Expert Systems With Applications 292 (2025) 128412

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

ANZIL: Attention-based network for zero-risk inspection of LiDAR point cloud in self-driving cars

Grafika Jati [a,*], Martin Molan [a], Francesco Barchi [a], Andrea Bartolini [a], Giuseppe Mercurio [b], Andrea Acquaviva [a]

[a] University of Bologna, Viale del Risorgimento 2, Bologna, 40136, Italy
[b] FEV Italia s.r.l., Via Livorno 60, Torino, 10144, Italy

ARTICLE INFO

Keywords:
LiDAR contamination
Point cloud corruption
Object detection
Graph attention networks
Edge processing
Model-agnostic
Safety–critical autonomous self–driving cars

ABSTRACT

LiDAR is widely used in autonomous vehicle perception, and its performance relies on algorithmic confidence. However, sensor contamination can lead to catastrophic mistakes in downstream tasks, such as incorrect object detections occurring with high confidence. This underscores the need for point cloud contaminant detection to identify data reliability before downstream processing. To overcome this, we propose a model-agnostic approach that integrates contaminant detection with downstream tasks, using graph representations and attention networks. Trained on real contaminated LiDAR, the contaminant detector complements any existing clean-trained downstream model. Point clouds passing through the contaminant detector are discarded if contamination is detected. We propose a novel cost-benefit methodology, evaluating contaminant detectors in downstream processing. The benefit measures the proportion of discarded frames that would have caused high-confidence errors in the downstream task. The cost includes the model misalignment cost, representing frames wrongly discarded despite being correctly processed, and the true cost, which reflects uncontaminated frames falsely identified as contaminated. The contaminant detector was tested on 78,000 frames with water, dust, mud, salt, oil, and foam contamination from tunnel and outdoor locations. It achieved an F1-score of 0.884 and a recall of 0.957 on a static dataset, and a 0.975 F1-score in unseen real-world environments, demonstrating high sensitivity. In model-agnostic evaluation, the method is applicable to any object detector and reduces catastrophic mistakes by at least 97 %, successfully identifying 4046 failure cases. Deployed on the NVIDIA Jetson AGX Xavier, it achieved inference in under 5 ms and point cloud transformation in 200 ms, making it suitable for edge processing. This enhances sensor reliability, enables automatic cleaning, and improves vehicle safety. Data and code available at: https://gitlab.com/ecs-lab/anzil

1. Introduction

The pursuit of unparalleled safety and reliability in self-driving cars has driven advances in sensor technologies, artificial intelligence, and computer vision (Khan, Menouar, Abdallah, & Abu-Dayya, 2024). Among these, LiDAR stands out for providing precise 3D representations of the environment (Minh Mai et al., 2022; Tang, Shi, He, Sadek, & Qiao, 2020). In autonomous driving, LiDAR enables object detection models that underpin key tasks such as reinforcement learning, planning, control, and navigation (Pandharipande et al., 2023; Pérez-Gil et al., 2022). These models are often trained on clean sensor data, overlooking real-world conditions where data may be noisy or corrupted. Such contamination can distort object representations and erode user trust. Adverse weather like rain or fog further degrades performance, reducing detection accuracy (Zhang, Carballo, Yang, & Takeda, 2023). For instance, PointRCNN's Average Precision drops from 80.57 to 51.27 in rain (Dong et al., 2023a; Mirza et al., 2021; Zhao et al., 2024b), severely impairing decision-making (Zhao et al., 2024a).

Sensor contamination presents a critical challenge distinct from environmental conditions, as it directly affects raw sensor data, leading to distorted or incomplete point clouds (Matos, Bernardino, Durães, & Cunha, 2024). Studies indicate that even synthetic contamination significantly degrades performance, with Gaussian noise reducing object detection accuracy from 80.57 to 61.20 (Dong et al., 2023b) and LiDAR point reduction further impacting performance (Beemelmanns, Zhang, Geller, & Eckstein, 2024). Given the adverse effects of synthetic

* Corresponding author.
E-mail addresses: grafika.jati2@unibo.it (G. Jati), martin@molan.net (M. Molan), francesco.barchi@unibo.it (F. Barchi), a.bartolini@unibo.it (A. Bartolini), mercurio_g@fev.com (G. Mercurio), andrea.acquaviva@unibo.it (A. Acquaviva).

https://doi.org/10.1016/j.eswa.2025.128412
Received 2 February 2025; Received in revised form 18 April 2025; Accepted 29 May 2025
Available online 7 June 2025
0957-4174/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

contamination, real-world contamination is expected to have an even more severe impact. Beyond reduced accuracy, contamination can cause catastrophic failures, such as high-confidence false positives, false negatives, and ghost obstacles. Built-in object detection systems often fail to address contamination effectively, leading to false detections with high confidence scores (e.g., 0.9 out of 1.0) (Jati et al., 2024b). Contaminants like mud and lubricant are particularly detrimental, posing significant safety risks, including ghost object detection, which can trigger fatal automatic braking events. However, research on real contamination in LiDAR perception remains limited and underexplored.

Retraining downstream perception models, such as object detection, presents significant challenges, particularly due to data dependency, which requires large, diverse, and well-annotated datasets covering various contaminant scenarios-making the process costly and time-consuming (Gao, Pan, Li, Geng, & Zhao, 2022). Training with simulated or artificially generated noisy data often fails to replicate real-world noise characteristics, resulting in a 14 % reduction in object detection performance when trained on simulated data (Huch, Scalerandi, Rivera, & Lienkamp, 2023). This indicates a significant performance gap between simulated and real-world data (Chen, Wang, Zou, Chen, & Jia, 2022).

While quality metrics exist for cameras and testing standards for radar and LiDAR, there is no standardized method to detect data degradation level before object detection processing. Existing LiDAR degradation scores (Shahbeigi, Robinson, & Donzella, 2024) are well-suited for monitoring overall data quality levels. However, they are not primarily designed to provide timely alerts regarding the reliability of LiDAR data for downstream tasks or to pinpoint specific causes such as contamination. A dedicated framework for data integrity checks tailored to automotive perception systems is essential to ensure model reliability under degraded sensor conditions (Baccari, Hadded, Ghazzai, Touati, & Elhadef, 2024; Bogdoll, Nitsche, & Zöllner, 2022).

In this paper, we propose a model-agnostic safety system that integrates contaminant detection with any downstream perception model by pre-processing LiDAR point clouds and filtering out contaminated frames before they reach downstream tasks. This approach improves perception accuracy and system resilience without modifying existing models. Our system leverages graph representations and Graph Attention Networks (GAT) for robust contaminant detection, offering invariance to point cloud resolution, intensity, and sparsity. Unlike CNNs, which process empty space in sparse or dense inputs, graph-based methods focus on meaningful point connections, enhancing feature extraction and ensuring consistent performance across different LiDAR types and scenes (Li et al., 2024). The interaction between our safety and perception models is illustrated in Fig. 1.

Our Graph Attention Network (GAT)-based contaminant detection algorithm achieves an average F1 score of 88 % and 95 % recall, reliably identifying all critical contaminations. Operating independently, it delivers nearly 100 % benefit by detecting 4046 catastrophic frames with minimal true cost. Deployed on the NVIDIA Jetson AGX Xavier, the system supports real-time ADAS use, processing data in 200 ms with sub-5 ms inference.

1.1. Contributions

The main contributions of this paper are as follows:

- **Model-agnostic safety system:** We introduce a model-agnostic safety system that can be integrated with any downstream perception model. This safety system is a contaminant detection developed using graph representations and Graph Attention Networks (GAT), enabling it to generalize across dynamic environments. The approach enhances computational efficiency and feature extraction by focusing on meaningful connections within the point clouds.
- **Cost-benefit evaluation methodology:** We propose a cost-benefit framework to evaluate contaminant detection in perception models. The benefit score measures the reduction of high-confidence errors, while cost metrics assess the impact of discarding contaminated point clouds, particularly when they do not significantly affect perception performance. An adjustable aggressiveness threshold for contaminant detection optimizes the trade-off between benefit and cost.
- **Comprehensive real contaminated point cloud evaluation with a novel dataset:** We evaluate the scalability and generalization of contaminant detection using unseen, realistic, various LiDAR specification, and dynamic contaminated point cloud frames. Object detection performance is then benchmarked through one-to-one frame matching with contaminant detection results, providing a comprehensive assessment of its impact on perception models in preventing high-confidence false object detections.
- **Edge device implementation:** The contaminant detection is deployed on the NVIDIA Jetson AGX Xavier, completes processing in around 200 milliseconds, and inference in under 5 milliseconds, suitable for edge computing in ADAS while improving sensor reliability and reducing critical downstream errors.

This paper is organized as follows: Section 1 introduces the problem, motivation, and significance of this research; Section 2 reviews related work; Section 3 presents a concise formulation of the problem, the proposed contaminant detection approach, and the novel cost-benefit framework for evaluating contaminant detection in downstream tasks; Section 4 details the experimental setup, results, and evaluation metrics; Section 5 presenting contaminant detection as model agnostic to any perception model along with cost benefit analysis; Section 6

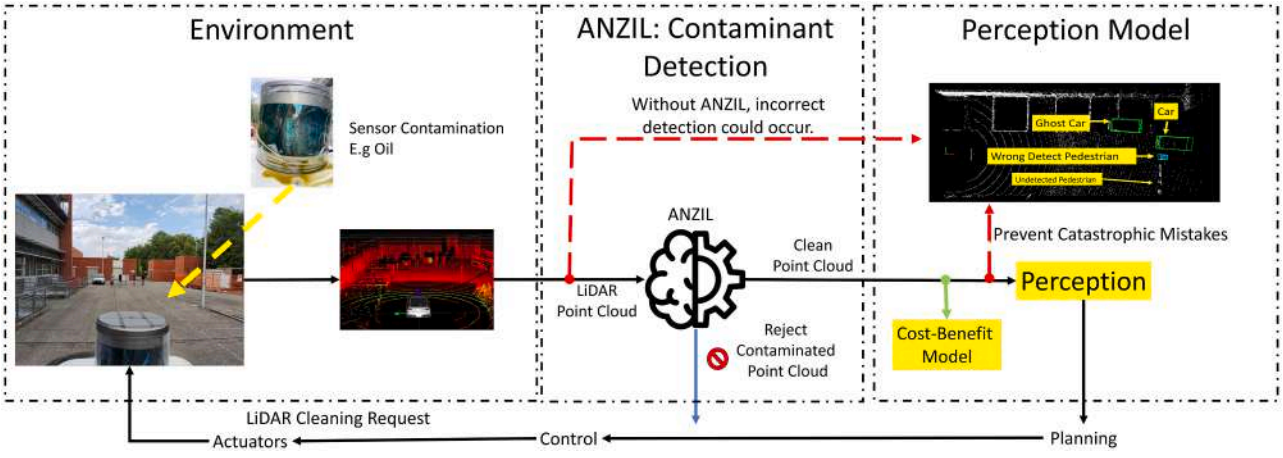


Fig. 1. Proposed integration of contaminant detection as a safety model within the perception model of a self-driving system.

describing novel point cloud contamination dataset and assessing our Contaminant Detection in unseen dynamic environment; Section 7 discusses the deployment of contaminant detection on edge processing; and finally, Section 8 concludes the study.

