mathcal{Z}_a$ are latent vectors from the positive, normal, and anomalous sets, respectively. The first two terms pull positive and normal samples toward the hypersphere center, while the third term pushes anomalous samples away through inverse distance. ζ is a small constant that prevents numerical divergence. This formulation constructs a positive boundary that enables anomaly detection without requiring auxiliary prototypes.

Reconstruction Loss. While the anomaly loss effectively learns hypersphere boundaries, pulling positive features toward the center without any margin may cause the encoder to overfit to specific positive patterns. To mitigate this, we employ a reconstruction loss formulated as the Mean Squared Error (MSE) to preserve general representations:

$$\mathcal{L}_{\text{Recon}} = \frac{1}{N_p} \sum_{\mathcal{U}_p} \|\mathbf{u}_p - \mathbf{q}_p\|^2, \quad (12)$$

where $\mathbf{u}_p \in \mathcal{U}_p$ represents the reconstructed feature decoded from the latent vector \mathbf{z}_p , and \mathbf{q}_p denotes the corresponding original positive input. As a regularizer, this loss complements the anomaly loss by preventing feature collapse and ensuring robust representations for anomaly detection.

Regression Loss. Our regression loss leverages experience-aware joint learning to achieve efficient traversability estimation. Existing methods [7] rely on reconstruction error as a heuristic anomaly indicator with manually defined thresholds. Unlike these methods, our approach utilizes the classified samples, eliminating the need for such thresholds. We formulate the regression objective using the MSE as follows:

$$\mathcal{L}_{\text{Reg}} = \frac{1}{N_p} \sum_{\mathcal{T}_p} \|\mathbf{t}_p - \tau\|^2 + \frac{1}{N_a} \sum_{\mathcal{T}_a} \|\mathbf{t}_a - 0\|^2, \quad (13)$$

where $\mathbf{t}_p \in \mathcal{T}_p$ and $\mathbf{t}_a \in \mathcal{T}_a$ denote the traversability predictions for positive and anomalous samples, respectively. The first term

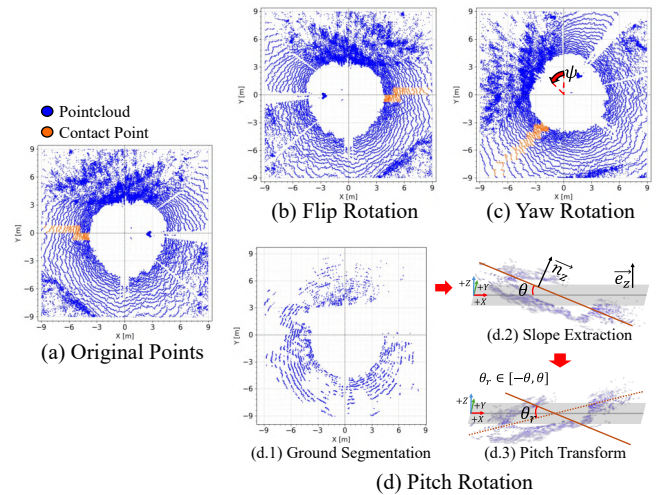


Fig. 3. Point augmentation methods demonstrated on RELIS-3D dataset. (a) Sample point clouds. (b) Flipping augmentation reflecting points across the yz -plane. (c) Yaw rotation augmentation around the z -axis. (d) Pitch augmentation process: (d.1) RANSAC-based ground segmentation, (d.2) slope angle extraction, (d.3) pitch transformation to simulate terrain inclinations.

supervises \mathbf{t}_p toward the target score τ , while the second term constrains \mathbf{t}_a toward zero to reflect risk regions. This joint formulation enables efficient traversability estimation.

Total Loss. The final objective function combines all three loss components as follows:

$$\mathcal{L}_{\text{Total}} = \lambda_1 \mathcal{L}_{\text{Anom}} + \lambda_2 \mathcal{L}_{\text{Recon}} + \lambda_3 \mathcal{L}_{\text{Reg}}, \quad (14)$$

where λ_1 , λ_2 , and λ_3 weight the respective loss terms.