2. Related work

2.1. Ensuring reliability of sensors for autonomous driving

LiDAR, widely used for accurate depth estimation, is now a production-ready sensor in commercial markets (Liang, Zhang, Zhang, Liu, Li, Niu, Rao, Zhao, Chen, Li, & Huo, 2024). The focus has shifted to robustness testing and functional safety (Schillgalies & Sharman, 2022), emphasizing the need to maintain data quality under adverse road conditions. Also, resilience as the ability of a system to recovery its functionality in presence of a damage of the sensor is a relevant aspect (Han, Liu, & Zhang, 2016). According to Bogdoll's survey, research on detecting perception anomalies remains limited (Bogdoll et al., 2022). Perception models perform well under normal conditions but struggle when the world model cannot explain observations, leading to false positives and negatives. End-to-end evaluations make identifying root causes difficult, often resulting in error propagation. Isolating the issue is crucial, as it may stem from a mix of internal and external factors, known as domain-level phenomena. Domain shifts in appearance caused by snow, fog, or rain introduce noisy LiDAR readings. Studies by Filgueira, González-Jorge, Lagüela, Díaz-Vilariño, and Arias (2017); Zhang, Huang, Ang, and Rus (2021) quantify LiDAR degradation under the rain, with Zhang proposing a degradation score formula using DeepSAD for feature extraction. Jati et al. conducts the latest evaluation on real contaminated LiDAR evaluates LiDAR performance under more dynamic contaminants and realistic settings (Jati et al., 2024b). The results show that object detection not only suffers from reduced accuracy but also catastrophic errors, where PointRCNN, trained on clean datasets, misclassifies objects with high confidence.

2.2. LiDAR point cloud corruption

Large-scale datasets like KITTI, NuScenes, and Waymo primarily cover limited scenarios and are captured under clean conditions, as noted (Ren, Pan, & Liu, 2022). Dong et al. evaluated 3D object detection using synthetic data by introducing five corruption types-weather, sensor, motion, object, and alignment issues-into clean datasets (Dong et al., 2023a). Their results revealed a significant reduction in detection precision, emphasizing the need to train models on corrupted data. Most autonomous vehicle perception models, such as clear weather and clean sensors, remain trained and tested under ideal conditions.

Training with synthetic corrupted data is insufficient due to its discrepancies with real-world corruption, emphasizing the need for real-world data. Motion and weather corruption significantly impact system performance, with weather being critical. Sensor cover contamination also poses a major external corruption challenge. Li et al. classified LiDAR corruptions into scene-level (e.g., rain, fog, snow) derived from real datasets and object-level (e.g., noise, density changes) simulated corruptions (Li et al., 2023). Beemelmanns et al. introduced MultiCorrupt, simulating adverse conditions with ten synthetic corruption types affecting LiDAR (L), cameras (C), or both (LC), including motion blur, spatial misalignment, and point reduction (Beemelmanns et al., 2024). However, these datasets lack sensor cover contamination, a critical factor for understanding point cloud corruption.

The study by Schlager, Goelles, Muckenhuber, and Watzenig (2022) introduces real sensor-level contamination with various corruptions, focusing on performance degradation and applications in fault detection. However, the dataset was collected under laboratory conditions, using a single static target with no obstacles, resulting in limited point cloud data that fails to represent real-world scenarios. In contrast, Jati et al. (2024b) provides a state-of-the-art LiDAR corruption dataset featuring diverse contamination scenarios across multiple environments and road-relevant objects, making it highly relevant for advancing contaminant detection methods.

2.3. Ensuring resilience of LiDAR point cloud processing task

Several studies have been conducted to identify conditions during point cloud acquisition. Heinzler et al. achieved 97.14 % accuracy in classifying weather conditions (clear, rain, fog) using LiDAR point clouds from Velodyne and Scala sensors with SVM and KNN in controlled environments (Heinzler, Schindler, Seekircher, Ritter, & Stork, 2019). Their 1D aggregation method required complex feature engineering and focused on weather classification rather than LiDAR contamination. Similarly, Rivero et al. employed KNN to classify LiDAR data under weather conditions such as rain, fog, and snow but did not address sensor contamination (Vargas Rivero, Gerbich, Teiluf, Buschardt, & Chen, 2020). Sebastian also focused on weather classification, not on sensor contamination (Sebastian, Vattem, Lukic, Bürgy, & Schumann, 2021). Egelhof's latest study evaluated filtering methods for detecting particles like fog, dust, rain, and snow (Egelhof, Wolf, & Berns, 2022).

In terms of sensor contamination, James et al. proposed a LiDAR sensor contaminant classification method using a Deep Neural Network (James, Puhlfürst, Golyanik, & Stricker, 2018). They created datasets with contaminants like dirt, salt, and frost, converting LiDAR data into 2D images and applying a 2D CNN to classify clean and fully contaminated data. However, this method has not been validated for partial or gradual contamination, leaving the application of ML/DL for detecting multiple contaminants on LiDAR covers underexplored.

3. Methodology

Autonomous Driving and ADAS rely on sensor data, such as LiDAR point clouds, to feed critical information to planning, control, and actuation modules. The proposed model-agnostic contaminant detection solution enhances the perception system's integrity and can be applied to any downstream task. Its integration is supported by two key modules: the Contaminant Detection model and the Cost-Benefit model, as illustrated in Fig. 1, emphasizing their role within the ADAS framework.

After acquiring point cloud data, we propose adding a Contaminant Detection system before passing it to perception modules like object detection. This system evaluates whether the point cloud is clean or contaminated. Clean data proceeds to tasks such as object detection, segmentation, or tracking, while contaminated data is rejected to prevent errors like ghost objects or misclassified pedestrians. Additionally, the system can trigger automated cleaning procedures, addressing sensor contamination in real time and supporting maintenance for non-real-time scenarios. Unlike adverse weather, sensor contamination is manageable through timely interventions enabled by contaminant detection.

We propose a model-agnostic cost-benefit framework to determine the optimal integration of contaminant detection before downstream perception models. Using object detection as a use case and a ground-truth contaminant point cloud dataset, we benchmark detection models to evaluate the necessity of contaminant detection in terms of cost and benefit. The cleaning process is designed for implementation on ADAS platforms, demonstrated through deployment on the NVIDIA platform, a cutting-edge edge computing solution for autonomous driving. This section covers LiDAR corrupted point clouds using the latest comprehensive dataset on sensor contamination, the proposed Graph Transformation and Graph Convolutional Network for contaminant detection, and the Cost-Benefit Model to evaluate the necessity of contaminant detection for downstream perception algorithms.

3.1. Corrupted LiDAR point cloud

We used the high-resolution contaminated point cloud dataset from Jati et al. (2024b) called LIDAROC, collected with a 128-channel

Fig. 2. Implementation of Various Types and Levels of Contamination on the LiDAR Surface.

RS-Ruby LiDAR sensor, featuring 0.1° vertical resolution, 360° horizontal FoV, 40° vertical FoV, and a 250-meter range, supporting Level 4+ autonomy. The dataset was generated under controlled real-world conditions, testing contaminants such as water, mud, dust, salt water, lubricant, and foam at three levels. Contaminants were applied via spraying, dripping, or drying, simulating realistic effects, with liquid-based contaminants partially cleaning the sensor at high levels. Each scenario was repeated three times to ensure diverse and consistent data, as shown in Fig. 2. This data set consists of three distinct subsets: an underground tunnel with a sensor distance of 5 m to the object, referred to as 'tunnel-5m', and an outdoor parking area with sensor distances of 10 m and 20 m to the object, referred to as 'outdoor-10m' and 'outdoor-20m', respectively. The dataset includes cars, motorcycles, pedestrians, and reflective materials to simulate relevant road scenarios. Fig. 3 depicts the experimental settings and the collected point clouds. Fig. 3 visualizes the effect of sensor conditions on point clouds, including clean sensors and those contaminated with water, mud, and oil, across three different environments.

3.2. Problem in detecting corrupted Lidar point cloud

Contaminant detection is challenging in dynamic ego-car environments, as varied settings produce diverse point cloud structures and attributes. Fig. 3 highlights contamination effects at the scene and object levels, altering point count, shape, and intensity. Red (255, 0, 0) in the RGB spectrum indicates areas with the lowest intensity. While all contamination types show similar symptoms, their severity varies. Since LiDAR point cloud intensity depends on object material and distance (Benedek, Majdik, Nagy, Rozsa, & Sziranyi, 2021), contaminants cannot be identified solely by intensity changes, as their effects can

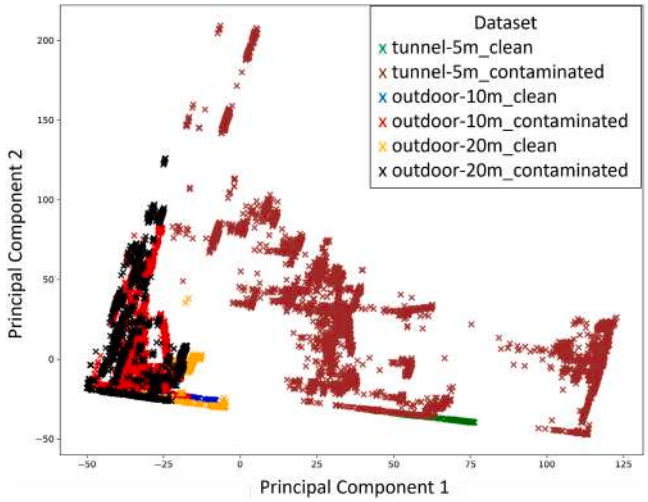


Fig. 4. Data distribution of clean and contaminated LiDAR point clouds from three different environments.

mimic increased sensor-to-object distance. For instance, clean data from an outdoor-20m parking area may resemble water-contaminated data from a tunnel-5m, complicating detection in both dense and sparse point clouds.

The distribution and classification complexity of "clean" and "contaminated" data across tunnel-5m, outdoor-10m, and outdoor-20m datasets reveals significant overlap, as shown in Fig. 4. This figure is produced by applying PCA to reduce 28 statistical features (derived from point cloud coordinates and intensity) into two dimensions. The overlap is particularly evident in the tunnel-5m and outdoor-10m datasets, where "clean" data forms tight clusters, while "contaminated" data exhibits broader and more varied distributions, reflecting diverse contamination levels and types. These findings highlight the difficulty of distinguishing between the two classes using statistical features alone.

Training and testing across datasets from different environments are crucial for generalizable contaminant detection but face challenges like distribution overlap, domain shifts, and contaminant diversity. Distribution overlap in tunnel-5m and outdoor-10m datasets blurs feature boundaries between clean and contaminated classes, increasing false positives and negatives. Domain shifts, such as training on tunnel-5m and testing on outdoor-10m or outdoor-20m, alter feature distributions and reduce model accuracy, especially with the more significant contamination variability in outdoor-20m datasets.

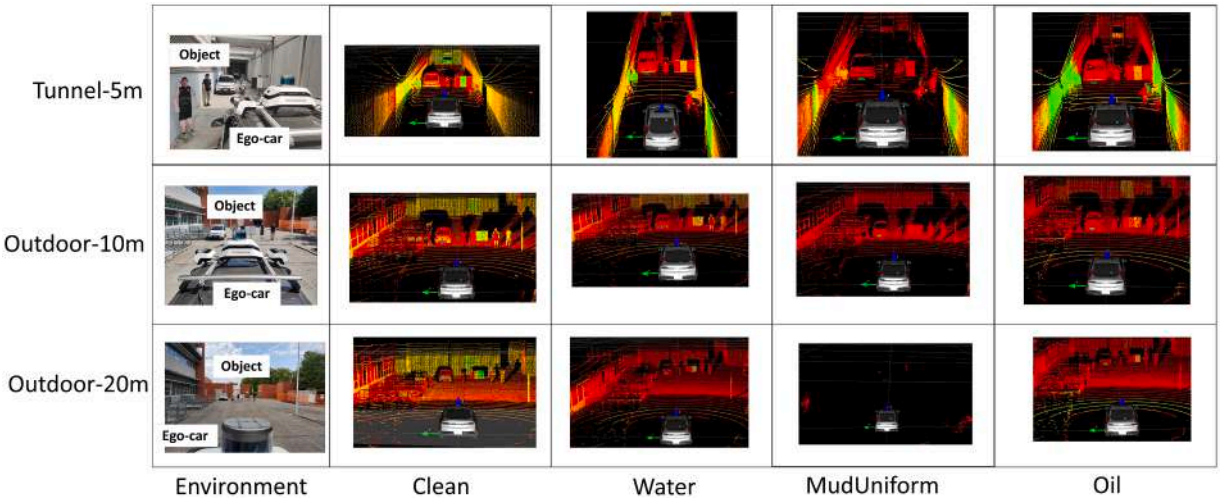


Fig. 3. Visualization of effect sensor condition: clean and contaminated, for point cloud in three different environments.

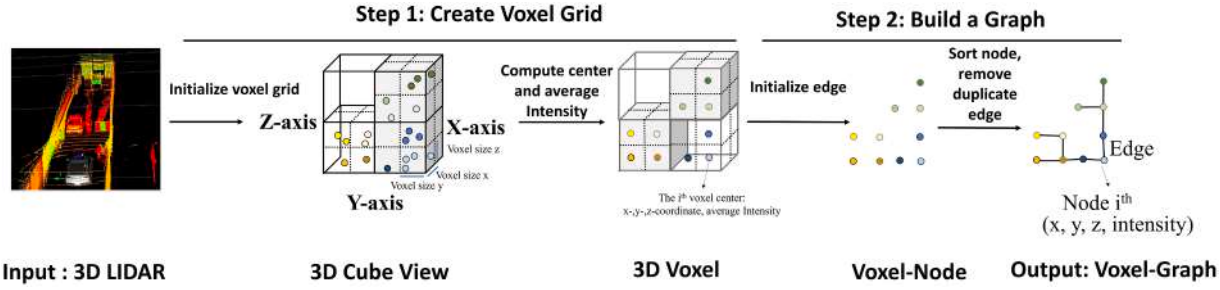


Fig. 5. The proposed graph transformation for the LiDAR point cloud.

Contaminant diversity, with varying types and severities across datasets, affects a model's ability to generalize and perform consistently in different environments. Classifiers overfitted to specific datasets often misclassify or lose accuracy on new data. This underscores the need for effective feature design and domain adaptation to enhance performance. A contaminant detection model resilient to distance and environmental variations is crucial for practical applications.

3.3. Proposed graph-attention contaminant detection

We propose a framework for detecting contaminants on LiDAR sensors. The method consists of three stages: 1) voxelization to sparsify the input point cloud, 2) construction of a spatial graph structure from the voxels, and 3) application of a graph attention network to classify whether the sensor is contaminated.

3.3.1. Point cloud graph transformation

The proposed method transforms point clouds into graphs by voxelization, as shown in Fig. 5. Point cloud data are converted into a 3D voxel grid of uniform size, where each voxel aggregates spatial and intensity features through average pooling, reducing computational complexity while preserving key information. Non-empty voxels are transformed into graph nodes, with edges representing spatial adjacency in three dimensions. Empty voxels are excluded, optimizing processing efficiency. The resulting graphs vary in size based on scene and contamination effects. As illustrated in Algorithm 1, graph construction sorts voxel centers along the x, y, z axes using QuickSort, connecting adjacent voxels with undirected edges to capture critical spatial relationships for contamination detection.

Algorithm 1: Point cloud to graph conversion.

Input: Point cloud data points, Voxel size voxel_size
Output: Graph node features graph_x, Edge indices graph_edges_index

Step 1: Create voxel grid
Initialize voxel_grid from points with voxel_size;
foreach *voxel in voxel_grid* **do**
    Compute center (x, y, z) and average intensity;
    Append $(x, y, z,$ intensity$)$ to voxel_data;
end
Step 2: Build graph
Initialize edge_index;
foreach *axis in* $\{x, y, z\}$ **do**
    Sort voxel_data along the axis;
    Connect consecutive voxels and add edges to edge_index;
end
Remove duplicates and self-loops from edge_index;
Step 3: Output graph
Set graph_x = voxel_data;
return *graph_x, graph_edges_index*;

The algorithm for Point cloud graph transformation consists of two essential blocks: voxel and graph making. The voxel-making process has a complexity of $O(N + V)$, where N is the number of points in the point cloud, and V is the number of voxels created. Mapping points to voxels requires $O(N)$ while aggregating voxel features such as center coordinates and average intensity requires $O(V)$. For graph making, sorting the voxel centers along three dimensions (x, y, z) using QuickSort dominates the complexity at $O(V \log V)$. Edge construction, where consecutive points in sorted lists are connected, adds $O(V)$, and duplicate edge removal using coalescing incurs an additional $O(V \log V)$. Overall, the graph-making process has a complexity of $O(V \log V)$, dominated by the sorting step. Edges are constructed as undirected because the contamination primarily affects intensity and object structure rather than introducing directional effects. This simplifies the graph while preserving the spatial relationships between nodes.

Finally, the proposed transformation method offers several advantages: Sorting Dominance, where sorting voxel centers, though computationally intensive, preserves spatial relationships for accurate graph representation; Efficient Edge Construction, operating at $O(V \log V)$ and scaling well with voxel count; Undirected Edges, simplifying graph construction while capturing spatial structures and intensity distributions; and Scalability, with voxelization reducing data size $V \ll N$ for efficient graph creation in large, complex point clouds.

3.3.2. Point cloud graph attention for detection

We developed a contaminant detection model using Graph Attention Networks (GATs), which effectively handle the sparse nature of LiDAR-generated point clouds by representing them as graphs, with each point as a node. The model uses multi-head self-attention, where each attention head computes distinct attention coefficients between a node and its neighbors. We utilize the GATConv layer from the torch_geometric.nn library inpired from Veličković et al. (2018), which encapsulates this functionality. The attention mechanism relies on LeakyReLU activation with a default negative slope of 0.2. We use concatenation, so the outputs from all heads are concatenated. Our approach focuses on the spatial relationships between LiDAR points within a single frame, aligning with the spatial nature of GAT attention-meaning it emphasizes the relational structure between nodes in a graph rather than temporal sequences. GATs leverage attention mechanisms to focus on relevant node subsets, adapt to local graph structures, and are thus well-suited for contamination detection tasks.

The proposed GATs model set input voxel size of $0.5 \times 0.5 \times 0.5$ m and a two-layer GATConvolution architecture. Fig. 6 outlines the proposed GATs model for contaminant detection. The model architecture (Table 1) includes two GATConv layers for processing node features, with two attention heads in the first layer and 1 in the second, reducing computational complexity while maintaining efficiency. Batch Normalization, dropout, and global mean pooling enhance stability and prevent overfitting. The training setup uses a batch size of 32, early stopping with a patience of 10, and a maximum of 100 epochs. Designed for binary classification, the model outputs normalized class probabilities using Log_softmax and is trained with the Adam optimizer (learning rate

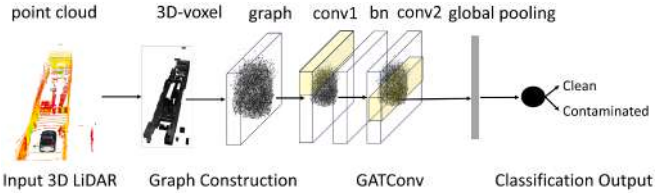


Fig. 6. The proposed LiDAR contaminant detection using graph attention networks.

Table 1
Architecture of GAT for LiDAR contaminant detection.

Layer	Parameters
conv1 (GATConv)	
Attention Heads	2
Output Features (Hidden Dimension)	16
Parameters:	
att_src	32
att_dst	32
bias	32
lin_src.weight	128
Batch Normalization (bn1)	Output Features: $16 \times 2 = 32$
Activation Function	Exponential Linear Unit (ELU)
Dropout (First Layer)	Dropout Probability: 0.3
conv2 (GATConv)	
Attention Heads	1
Output Features (Classes)	2
Parameters:	
att_src	1
att_dst	1
bias	1
lin_src.weight	32
Global Pooling	Global Mean Pooling
Output	Log-Softmax
Total Parameters	259

and weight decay of 1×10^{-4}). A StepLR scheduler reduces the learning rate by 0.7 every 10 epochs.

3.4. The cost-benefit of model-agnostic safety system

As a model-agnostic solution, contaminant detection can be applied to any downstream task. However, contaminant detection needs to be evaluated on an annotated dataset comprising clean and contaminated LiDAR cases to assess the benefits and costs of its implementation. Its integration should balance benefits, such as reducing perception errors, against costs of discarding clean data. To analyze these trade-offs, we propose a cost-benefit model and use Object Detection (OD) as a representative task to assess the impact of Contaminant Detection (CD). Fig. 7 illustrates how CD prevents catastrophic failures by addressing high-confidence errors in OD systems in public available and realistic contaminated LiDAR dataset.

The cost-benefit concept, illustrated in Fig. 7, is rooted in Object Detection (OD) outcomes, meaning it can only be calculated after obtaining OD results. These results are categorized by correctness and confidence, branching into Correct and Wrong predictions. The "Correct" branch includes frames where all predictions are accurate, with bounding boxes (BB) within *1.0* meters on the X axis and *0.5* meters on the Y axis of the ground truth (GT) (Jati et al., 2024b). The "Wrong" branch consists of frames where at least one prediction exceeds these thresholds, results in misdetection, or fails to detect any object.

The second level analyzes "Correct" and "Wrong" predictions based on detection confidence value: "High Confidence" if value ≥ 0.9 and "Low Confidence" is value < 0.9. We use 0.9 as the threshold to determine high confidence in the object detection results produced by the perception model. We refer to this threshold as the *Catastrophic Confidence Threshold*. As a consequence, this threshold justifies the classification of an error as a catastrophic mistake when a wrong detection is made with high confidence.

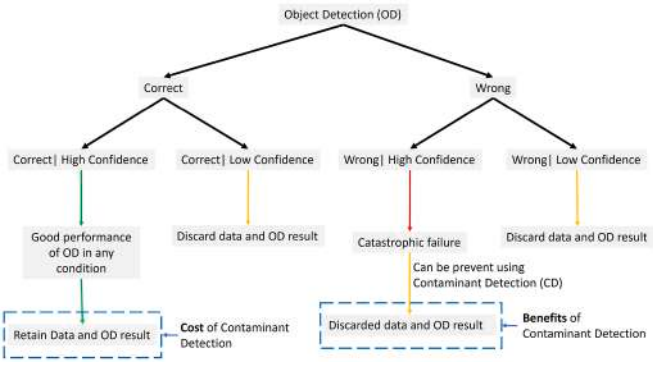


Fig. 7. The cost-benefit definition flow of model-agnostic contaminant detection.