D. Strategy for Mitigating Limited Data Diversity

While our anomaly detection framework expands the positive hypersphere using normal samples from unlabeled data, the positive samples themselves often exhibit limited diversity due to consistent human operational patterns. Therefore, we apply targeted geometric augmentations to mitigate orientation and terrain biases inherent in positive traversal data. We employ flipping and yaw rotation to address directional bias, and pitch rotation based on ground estimation to increase slope diversity. The detailed augmentation strategies are as follows:

Flipping. As shown in Fig. 3(b), we flip points across the yz -plane using a diagonal matrix $\mathbf{S}_x = \text{diag}(-1, 1, 1)$. The transformation $f_x(\mathbf{p}) = \mathbf{S}_x \mathbf{p}$ is applied to each point $\mathbf{p} = [x, y, z]^\top$, creating symmetric trajectory patterns and mitigating distributional bias in unidirectional sampling.

Yaw Rotation. As shown in Fig. 3(c), we apply random yaw rotations about the z -axis using a rotation matrix $\mathbf{R}_z(\psi)$, where $\psi \in [-90^\circ, 90^\circ]$. This transformation $f_\psi(\mathbf{p}) = \mathbf{R}_z(\psi) \mathbf{p}$ enhances angular diversity.

Pitch Rotation. As shown in Fig. 3(d), we simulate terrain slope variations through pitch rotation based on RANSAC ground segmentation. The ground slope angle $\theta = \arccos(\mathbf{n}_z \cdot \mathbf{e}_z)$ determines the rotation range, where \mathbf{n}_z is the estimated ground normal and $\mathbf{e}_z = [0, 0, 1]^\top$ is the vertical unit vector. Assuming that any slope up to θ remains traversable, a random angle $\theta_r \in [-\theta, \theta]$ is then applied as $f_g(\theta_r, \mathbf{p}) = \mathbf{R}_y(\theta_r) \mathbf{p}$, generating diverse slope conditions.

IV. EXPERIMENTS ON ANOMALY CLASSIFICATION

We conducted comprehensive ablation studies to analyze the impact of two design choices within our anomaly classification: the treatment of unlabeled data in the anomaly loss and the selection of augmentation strategies. Following the experimental setup, including datasets and evaluation metrics (Sec. IV-A), we quantitatively analyze the anomaly loss formulation (Sec. IV-B) and evaluate each augmentation component through both quantitative and qualitative assessments (Sec. IV-C).

A. Experimental Setup

1) *Datasets*: We conducted experiments on two public datasets: RELLIS-3D [20], which features consistent positive sample patterns, and DITER++ [21], which provides diverse spatial environments. For both datasets, we sampled approximately 50 robot poses per training to define the positive samples. **RELLIS-3D**: This dataset was collected with an OS1-64 LiDAR sensor in off-road environments and provides point-wise semantic annotations. We used sequences 00 and 03 for training and sequence 01 for evaluation. For anomaly evaluation, we redefined the provided semantic labels into traversability categories. Specifically, common terrain types (e.g., *ground, grass, mud, and puddles*) were grouped as normal, whereas all other labels (e.g., *trees, rocks*) were treated as anomalous. **DITER++**: This dataset was collected with an OS1-32 LiDAR sensor in diverse urban and park environments. Training was conducted on the park-night sequence and evaluation on the park-day sequence. For evaluation, since semantic annotations were not provided, we manually labeled non-ground and obstacle regions as anomalous, while areas consistent with prior experience were labeled as normal.

2) *Data Processing and Evaluation Metrics*: We utilized LiDAR point clouds, extracting supervision signals from SLAM-based robot poses. Since command velocity information was unavailable, we set the target traversability score to $\tau = 1$. To evaluate anomaly classification, point-wise semantic labels were projected onto a BEV grid of 12×12 m with a 0.15 m resolution. Cells containing any points from predefined anomalous classes were labeled as anomalous, while cells with only normal class points were labeled as normal.