The OD alone retains all high confidence results, regardless of correctness, and only discards low confidence ones. Applying CD unnecessarily discards valid data in correct and highly confident scenarios, incurring extra costs since data is discarded. However, in high confidence wrong predictions, CD acts as a safety filter critical for ADAS. To quantify these costs and benefits, we define the following terms:

- *Positive*: Indicates frames detected as contaminated by the contaminant detection system. This variable represents the number of frames classified as contaminated.
- *Negative*: Indicates frames detected as clean by the contaminant detection system. This variable represents the number of frames classified as clean. It is not included in the cost-benefit calculation since it does not affect the downstream process.
- *Correct* | *High Confidence*: Frames where all object detections are accurate, made with high confidence, and no catastrophic failures occur.
- *Wrong* | *High Confidence*: Frames containing at least one catastrophic failure, defined as an incorrect detection made with high confidence.
- $GT_Contaminated$: Ground truth labels confirming the frame is contaminated.
- $GT_NonContaminated$: Ground truth labels confirming the frame is not contaminated.

Using five variables, we calculate the cost and benefit of Contaminant Detection on the downstream task as follows:

$$Cost = \frac{Positive \cap Correct \mid High_Confidence}{Correct \mid High_Confidence} \tag{1}$$

$$Model_Misalignment_Cost = \frac{\text{Positive} \cap \text{GT_Contaminated} \cap \text{Correct} \mid \text{High_Confidence}}{\text{Correct} \mid \text{High_Confidence}} \tag{2}$$

$$True_Cost = \frac{Positive \cap GT_NonContaminated \cap Correct \mid High_Confidence}{Correct \mid High_Confidence} \tag{3}$$

$$Total_Cost = Model_Misalign_Cost + True_Cost \tag{4}$$

$$Benefit = \frac{Positive \cap Wrong \mid High_Confidence}{Wrong \mid High_Confidence} \tag{5}$$

Our objective is to quantify the performance trade-offs of employing Contaminant Detection (CD) in object detection (OD) systems, par-

ticularly in scenarios involving high-confidence predictions, as these carry significant safety implications for Autonomous Driving Assistance Systems (ADAS). Eq. 1 quantifies the overall proportion of correct object detection with high-confidence detection that CD discards. Intuitively, this represents the "cost" incurred by unnecessarily discarding valid predictions due to CD intervention. The initial cost shown in Eq. 1 can be further divided into $Model_Misalignment_Cost$ (in Eq. 2) and $True_Cost$ (in Eq. 3), as $Positive$ contaminant detection can be split into mutually exclusive subsets: $Positive \cap GT_Contaminated$ and $Positive \cap GT_NonContaminated$.

Eq. 2 further refines this by focusing specifically on detect frame as contaminated and its correctly according to ground truth (GT) or True Positive case. This cost arise when OD correctly detects objects even under contamination, but the CD discards these results. This cost varies according to specific downstream tasks and depends significantly on the performance of the downstream OD model itself.

Eq. 3 represents the percentage of frames mistakenly flagged as contaminated by the CD model, despite ground truth indicating no contamination or False Positive case, while the OD system still operates correctly. It reflects an inherent cost arising solely from errors within the contaminant detection model itself, independent of downstream model characteristics. Thus, it illustrates the intrinsic, model-agnostic penalty of implementing CD. Finally, Eq. 4 integrates both $Model_Misalignment_Cost$ and $True_Cost$, providing a complete measure of the operational drawbacks of incorporating CD. This combined metric enables stakeholders to assess the cumulative impact of CD-related errors and misalignments comprehensively.

The primary benefit of employing CD lies in its ability to mitigate catastrophic failures in downstream systems, such as OD, especially under high-confidence yet incorrect predictions (captured by $Wrong \mid High_Confidence$). The benefit is explicitly quantified by Eq. 5, reflecting the proportion of high-confidence incorrect detections effectively identified and discarded by CD intervention. A higher benefit value demonstrates CD's efficacy in preventing critical errors, enhancing overall system reliability, safety, and resilience by significantly reducing the risk of hazardous object mis-detections.

4. Contaminant detection result and discussion

4.1. Setting experiments

For training, 78 000 frames were sampled across contamination and environment scenarios, selecting 60 out of 600 frames per scenario to minimize redundancy from static objects. Contaminated class instances total 360, 328, and 364 for tunnel-5m, outdoor-10m, and outdoor-20m datasets, respectively, with 360 cases clean per dataset for balance. Since contamination occurs only in the sensor's front area, point clouds (70 000-140 000 points) were cropped accordingly and transformed into graphs with node-edge pairs: (n)1361-(e)3879 for tunnel-5m, (n)2209-(e)6359 for outdoor-10m, and (n)2514-(e)7242 for outdoor-20m. The training utilized an AMD EPYC 74F3 CPU, NVIDIA GeForce RTX 3090 GPU, and 250GB DDR4 RAM.

Robustness was evaluated through cross-environment testing by training models on one dataset (e.g., tunnel-5m) and testing on others (outdoor-10m and outdoor-20m), as well as vice versa, using the full, unsampled datasets. This approach aligned contaminant detection outputs with object detection results, reflecting real-world dynamics and showcasing the model's capability to generalize with limited training data.

4.2. Contaminant detection methods

To emphasize the complexity of contaminant detection in dynamic scenarios, we compare contaminant detection models spanning manual feature engineering from 1D to 2D and 3D transformations.

1D contaminant detection. Inspired by Jati et al. (2024c), we employ 1D transformation for clean and contaminated point cloud. Each point cloud feature, including the x, y, z coordinates and intensity (i) values, is extracted using seven statistical features: minimum, maximum, mean, standard deviation, and percentiles at 25 %, 50 %, and 75 %. This results in 28 statistical features. The 1D input is then processed using shallow machine learning algorithms, including Logistic Regression, Multilayer Perceptron, Support Vector Machine, Naive Bayes, k-nearest Neighbors, Decision Tree, and Random Forest.

2D contaminant detection. For 2D transformation, we use a combination of 2D transformation and 2D EfficientNet that was introduced by Jati, Molan, Barchi, Bartolini, and Acquaviva (2024a), which has already been proven to classify small-sized point clouds. We transform a 3D LiDAR point cloud into a structured 2D grid on the YZ plane. It normalizes y and z coordinates to map points onto a defined grid size, accumulates x and intensity values, and calculates their mean within each grid cell. The resulting 2-channel 2D grid preserves spatial and intensity information. Then, we trained a 2D contaminant detection model called EffNetPoint, a tailored adaptation of EfficientNet-B0 (Tan & Le, 2019). We modified the structure of EfficientNet to accommodate contaminant detection. The base model was initialized without pre-trained weights, and the first convolutional layer was replaced to handle input with 2 channels, x and i, features of the point cloud, using a kernel size of 3x3. Additionally, the classifier was adjusted to align with binary classification.

3D contaminant detection. The 3D transformation uses voxelization to preserve spatial information with a simple 3D Convolutional Neural Network (3D CNN) for benchmarking. This comparison highlights the effectiveness of each approach in addressing contaminant detection challenges. The voxel method preserves the 3D structure of LiDAR data by converting sparse point clouds into a volumetric grid, with each voxel representing a portion of 3D space. While this approach effectively captures spatial structure, LiDAR data's sparsity leaves most voxels empty, increasing computational and memory costs. This necessitates a balance between high resolution and resource efficiency.

4.3. Contaminant detection results

Table 2 compares the F1 scores of various contaminant detection models across six scenarios, varying in training and testing distances (tunnel-5m, outdoor-10m, and outdoor-20m). Among the 1D approaches, kNN and SVM perform best, while other methods like Logistic Regression and Decision Tree remain below 0.5 F1-score. EffNetPoint demonstrates the effectiveness of 2D architectures, while 3D methods, including 3D-CNN, 3D-GCN (Graph Convolutional Network), and 3D-GAT (Graph Attention Network), perform consistently better in capturing spatial and contextual richness. The proposed 3D-GCN and 3D-GAT achieve the most stable performance across scenarios, leveraging graph-based attention mechanisms that surpass 3D-CNN in preserving spatial and contextual information. This highlights the importance of advanced 2D and 3D methods for reliable contaminant detection.

Training and testing data are crucial for evaluating model performance. Models trained and tested within the same location, such as outdoor-10m and outdoor-20m, achieve higher accuracy. However, GAT performs well even when training and testing are conducted across different datasets, such as tunnel-5m to outdoor-10m or outdoor-20m. Notably, training on outdoor 20 m and testing on tunnel 5 m still yields strong generalization, with GAT achieving an F1-score above 0.880.

In contaminant detection, both the F1 score and Recall are critical metrics, with Recall being particularly important for safety-critical systems like autonomous vehicles. Recall measures the model's ability to identify all contaminated cases, minimizing false negatives and detecting most contamination instances. This is essential to avoid hazardous

Table 2
The evaluation result of all cross-trains and test data scenarios for contaminant detection from baseline and the proposed method GAT. The numbers reported are Accuracy, Precision, Recall, and F1-score.

Train Data	Test Data	Scenario	Transformation-Algorithm	Acc	Pre	Rec	F1-score
tunnel-5m	outdoor-10m	1	1D-KNN	0.872	0.436	0.500	0.466
			1D-SVM	0.934	0.965	0.742	0.807
			2D-EffNetPoint	0.904	0.950	0.623	0.672
			3D-CNN	0.537	0.608	0.735	0.497
			3D-GCN	0.951	0.870	0.940	**0.900**
			3D-GAT	0.894	0.773	0.939	**0.820**
	outdoor-20m	2	1D-KNN	0.812	0.406	0.500	0.448
			1D-SVM	0.903	0.947	0.742	0.798
			2D-EffNetPoint	0.886	0.938	0.696	0.749
			3D-CNN	0.607	0.662	0.758	0.585
			3D-GCN	0.959	0.910	0.975	**0.938**
			3D-GAT	0.934	0.870	0.960	**0.904**
outdoor-10m	tunnel-5m	3	1D-KNN	0.556	0.603	0.749	0.504
			1D-SVM	0.271	0.568	0.588	0.270
			2D-EffNetPoint	0.938	0.871	0.805	**0.834**
			3D-CNN	0.937	0.967	0.727	0.795
			3D-GCN	0.892	0.758	0.939	0.808
			3D-GAT	0.909	0.780	0.949	**0.832**
	outdoor-20m	4	1D-KNN	0.962	0.962	0.912	0.934
			1D-SVM	0.902	0.828	0.938	0.864
			2D-EffNetPoint	0.915	0.865	0.850	0.857
			3D-CNN	0.947	0.890	0.967	**0.922**
			3D-GCN	0.912	0.841	0.946	0.877
			3D-GAT	0.968	0.927	0.980	**0.951**
outdoor-20m	tunnel-5m	5	1D-KNN	0.363	0.577	0.640	0.352
			1D-SVM	0.173	0.561	0.533	0.170
			2D-EffNetPoint	0.933	0.882	0.759	0.805
			3D-CNN	0.885	0.442	0.500	0.469
			3D-GCN	0.934	0.817	0.962	**0.869**
			3D-GAT	0.940	0.830	0.966	**0.880**
	outdoor-10m	6	1D-KNN	0.945	0.849	0.968	0.894
			1D-SVM	0.888	0.766	0.934	0.812
			2D-EffNetPoint	0.923	0.812	0.933	0.856
			3D-CNN	0.950	0.894	0.879	0.886
			3D-GCN	0.956	0.873	0.970	**0.913**
			3D-GAT	0.959	0.890	0.945	**0.915**

situations caused by undetected contaminants. However, balancing precision is necessary to prevent excessive false positives, which could degrade system performance. Table 2 compares the recall performance of various models for contaminant detection across six scenarios with different training and testing distances (tunnel-5m, outdoor-10m, and outdoor-20m). Among the models, 3D-GAT performs best with a mean recall of 0.92, excelling in capturing spatial and contextual relationships in point cloud data. EffNetPoint, a 2D model, performs well with an average recall of 0.85, while 1D models like Random Forest struggle, averaging 0.79 and showing poor generalization across varying distances. Training data significantly impacts performance, with models trained on outdoor-20m data yielding the highest recall scores. In Scenario 6, 3D-GCN achieves 0.970, and EffNetPoint scores 0.933, demonstrating that longer-range training data enhances contamination detection under diverse conditions. This highlights the importance of optimizing training datasets for robust contaminant detection. 3D-GAT outperforms other methods, excelling in F1 score and recall due to its ability to capture spatial relationships and contextual information in 3D point cloud data. While 2D-EfficientNet performs well for 2D transformations, its slightly lower recall makes 3D-GAT more suitable for high-priority tasks like contamination detection, where minimizing false negatives is crucial.

4.4. Contaminant detection confusion matrix

The confusion matrix for GAT in Fig. 8 shows high true positive and true negative values, with relatively few false negatives. False negatives, analyzed in the heatmap in Fig. 9, reveal that low-level contaminants, such as dust, foam, mudDrop, mudUniform, oil, salt, and water, are often misclassified as clean. These contaminants resemble clean conditions, making detection challenging. For instance, mudDrop, consisting of sparse mud particles, minimally affects the data and is likely classified as clean. Similarly, small quantities of contaminants like salt_high and mudUniform_mid also contribute to misclassifications. In Scenario 6, where the model is trained on outdoor-20m data and tested on outdoor-10m data, 266 false positives occurred, all involving clean covers made of transparent plastic. The protective cover introduced noise, behaving like a contaminant and causing some misclassifications. However, the classifier never falsely identified clean, uncovered LiDAR data as contaminated. This issue does not affect the validity of the experimental data or detection performance, as critical anomalies impairing object detection were correctly identified. This problem is irrelevant in real-world applications since production-ready LiDAR systems do not use additional protective covers.

4.5. Why graph embedding and GAT are robust for contaminant detection

Graph embeddings, visualized in PCA space for clarity (Fig. 10), effectively represent clean and contaminated classes across the tunnel-5m, outdoor-10m, and outdoor-20m datasets using distinct colors and markers. Clean data points form tight clusters, reflecting consistency, while contaminated points are more dispersed, highlighting their diverse and variable nature. Overlaps between classes, particularly across datasets,

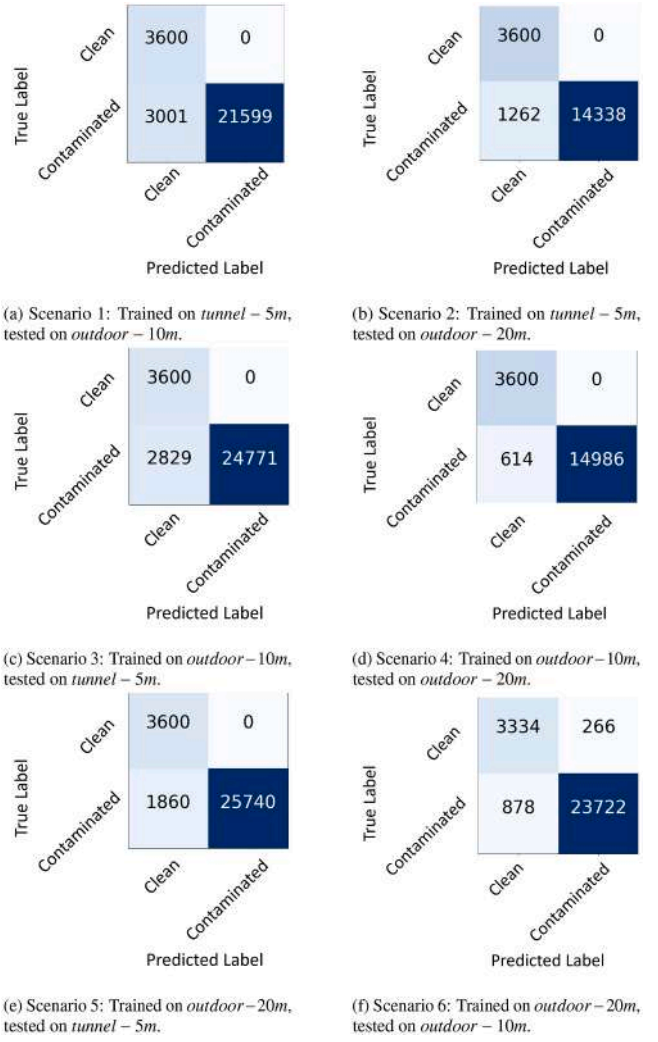


Fig. 8. Confusion matrix of the proposed contaminant detection using 3D-GAT in six scenarios training-testing dataset.

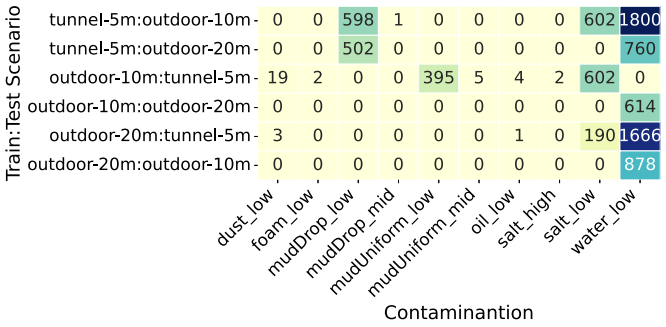


Fig. 9. Heatmap of False Negative contaminant detection of the proposed method, 3D-GAT, in all scenarios.

emphasize the challenges of classification. Graph embeddings provide a strong foundation for Graph Attention Networks (GATs) by capturing rich relational information. GATs leverage these embeddings to focus on meaningful relationships, enhancing classification efficiency. This approach excels in cross-dataset scenarios, enabling robust generalization and demonstrating the suitability of graph-based representations for contamination detection tasks.

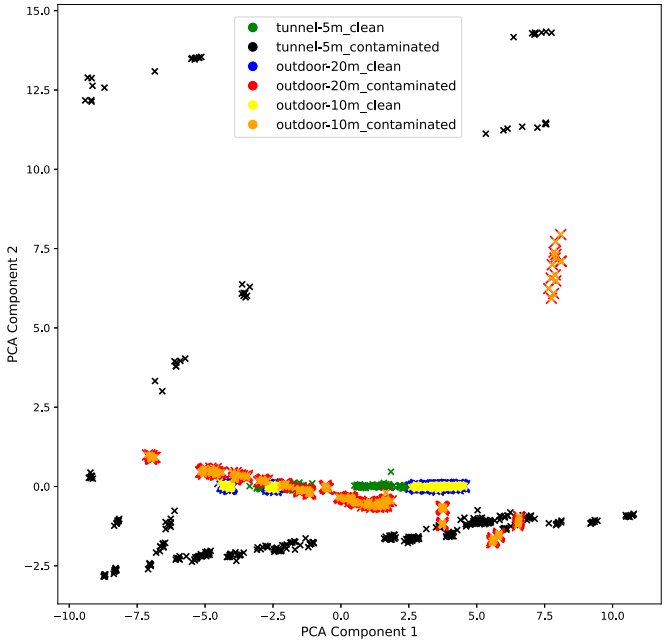


Fig. 10. Visualization of graph embedding for LiDAR contaminated dataset across three different environments.

5. Cost-Benefit analysis of contaminant detection on downstream task

5.1. Benchmarking object detection on corrupted point clouds

In this paper, we empirically evaluate detection outputs from the PointRCNN object detector (Shi, Wang, & Li, 2019), known for its robust performance and low relative corruption error (RCE) when tested on synthetic corruptions (Dong et al., 2023a). The model, trained on the KITTI dataset (Team, 2020). The object detector employed in our experiments was trained solely on clean LiDAR data, as only clean-labeled samples are available. Accordingly, evaluation on contaminated inputs is conducted without any retraining or fine-tuning, reflecting a realistic deployment scenario in which models trained under ideal conditions must remain resilient in the presence of sensor degradation.

To evaluate the impact of contamination, we measure the distance between the center point (X, Y) of the detected bounding box (BB) and the corresponding ground truth (GT) and the number of object should be detected. We evaluate Car detection as a important task for autonomous driving.

Fig. 11 categorizes the detection outcomes into four groups. For PointRCNN, a *Catastrophic Confidence Threshold* of 0.9 is applied to explicitly capture high-risk failure cases, characterized by incorrect detections made with exceptionally high confidence, which pose significant threats to downstream decision-making. Most frames fall under $Correct|High_Confidence$, as the dataset primarily contains clean sensor conditions, low-level contamination, and contaminants that minimally impact object detection performance. However, occurrences of $Correct|Low_Confidence$ and $Wrong|Low_Confidence$ in the tunnel-5m and outdoor-20m datasets indicate that contamination reduces detection confidence. Empirical evaluation reveals that the outdoor-10m dataset does not exhibit any instances of $Correct|Low_Confidence$ or $Wrong|Low_Confidence$. Notably, the 10m dataset was captured at the same location as the outdoor-20m dataset but at a shorter distance. This suggests that outdoor-10m represents an optimal range, as increasing the sensor-to-object distance and contamination severity naturally degrade PointRCNN's confidence and performance. This is especially pronounced when PointRCNN is trained exclusively on clean data, making

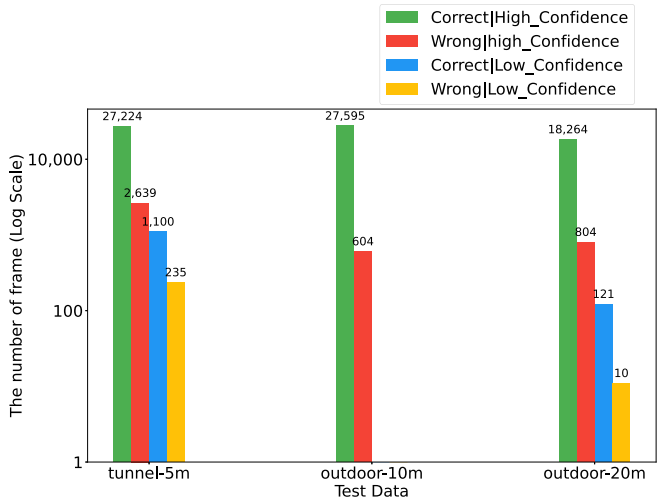


Fig. 11. Benchmarking object detection model in contaminated LiDAR point cloud.