Anomaly classification was performed using a decision boundary determined by the learned hypersphere center and radius, as defined in Eq. (10). We evaluated binary classification performance using Precision, Recall, and F1-score. To focus on regions with reliable measurements, empty cells without sufficient point cloud data were excluded from the evaluation. In Table I and Table II, green and yellow highlights denote the best and second-best results, respectively.

3) *Implementation Details*: The network was trained using the Adam optimizer with a learning rate of 5×10^{-4} , a batch size of 12, and 100 training epochs. The hypersphere parameters were updated every 5 epochs ($k = 5$) using a momentum of $\varepsilon = 0.5$. For numerical stability, we set $\zeta = 10^{-6}$ in the anomaly loss term. For the classification evaluation, the hypersphere center and radius were selected from the best-performing model on the evaluation set. The loss weights were empirically set to $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, and $\lambda_3 = 20.0$. All models were trained

TABLE I
QUANTITATIVE RESULTS OF THE UNLABELED DATA ABLATION STUDY
FOR ANOMALY LOSS FORMULATION

Dataset	Anomaly Loss		Metric		
	Normal	Anomalous	Pre. \uparrow (%)	Rec. \uparrow (%)	F1. \uparrow (%)
RELLIS-3D	–	–	54.75	99.86	69.87
	–	All unlabeled	96.75	52.52	68.27
	–	Anomalous	98.10	59.14	73.59
	Normal	Anomalous	97.11	64.93	77.61
DITER++	–	–	77.34	98.35	85.91
	–	All unlabeled	87.45	57.56	68.50
	–	Anomalous	87.69	75.93	80.61
	Normal	Anomalous	96.69	81.01	88.04

Ours: Normal/Anomalous configuration

and evaluated on an NVIDIA RTX 3090 GPU with 24GB of memory. To ensure reliability, the results were averaged over 3 independent runs with different random seeds.

B. Anomaly Loss Ablation Studies

Table I presents a quantitative evaluation of four configurations on the RELLIS-3D and DITER++ datasets. Specifically, we analyze how the handling of unlabeled data is formulated within the anomaly loss.

w/o Anomaly Loss (–/–): This configuration, equivalent to Deep-SVDD [19] one-class learning, predicted most regions as normal, classifying even non-traversable areas as traversable. Consequently, it yielded the highest recall (99.86%, 98.35%) but the lowest precision (54.75%, 77.34%). Without loss terms for unlabeled data, the model failed to distinguish between normal and anomalous latent features, leading to inflated recall scores that do not reflect true classification performance.

All Unlabeled as Anomalous (–/All unlabeled): This configuration treated all unlabeled data as anomalous, pushing them away from the positive hypersphere center. Both datasets exhibited significantly reduced recall (52.52%, 57.56%) because the model overfitted to positive patterns, creating an overly restrictive hypersphere that missed normal regions in unexplored areas. While this approach achieved high precision (96.75%, 87.45%) by being conservative, it failed to generalize beyond experienced terrain.

Anomalous-only Loss (–/Anomalous only): This configuration constructed the hypersphere using only positive samples while pushing anomalous samples away from the center, without utilizing normal samples. This approach demonstrated an improved balance between precision (98.10%, 87.69%) and recall (59.14%, 75.93%). However, learning the hypersphere based solely on positive samples limited its generalization capability, as normal samples were not guided toward the center.

Proposed Configuration (Normal/Anomalous): Our approach optimized the positive hypersphere by pulling both positive and normal samples toward the center. This incorporation of normal samples significantly improved performance compared to positive-only approaches, achieving the highest F1-scores (77.61%, 88.04%). These results validated our hypothesis that leveraging normal samples enhances hypersphere construction and improves anomaly detection performance.

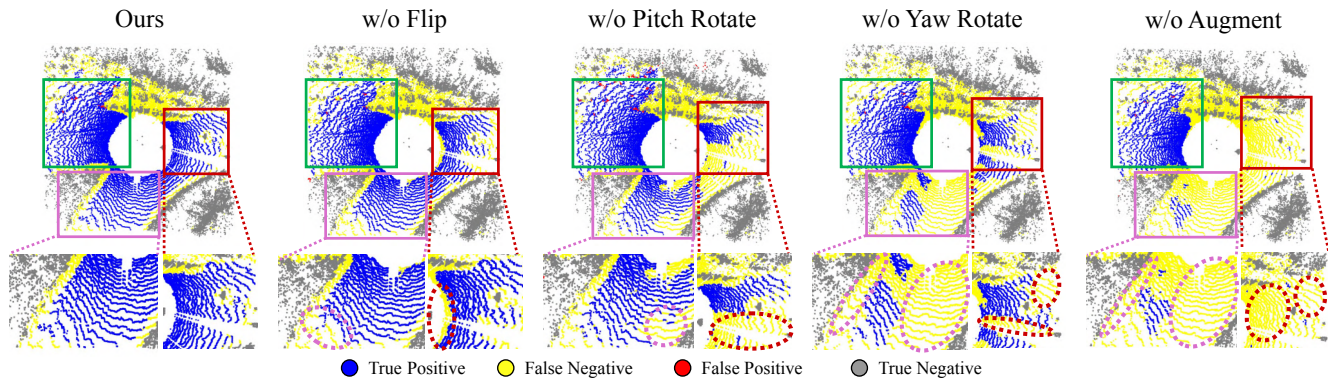


Fig. 4. Qualitative results of the data augmentation ablation study on the RELIS-3D dataset, demonstrating spatial prediction performance across different augmentation configurations.

TABLE II
QUANTITATIVE RESULTS OF DATA AUGMENTATION ON RELIS-3D

Configuration	Pre. \uparrow (%)	Rec. \uparrow (%)	F1. \uparrow (%)
w/o Flipping	96.67	61.32	74.86
w/o Pitch rotation	93.82	61.72	74.16
w/o Yaw rotation	96.78	45.54	61.81
w/o Augmentation	98.28	27.99	43.13
Ours	97.11	64.93	77.61

The performance difference between the datasets reflected ground-truth annotation quality. RELIS-3D employed semantic class-based labeling with inherent inaccuracies (e.g., grass regions were often labeled as traversable despite being non-traversable due to their irregular surfaces). Consequently, achieving high precision on this dataset is a more meaningful indicator of performance than recall. In contrast, DITER++ used more precise manual ground-only labeling, resulting in clearer ground-truth classification and consistently higher overall performance across all configurations.

C. Ablation Studies on Geometric Augmentation

We conducted augmentation ablation studies on the RELIS-3D dataset to evaluate the effectiveness of our geometric augmentation strategies. Unlike DITER++, RELIS-3D exhibits highly consistent trajectory patterns and terrain biases (e.g., directional and slope biases), making it an ideal benchmark for validating the impact of augmentation. Furthermore, to avoid generating misleading positive samples, pitch rotation was selectively applied only to scenes containing a single ground segment with a slope of less than 10° .

Table II presents the quantitative evaluation of each augmentation component. Our augmentation approach achieved the highest F1-score (77.61%), with each technique contributing significantly to performance. Notably, excluding yaw rotation resulted in the most severe performance drop with an F1-score of 61.81%, demonstrating its critical importance for handling directional diversity. Without any augmentation, the model achieved high precision (98.28%) but suffered from extremely low recall (27.99%), indicating overfitting to the limited positive training patterns and weak generalization to unexplored regions.