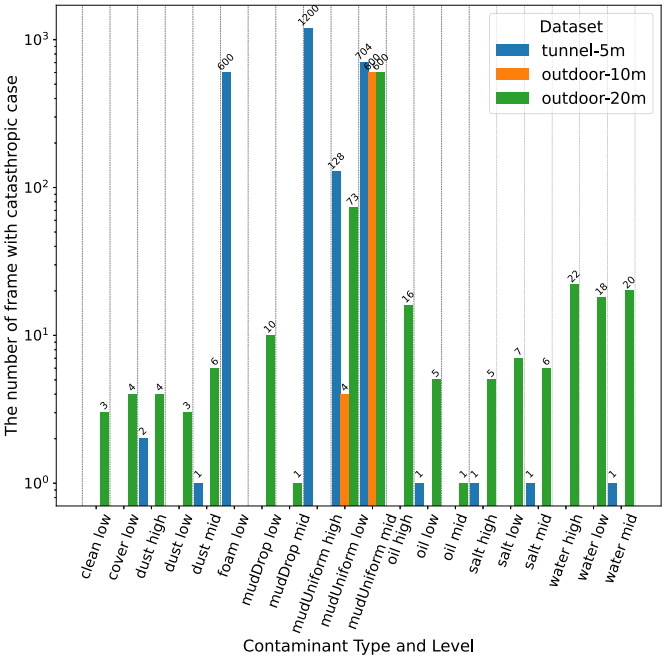


Fig. 12. The number of wrong detections with high confidence (Catastrophic mistake) over various contaminant types and levels across three different datasets.

it less adaptable to contaminated conditions. In the tunnel-5m dataset, captured in a narrow underground tunnel, denser point clouds amplify the effects of contamination, making its impact more evident compared to other datasets. It is crucial to note that contamination application is not uniform across datasets resulting in variability.

The most critical aspect of this benchmarking is the presence of catastrophic mistakes, indicated by $Wrong|High_Confidence$ cases is exist in total 2639, 604, and 804 frames in the tunnel-5m, outdoor-10m, and outdoor-20m datasets, respectively, and are filtered by the contaminant detection process. The $Correct|High_Confidence$ frames contribute to calculating system cost, while $Wrong|High_Confidence$ frames determine the benefit, as they indicates critical errors. Metrics such as $Correct|Low_Confidence$ and $Wrong|Low_Confidence$ are excluded from the cost-benefit analysis, as the object detection model automatically ignores low-confidence frames. Fig. 12 highlights catastrophic errors caused by different contaminants across datasets. Hazardous contaminants include foam, mudUniform, and oil across all datasets, along with water in the outdoor-20m dataset, where the sensor-to-object distance is greater compared to outdoor-10m within the same environment. In the tunnel-5m dataset, the effects of contaminants are noticeable with mudDrops and foam. Notably, foam contamination is only present in the tunnel-5m dataset and is not applied to outdoor-10m or outdoor-20m datasets.

5.2. Benefit of contaminant detection in reducing catastrophic mistakes

In this subsection, we quantify how many catastrophic mistakes can be prevented by performing contaminant detection prior to object detection. A single *Wrong | High Confidence* detection is sufficient for the entire frame to be counted as a catastrophic mistake. Consequently, the reported number represents a minimal or optimistic estimate of the actual safety risk, as multiple such mistakes may occur within a single frame. Based on this definition, and to align with the operational level of our proposed contaminant detection mechanism-which functions at the frame level-the analysis of catastrophic mistakes is also conducted at the same level of granularity.

To reflect the real-world safety impact, we compute the absolute number of catastrophic mistakes avoided by incorporating our frame-level contaminant detection mechanism, as shown in Table 3. Under this definition, our method successfully prevents 604 catastrophic mistakes, amounting to a 100 % reduction in high-risk cases for the scenario where contaminant detection is trained on the tunnel-5m dataset and tested on the outdoor-10m dataset. Across all scenarios, the contaminant detection approach achieves a reduction rate ranging from 97 % to 100 %, indicating that the proposed method can effectively eliminate up to all catastrophic mistakes, and at minimum, prevent the vast majority of them.

As a model-agnostic approach, contaminant detection is designed to be independent and standalone from any specific perception model, making it easily applicable to other object detection models. We evaluate its applicability on the SECOND object detector (Yan, Mao, & Li, 2018), which utilizes voxel-based representations and achieves the highest Average Precision (AP) for object detection under rain conditions (Dong et al., 2023b). Contaminant detection shows 100 % benefit in the test case using the tunnel-5m data. However, in the outdoor-10m and outdoor-20m datasets, no benefit was observed. This is primarily due to the high Catastrophic Confidence Threshold (0.9), which is deliberately conservative in justifying catastrophic mistakes. When the threshold is lowered to 0.7, we observe a significant increase in the number of catastrophic mistakes-up to 7641 cases in the outdoor-20m test set. In this case, contaminant detection achieves a 91 % reduction in catastrophic mistakes when trained on outdoor-10m data, and an 84 % reduction when trained on tunnel-5m data. Detailed experimental results on the benefits of contaminant detection for the SECOND model are provided in the appendix.

Table 3
Collision risk metric: Reduction of catastrophic mistakes in object detection pointRCNN under cross-scenario contamination.

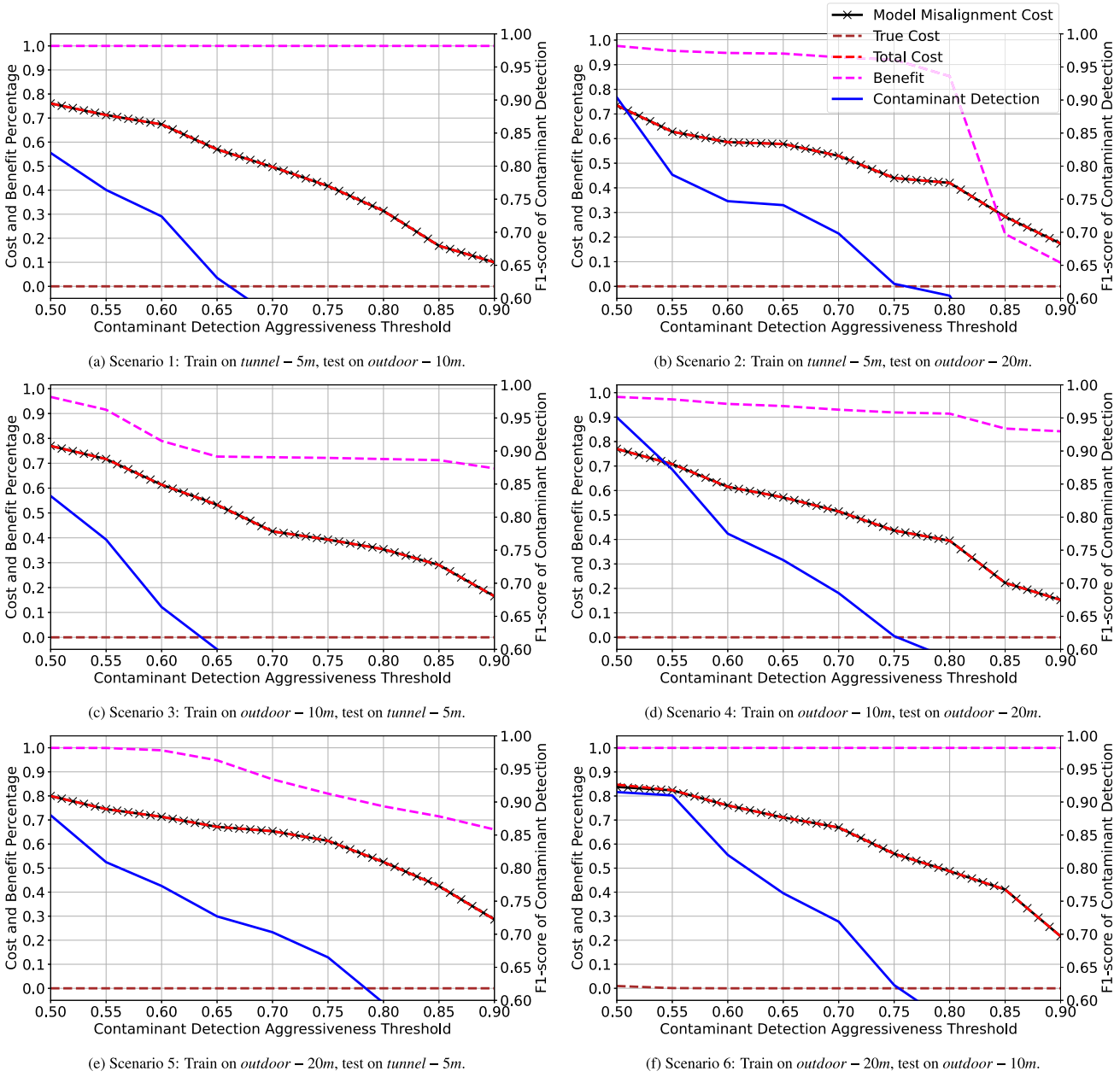
Train Data	Test Data	Catastrophic Mistakes in Object Detection	Catastrophic Mistakes prevent by Contaminant Detection	Benefit: Reduction Rate of Catastrophic Mistakes
tunnel-5m	outdoor-10m	604	604	100 %
	outdoor-20m	804	785	98 %
outdoor-10m	tunnel-5m	2639	2551	97 %
	outdoor-20m	804	790	98 %
outdoor-20m	tunnel-5m	2639	2638	100 %
	outdoor-10m	604	604	100 %

We highlight this frame-level benefit to demonstrate the practical utility of our approach, as these rare but severe errors are often obscured by average-case metrics-such as mean Average Precision (mAP) in object detection and confusion matrices in contaminant detection. Naturally, focusing solely on the benefits would be insufficient without also considering the associated costs. Therefore, assessing the cost-benefit trade-offs of incorporating contaminant detection into object detection systems is essential, and this is discussed in detail in the following subsection.

5.3. Assessing the cost-benefit trade-offs of contaminant detection in object detection

As a model-agnostic that ensembles contaminant detection with downstream processing, implementing contaminant detection must consider its benefit to downstream tasks by calculating the number of catastrophic failures it can prevent by accurately identifying contaminated frames. The benefit measures these mitigated failures, while the cost reflects correctly processed frames by the downstream process but discarded due to contaminant detection, which is unnecessary intervention. The model contaminant detection performance, evaluated by the F1-score, is a supplementary metric, with high-precision detection aiding operations like automated sensor cleaning. Additionally, adjusting the contaminant aggressiveness threshold allows for optimizing the cost-benefit rate. The minimum probability required by the contaminant detection model defines the aggressiveness threshold. We apply this probability threshold to the output of the contaminant class. It serves as an aggressiveness control-lower values result in more sensitive (aggressive) detection of contamination, while higher values lead to more selective (conservative) predictions, helping to reduce false positives.

The six plots in Fig. 13 explain the relationship between the Contaminant Detection Aggressiveness Threshold and key metrics: Model Misalignment Cost (black with X), True Cost (brown), Total Cost (red), Benefit (pink), and F1-score (blue). Cost and benefit values are represented



(a) Scenario 1: Train on *tunnel* – 5*m*, test on *outdoor* – 10*m*.

(b) Scenario 2: Train on *tunnel* – 5*m*, test on *outdoor* – 20*m*.

(c) Scenario 3: Train on *outdoor* – 10*m*, test on *tunnel* – 5*m*.

(d) Scenario 4: Train on *outdoor* – 10*m*, test on *outdoor* – 20*m*.