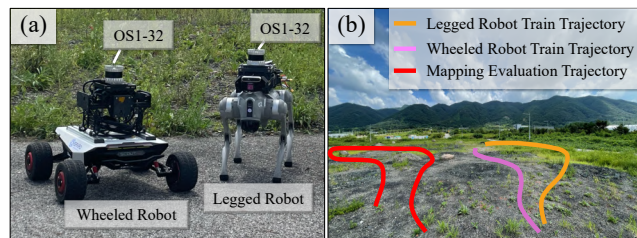


Fig. 5. Experimental setup for traversability mapping evaluation. (a) Robot platforms: SCOUT MINI (left) and Go2 (right). (b) Data collection trajectories in the outdoor environment with training and mapping evaluation paths.

Fig. 4 presents qualitative results showing the effect of each augmentation on prediction performance. Without augmentation, the model performed reliably only in regions similar to training trajectories but failed in unexplored orientations, showing extensive false negatives in the purple and red highlighted regions. The absence of yaw rotation particularly affected predictions in unseen headings, while removing pitch rotation degraded performance on sloped terrain (highlighted in red). Specifically, because the positive training data exhibited a leftward bias, the model severely struggled with rightward direction patterns without flip augmentation. These results demonstrate that geometric augmentation effectively addresses the inherent biases in training data, improving generalization to diverse terrains.

V. EXPERIMENTS ON DOWNSTREAM TASKS

To evaluate our experience-aware traversability estimation, we compared it against representative open-source baselines. Specifically, two methods were selected for comparison: DEM-Trav [22], a rule-based approach that assesses geometric terrain characteristics, and LeSTA [8], a self-supervised method that learns terrain properties. Platform-specific traversability accuracy was evaluated through mapping experiments (Sec. V-A), and its suitability for navigation in unstructured environments was assessed in simulation (Sec. V-B).

A. Traversability Mapping

1) *Experimental Setup*: We evaluated traversability mapping across heterogeneous robots using a custom dataset collected from wheeled and legged robots equipped with OS1-32 LiDAR sensors, as shown in Fig. 5(a). The environment consisted of

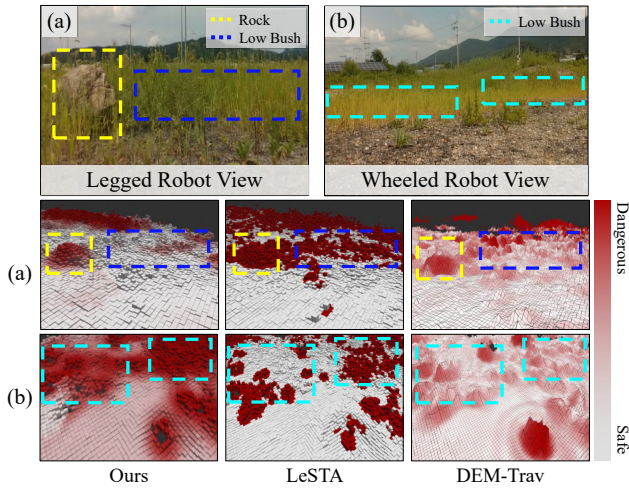


Fig. 6. Traversability mapping results comparing benchmark methods in the real-world. (a) Legged robot: low bush regions are traversable, rock regions are dangerous. (b) Wheeled robot: low bush regions are non-traversable.

three terrain types: uneven road, low bush (< 0.6 m), and high bush (≥ 0.6 m). Based on assumed platform-specific capabilities, the legged robot was assigned to traverse *uneven road* and *low bush*, while the wheeled robot was restricted to *uneven road*. Each robot’s trajectory was split into 80% for training and 20% for validation. For testing, both robots navigated an identical evaluation trajectory, as shown in Fig. 5(b).

Our approach utilized the training parameters detailed in Sec. IV-A. For LeSTA, we used input features (step-height, slope, roughness, and curvature) with a learning rate of 0.7. For DEM-Trav, we employed step-height, roughness, and slope features with robot-specific maximum thresholds: 1.4 m, 0.2 m, and 1.0 rad for the legged robot, and 0.6 m, 0.05 m, and 1.0 rad for the wheeled robot, respectively. All methods used a grid size of 8×8 m with a 0.15 m resolution.

2) *Traversability Mapping Results*: As shown in Fig. 6 and Fig. 7, our method successfully generated robot-specific traversability maps. For the legged robot, low bush regions were correctly identified as traversable, whereas for the wheeled robot, bush areas were appropriately predicted as non-traversable. Both robots consistently predicted rock and hill regions as non-traversable. Notably, the regions highlighted in green (Fig. 7) demonstrate distinct traversability scores for identical uneven road surfaces, accurately reflecting the wheeled robot’s kinematic limitations by assigning higher risk to such terrain.

In contrast, both LeSTA and DEM-Trav exhibited limited adaptability across platforms. LeSTA predicted low bushes as non-traversable for the legged robot and incorrectly estimated high bushes as traversable for the wheeled robot. Similarly, DEM-Trav predicted low bushes as high-risk terrain for the legged robot and also failed to handle dense bushes for the wheeled robot. Ultimately, DEM-Trav relies on human-defined capability thresholds that may not accurately reflect platform-specific mobility constraints, whereas LeSTA, by focusing on terrain properties, may fail to fully capture robot-dependent traversability characteristics.

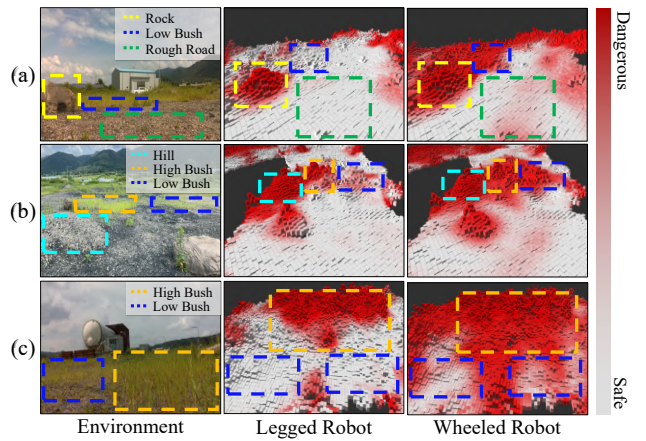


Fig. 7. Detailed traversability mapping results of the proposed method, illustrating robot-specific terrain assessment for legged and wheeled robots.

TABLE III
NAVIGATION PERFORMANCE COMPARISON

Method	Avg. Collisions ↓	Success Rate (N/10)
DEM-Trav	6.6	4/10
LeSTA	4.6	6/10
Ours	0.2	10/10

B. Traversability-aware Navigation

1) *Experimental Setup*: We evaluated off-road navigation in a Gazebo simulation environment including complex hillside terrain, varying slopes, rigid obstacles (e.g., trees, rocks, and high bushes), and passable vegetation such as low bushes. The experiments employed a Husky robot equipped with an OS1-64 LiDAR. The navigation stack utilized the DWA planner [23] operating on local traversability maps without a global map. Local costmaps were generated using a 0.5 threshold to distinguish occupied from passable regions, with a grid size of 8×8 m and 0.15 m resolution. For evaluation, we measured the average number of collisions and the success rate over 10 trials. After each collision, the robot executed a recovery procedure and resumed navigation. A mission was deemed unsuccessful if the recovery procedure failed.

To ensure a fair comparison, both learning-based methods (LeSTA and our approach) were trained on datasets collected from identical trajectories within the hill world environment [24], where positive samples consisted of traversals across *uneven roads*, *slopes*, and *low bushes*. The collected data were split into 80% for training and 20% for validation. The training parameters for our method and LeSTA followed the same configurations detailed in Sec. V-A. For DEM-Trav, platform-specific thresholds were configured for the Husky robot, set to 0.3 m, 0.1 m, and 1.0 rad for step height, roughness, and slope, respectively.