(e) Scenario 5: Train on *outdoor* – 20*m*, test on *tunnel* – 5*m*.

(f) Scenario 6: Train on *outdoor* – 20*m*, test on *outdoor* – 10*m*.

Fig. 13. Effect of Cost, Benefit, and Contaminant Detection due to changes in contaminant detection aggressiveness threshold across six scenarios.

on the left y-axis, ranging from 0 to 1, while the F1-score is plotted on the right y-axis, with a range of 0.6 to 1.0. Each scenario is based on different training and testing dataset combinations (tunnel-5m, outdoor-10m, and outdoor-20m) and demonstrates how threshold adjustments influence costs and benefits. Increasing the threshold preserves a high benefit while significantly reducing costs, though it also lowers the F1-score for contaminant detection. Despite this trade-off, the system effectively prevents catastrophic high-confidence errors, as reflected by benefit values approaching one across all scenarios, showcasing a balanced trade-off between cost and performance while minimizing critical errors.

Model agnostic related cost. The total cost comprises the Model Misalignment Cost and True Cost, with the True Cost consistently at 0, demonstrating the contaminant detection system's effectiveness in avoiding false detection of uncontaminated frames. The Model Misalignment Cost can be tuned in the downstream processing model; for this example, the object detection model train in the clean dataset can have a different Model Misalignment Cost from the model train in the contaminated dataset. As the True Cost remains zero, the total cost primarily depends on the Model Misalignment Cost. In Fig. 13, the Model Misalignment Cost (red line) aligns with the Total Cost (black line) in most scenarios, with a small True Cost contribution observed only in Scenario 6.

Contaminant detection aggressiveness setting. Adjusting the aggressiveness threshold for the contaminant optimizes the value of the cost-benefit by setting the minimum probability of classifying a frame as contaminated. A lower threshold increases sensitivity but raises costs, while a higher threshold reduces costs at the risk of lower sensitivity. The F1-score, displayed alongside cost and benefit, guides these adjustments, providing a balanced metric for decision-making. Tuning this threshold aligns the aggressiveness of the system with the desired trade-offs in cost, benefit, and detection performance trade-offs.

The Fig. 14 presents six subplots, each representing a different training and testing scenario for contaminant detection. The x-axis shows the contaminant detection aggressiveness threshold, ranging from 0.5 to 0.9. The left y-axis (red line) displays the Benefit/Cost ratio, where higher values indicate better cost efficiency. The right y-axis (blue line) illustrates the F1-score, a balance of precision and recall, with higher values reflecting better detection performance. Each subplot corresponds to specific training and testing conditions: scenarios 1 and 2 train on a tunnel-5m dataset and test on outdoor-10m and outdoor-20m datasets; scenarios 3 and 4 train on a outdoor-10m dataset and test on tunnel-5m and outdoor-20m datasets, and scenarios 5 and 6 train on a outdoor-20m dataset and test on tunnel-5m and outdoor-10m datasets, respectively. Across all scenarios, the optimal threshold range lies between 0.6 and 0.65, balancing a high F1-score (preferably above 0.75) and improving the Benefit/Cost ratio. For scenarios trained on a tunnel-5m dataset (scenarios 1 and 2) and a outdoor-20m dataset (scenarios 5 and 6), the threshold leans toward 0.65. For scenarios trained on a outdoor-10m dataset (scenarios 3 and 4), the threshold is closer to 0.6. This range ensures effective contaminant detection by maintaining high detection accuracy while optimizing benefit and cost efficiency. However, it is crucial to note that the contaminant detection aggressiveness threshold associated with capability detection is excessively stringent at a value greater than 0.8. This may result in the non-detection of several contaminants, potentially leading to a reduction in benefit, as observed in scenario 2. Utilizing this threshold enables selecting an optimal model-agnostic contaminant detection approach tailored to individual criteria. The F1-score is a metric of lesser importance than the primary focus: the impact of contaminant detection on downstream tasks, particularly object detection. This impact is most accurately reflected in the benefit parameter, highlighting the prevention of catastrophic mistakes.

The aggressiveness threshold in the proposed system controls how sensitively contaminated LiDAR frames are flagged as unreliable, balancing false positives and false negatives. It is configurable to adapt to different driving contexts, sensor configurations, and environmental conditions. In high-risk scenarios (e.g., highway driving), a lower threshold ensures safety by aggressively filtering uncertain data, while in low-risk settings (e.g., parking), a higher threshold allows more tolerance. The threshold can also be adjusted based on sensor redundancy and dynamically adapted using a continuous contamination confidence score. Overall, this mechanism enables a flexible, context-aware approach to maintain perception reliability across varying conditions.

6. Contaminant detection in dynamic environment

The goal of this evaluation is to demonstrate the generalization ability of the proposed contaminant detection method across various LiDAR types and real-world environments, including unseen and dynamic conditions.

6.1. Different LiDAR sensor type

Testing on data from multiple datasets allows us to evaluate the method's ability to generalize across different LiDAR sensors, including Robosense and Velodyne, which are widely used in autonomous driving. Table 4 provides a comparative overview of several LiDAR-based datasets used for testing contaminant detection. The KITTI dataset utilizes the Velodyne HDL-64E sensor, featuring 64 channels with a range of 120 m and generating 1.4 million points per second. Similarly, the NuScenes dataset employs the Velodyne HDL-32E, a lower-resolution sensor with 32 channels, a range of 100 m, and a slightly lower output of 1.3 million points per second. In contrast, our self-collected datasets, LIDAROC_road_clean and LIDAROC_road_mud, make use of the Robosense RS-Ruby sensor, which offers significantly higher resolution with 128 channels, an extended range of 250 m, and a much denser point cloud at 4 million points per second. These differences highlight the higher fidelity and broader coverage of the testing datasets, which are beneficial for evaluating the proposed contaminant detection method and its compatibility with various perception tasks under different LiDAR types.

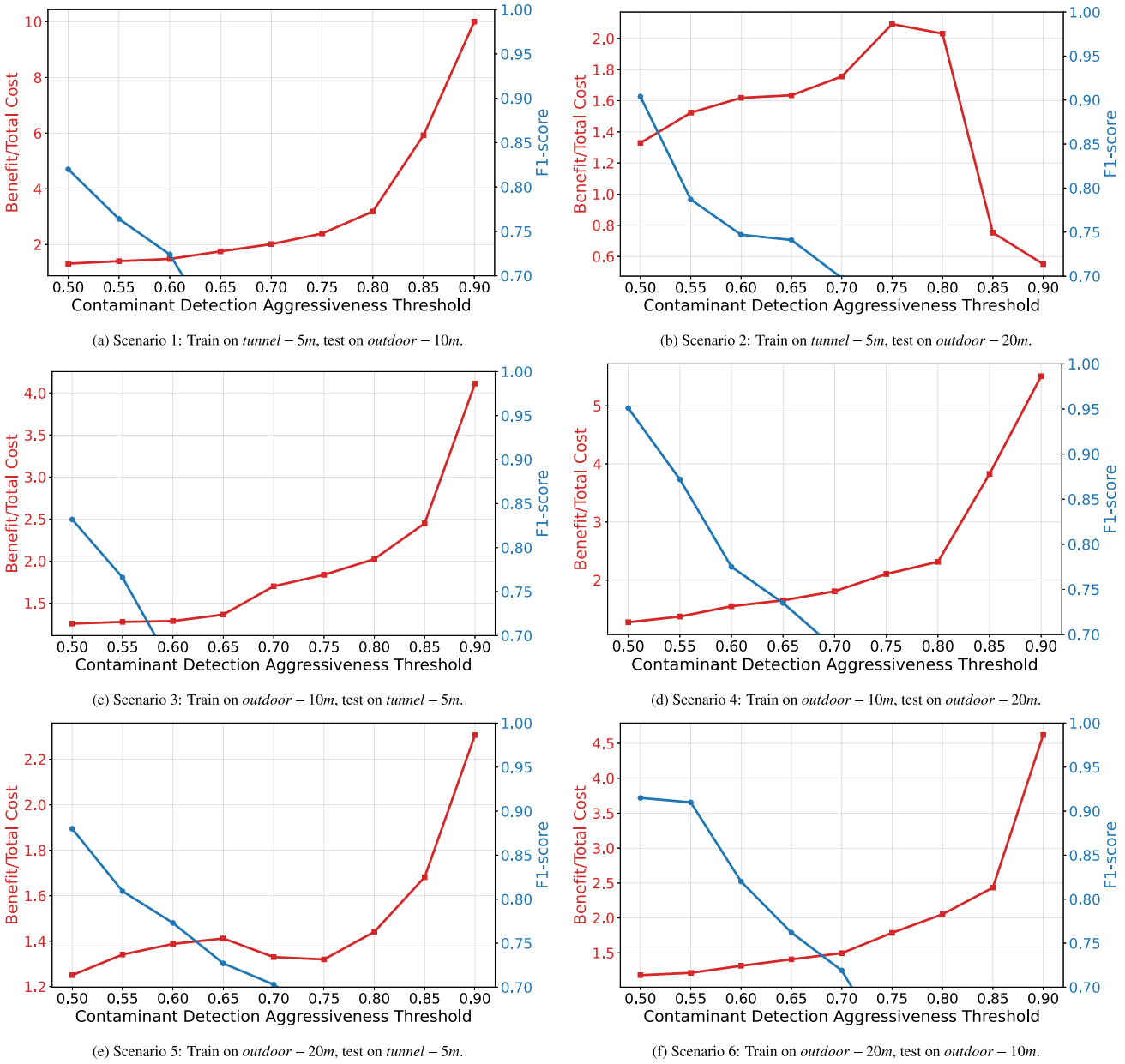
6.2. Unseen and dynamic environment point cloud

The proposed method is evaluated on data with the following characteristics: (1) unseen real-world environments, (2) diverse location types, including urban and highway scenarios, and (3) different times of day, covering both daytime and nighttime conditions. Furthermore, we investigate the method's robustness to various types of contamination, including mixed scenarios (e.g., mud combined with rain).

To evaluate the generalizability of the contaminant detection model in dynamic and unseen test data that differ from training data, we include public datasets such as KITTI (Geiger, Lenz, Stiller, & Urtasun, 2013) and nuScenes (Caesar et al., 2020) for comparison purposes. Although KITTI and nuScenes were not specifically designed with LiDAR contamination in mind, they are used only to evaluate performance under clean conditions. KITTI data are mostly collected during the day, whereas nuScenes provides a more diverse temporal distribution, including both daytime and nighttime scenes.

Table 4
Dataset and LiDAR sensor for testing LiDAR contaminant detection.

Dataset	LiDAR Sensor	Channel	Range	Point per second
KITTI	Velodyne HDL-64E	64	120 m	1.4 M pts/s
nuScenes	Velodyne HDL32E	32	100 m	1.3 M pts/s
LIDAROC_road_clean	Robosense (RS-Ruby)	128	250 m	4 M pts/s
LIDAROC_road_mud	Robosense (RS-Ruby)	128	250 m	4 M pts/s



(a) Scenario 1: Train on *tunnel* − 5*m*, test on *outdoor* − 10*m*.

(b) Scenario 2: Train on *tunnel* − 5*m*, test on *outdoor* − 20*m*.

(c) Scenario 3: Train on *outdoor* − 10*m*, test on *tunnel* − 5*m*.

(d) Scenario 4: Train on *outdoor* − 10*m*, test on *outdoor* − 20*m*.