2) *Navigation Results*: As shown in Table III, our method consistently reached the goal with stable trajectories, achieving a perfect success rate (10/10) and a minimal average collision count of 0.2. In contrast, both LeSTA and DEM-Trav frequently misclassified *low bushes* as non-traversable, leading to multiple navigation failures with success rates of 6/10 and 4/10,

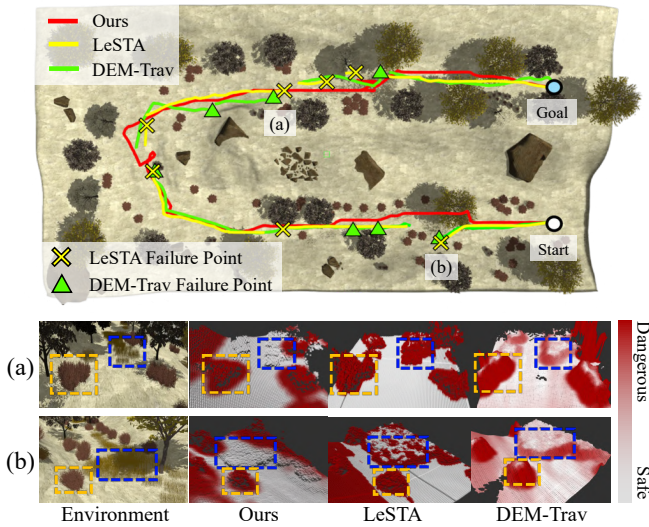


Fig. 8. A representative example summarizing a successful navigation task in the simulation. Trajectory lines and collision markers are shown for each method. Detailed views (a) and (b) highlight vegetation regions, where orange boxes indicate non-traversable areas and blue boxes denote traversable bushes.

respectively. The representative navigation trajectories for each method are visualized in Fig. 8.

Specifically, DEM-Trav relied on rigid, human-defined parameters, leading to inconsistent traversability predictions across varying slopes and step heights. Similarly, LeSTA struggled to achieve a balanced representation of diverse terrain features, which caused the misclassification of ambiguous regions, such as *low bushes*. Consequently, the planner treated passable vegetation as obstacles, leading to navigation failures. In contrast, our method successfully navigated the complex environment by leveraging traversal experience.

VI. CONCLUSION & DISCUSSION

This paper presents an experience-aware framework for self-supervised traversability estimation that leverages robot traversal experience. By incorporating anomaly detection into the traversability learning, our approach overcomes the inherent limitations of PU learning caused by the positive-only constraint. In addition, targeted augmentation strategies were applied to expand the diversity of traversal experiences, improving adaptability to unseen regions. We further validated the practical applicability of our approach through navigation tasks.

Despite its effectiveness, our framework has two limitations. First, providing supervision to empty cells may lead to training instability. Second, robot-state information such as proprioceptive signals (e.g., battery status or motor conditions) is not yet incorporated, which would be essential for a more comprehensive traversability assessment.

Future work will focus on incorporating uncertainty-aware learning to mitigate instability from incomplete measurements caused by occlusion and integrating robot-state information to achieve more robot-specific traversability estimation. In addition, as existing class-based datasets do not adequately reflect actual traversability, we plan to develop comprehensive datasets with ground-truth traversability to enable more accurate evaluation.

REFERENCES

- [1] S. Hosseinpoor, J. Torresen, M. Mantelli, D. Pitto, M. Kolberg, R. Maffei, and E. Prestes, "Traversability analysis by semantic terrain segmentation for mobile robots," in *2021 IEEE 17th international conference on automation science and engineering (CASE)*. IEEE, 2021, pp. 1407–1413.
- [2] 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.
- [3] T. Shan, J. Wang, B. Englot, and K. Doherty, "Bayesian generalized kernel inference for terrain traversability mapping," in *Conference on robot learning*. PMLR, 2018, pp. 829–838.
- [4] A. Leininger, M. Ali, H. Jardali, and L. Liu, "Gaussian process-based traversability analysis for terrain mapless navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10925–10931.
- [5] M. V. Gasparino, A. N. Sivakumar, and G. Chowdhary, "Wayfaster: a self-supervised traversability prediction for increased navigation awareness," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8486–8492.
- [6] S. Ægidius, D. Hadjivelichkov, J. Jiao, J. Embley-Riches, and D. Kanoulas, "Watch your step: Semantic traversability estimation using pose projected features," *arXiv preprint arXiv:2501.17594*, 2025.
- [7] M. Mattamala, J. Frey, P. Libera, N. Chebrolov, G. Martius, C. Cadena, M. Hutter, and M. Fallon, "Wild visual navigation: fast traversability learning via pre-trained models and online self-supervision," *Autonomous Robots*, vol. 49, no. 3, pp. 1–18, 2025.
- [8] I. Cho and W. Chung, "Learning self-supervised traversability with navigation experiences of mobile robots: A risk-aware self-training approach," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4122–4129, 2024.
- [9] H. Xue, X. Hu, R. Xie, H. Fu, L. Xiao, Y. Nie, and B. Dai, "Contrastive label disambiguation for self-supervised terrain traversability learning in off-road environments," *arXiv preprint arXiv:2307.02871*, 2023.
- [10] J. Seo, S. Sim, and I. Shim, "Learning off-road terrain traversability with self-supervisions only," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4617–4624, 2023.
- [11] J. Seo, T. Kim, K. Kwak, J. Min, and I. Shim, "Scate: A scalable framework for self-supervised traversability estimation in unstructured environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 888–895, 2023.
- [12] Y. Bu, Z. Sun, X. Li, J. Zeng, X. Zhang, and H. Shen, "Self-supervised traversability learning with online prototype adaptation for off-road autonomous driving," *IEEE Robotics and Automation Letters*, 2025.
- [13] S. Jung, J. Lee, X. Meng, B. Boots, and A. Lambert, "V-strong: Visual self-supervised traversability learning for off-road navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 1766–1773.
- [14] Y. Kim, J. H. Lee, C. Lee, J. Mun, D. Youm, J. Park, and J. Hwangbo, "Learning semantic traversability with egocentric video and automated annotation strategy," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10423–10430, 2024.
- [15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.
- [16] F. A. Ruetz, N. Lawrance, E. Hernández, P. V. Borges, and T. Peynot, "Forestnav: 3d lidar-only forest traversability estimation for autonomous ground vehicles," *IEEE Access*, vol. 12, pp. 37 192–37 206, 2024.
- [17] J. Frey, M. Patel, D. Atha, J. Nubert, D. Fan, A. Agha, C. Padgett, P. Spieler, M. Hutter, and S. Khattak, "Roadrunner-learning traversability estimation for autonomous off-road driving," *IEEE Transactions on Field Robotics*, 2024.
- [18] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.
- [19] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," *arXiv preprint arXiv:1906.02694*, 2019.
- [20] 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.
- [21] J. Kim, H. Kim, S. Jeong, Y. Shin, and Y. Cho, "Diter++: Diverse terrain and multi-modal dataset for multi-robot slam in multi-session environments," *arXiv preprint arXiv:2412.05839*, 2024.
- [22] M. Wermelinger, P. Fankhauser, and R. Kaestner, "Traversability estimation," GitHub repository, Robotic Systems Lab, ETH Zurich, 2025, accessed: 2025-09-08. [Online]. Available: https://github.com/leggedrobotics/traversability_estimation
- [23] 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, 2002.
- [24] M. Sánchez, J. Morales, J. L. Martínez, J. J. Fernández-Lozano, and A. García-Cerezo, "Automatically annotated dataset of a ground mobile robot in natural environments via gazebo simulations," *Sensors*, vol. 22, no. 15, p. 5599, 2022.