(e) Scenario 5: Train on *outdoor* − 20*m*, test on *tunnel* − 5*m*.

(f) Scenario 6: Train on *outdoor* − 20*m*, test on *outdoor* − 10*m*.

Fig. 14. Trade-off of Benefit/Cost over F1-Score in contaminant detection aggressiveness threshold across six scenarios.

To support this evaluation of sensor contamination, we also introduce our self-collected data set, namely the LIDAROC_road data set, an extension of the original LIDAROC data set, which includes dynamic LiDAR recordings. LIDAROC remains the only dataset that features physical and realistic LiDAR cover contamination. Fig. 15 show our setting for mud contamination mixed with water, as the dataset was collected during light rain. Water droplets can be observed on the LiDAR platform, indicating the presence of rain during acquisition. Figs. 16–19 provide visualizations of the clean and contaminated conditions, respectively. Additionally, the top-left corner of the images shows varying road conditions at the time of recording, ranging from heavy traffic to low traffic density. This highlights the high variability of the test data set in terms of location and traffic flow.

6.3. Contaminant detection result in dynamic point cloud

To test the proposed contaminant detection model, we select the best-performing method, the 3D-GAT model-based on the static dataset results shown in Table 2. For testing on dynamic data, we use the model trained on the tunnel-5m dataset, as it represents the most distinct environment in comparison to road scenes. In terms of location, all dynamic datasets cover a wide range of scenarios, including urban areas, rural roads, flyovers, and seaports. For the KITTI dataset, we use the official testing set and select 1700 point clouds as samples. From the nuScenes mini set, we extract 1125 point clouds from daytime scenes and 121 point clouds from nighttime scenes. For the LIDAROC_road dataset, we include 1700 point clouds encompassing both clean and contaminated conditions. Detailed dataset settings and corresponding results are presented in Table 5.

The experimental results on unseen and dynamic data are shown in Table 5. Based on these results, we can conclude that the 3D-GAT model achieves high accuracy, precision, recall, and F1-score for detecting clean LiDAR data across the KITTI, nuScenes, and LIDAROC_road_clean datasets. For contamination detection, the proposed 3D-GAT model also performs well, achieving scores of 0.951, 1.000, 0.951, and 0.975 for accuracy, precision, recall, and F1-score, respectively. These results

Fig. 15. Contamination setup with mixed mud under rainy conditions for LIDAROC_road_mud.

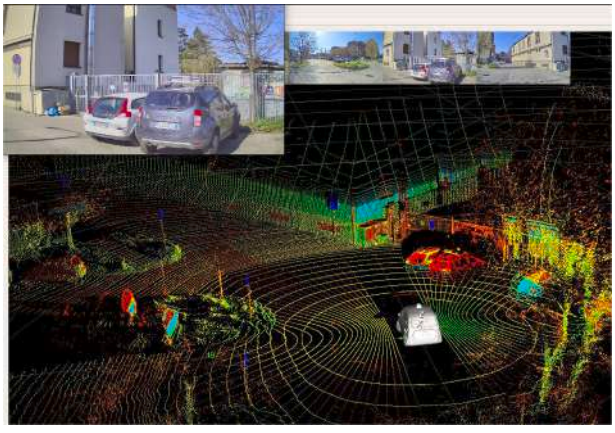


Fig. 16. Point cloud of LIDAROC_road_clean urban area.

indicate that a single model trained on a static dataset can generalize effectively to unseen dynamic datasets in both clean and contaminated conditions.

7. Deployment

We evaluate the contaminant detection model's running time on edge devices. The NVIDIA Jetson AGX Xavier, designed for real-time LiDAR data processing at up to 70 MB/s (Liu et al., 2021), was selected

Fig. 17. Point cloud of LIDAROC_road_clean in intersection road.

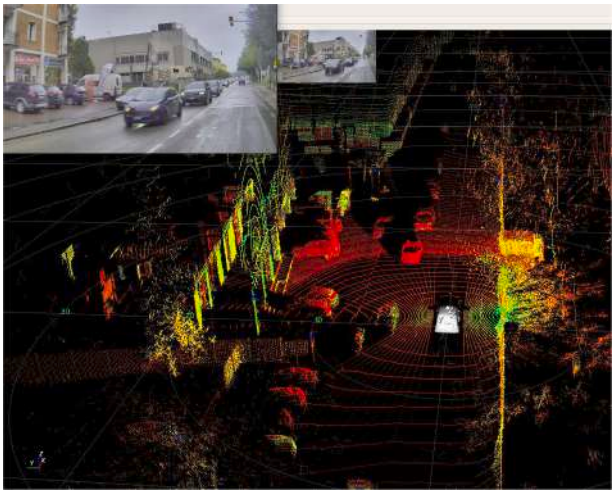


Fig. 18. Point cloud of LIDAROC_road_mud with light rain in urban area.

as the platform. Equipped with two Xavier cores, an 8-core CPU, and a 512-core Volta GPU, it delivers up to 320 trillion operations per second, make it suitable for edge computing applications. This paper presents the first comparison of contaminant detection on an edge platform and compares it with a server-based platform. The server, equipped with an AMD EPYC 74F3 CPU (24 cores, 48 threads, 3.2 GHz), an NVIDIA GeForce RTX 3090 GPU (24 GB memory), and 250 GB DDR4 RAM, was tested against the edge platform. We present the evaluation of point-cloud transformation time, inference time, and the total process from transformation to contaminant detection on both setups, along with CPU and GPU utilization.

Point cloud transformation time. Fig. 20 compares the average transformation time of point cloud data across different methods-1D, 2D, 3D Voxel, and Graph-on both a server and the Jetson AGX Xavier. The x-axis represents transformation methods, the y-axis denotes datasets (tunnel-5m, outdoor-10m, outdoor-20m), and the z-axis indicates

Table 5
Testing result of LiDAR contaminant detection in unseen dynamic dataset.

Dataset	Location	Time	LiDAR Condition	nSample	Acc	Pre	Rec	F1-score
KITTI	Urban and Rural Area	Day	Clean	1700	1.000	1.000	1.000	1.000
nuScenes	Urban and Seaport	Day	Clean	1125	0.963	1.000	0.963	0.981
nuScenes	Urban and Seaport	Night	Clean	121	1.000	1.000	1.000	1.000
LIDAROC_road_clean	Urban Area	Day	Clean	1700	1.000	1.000	1.000	1.000
LIDAROC_road_mud	Urban and Flyover road	Day	Mud and Light Rain	1700	0.951	1.000	0.951	0.975

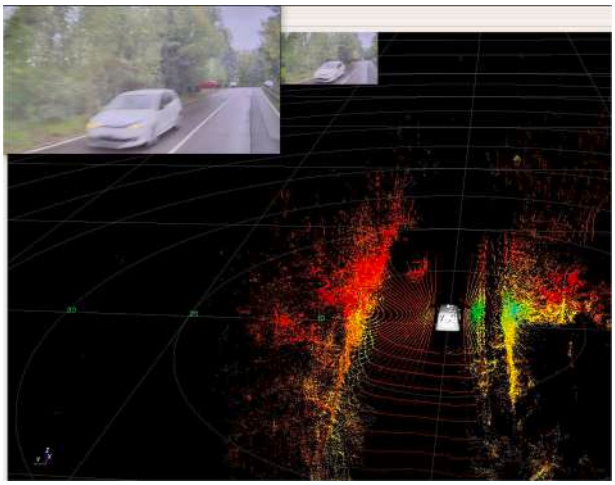


Fig. 19. Point cloud of LIDAROC_road_mud in sparse traffic.

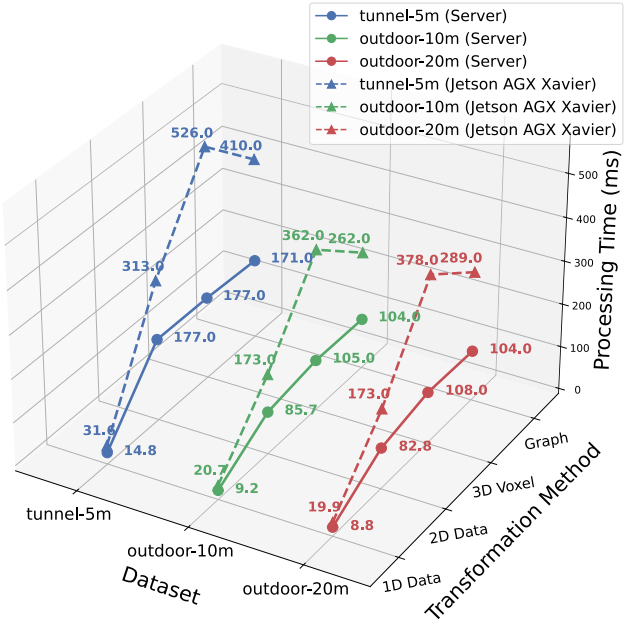


Fig. 20. Point cloud transformation time for contaminant detection.

processing time in milliseconds. The results show minimal time differences between methods on the server. On the Jetson AGX, processing times range from 19.9 ms for the 1D method to 526 ms for 3D voxel, while graph-based transformation takes approximately 200-410 ms, remaining feasible for edge computing. On a higher-performance server, graph transformation achieves faster processing times, ranging from 100 to 171 ms.

The Jetson AGX processes all transformation methods ranging from around 200 ms - 400 ms, with 3D voxel and graph-based transformations slightly slower than 2D methods but offering richer spatial context. The tunnel-5m dataset processes slower than outdoor-10m and outdoor-20m due to higher point density, with approximately 142 866, 70 840, and 69 909 points, respectively, and increased object density from closer walls. While the server outperforms the Jetson AGX Xavier in speed, all transformations run efficiently on the Jetson, confirming its suitability for edge computing. Graph-based transformations remain practical, with processing times comparable to 2D methods, demonstrating their applicability for scenarios requiring enhanced structural understanding.

Comparison of contaminant detection inference performance on edge. The comparison between CPU and GPU performance highlights the GPU's superior efficiency for contaminant detection inference, particularly on edge devices. Fig. 21 presents inference times for various algorithms on the NVIDIA Jetson platform, comparing CPU and GPU configurations across different test distances: tunnel-5m, outdoor-10m, and outdoor-20m. Processing times, measured in milliseconds, are evaluated for 2D-EfficientNet, 3D-CNN, 3D-GCN, and 3D-GAT. The 1D method is excluded from consideration due to its failure in contaminant detection, as detailed in Table 2.

The GPU consistently delivers significantly faster inference times across all algorithms. As shown in Fig. 21, the CPU processes most methods in under 200 ms, with 2D-EfficientNet being the slowest. The GPU accelerates computationally intensive methods like 2D-EfficientNet and 3D-CNN, reducing inference times to a range of 1-5 ms, as these methods benefit from GPU-optimized convolution operations.

For 3D-GCN and 3D-GAT, CPU and GPU performance show negligible differences, achieving runtimes around 5 ms across all datasets. With high accuracy and fast processing, these graph-based methods are well-suited for contaminant detection on edge devices. Additionally, their consistent processing t imes across datasets reinforce their reliability in varying conditions.

End-to-end contaminant detection time. Finally, this paper evaluates the end-to-end contaminant detection process, encompassing point cloud transformation and model inference times. Performance is assessed GPU configurations, comparing server and edge devices. Fig. 22 show that

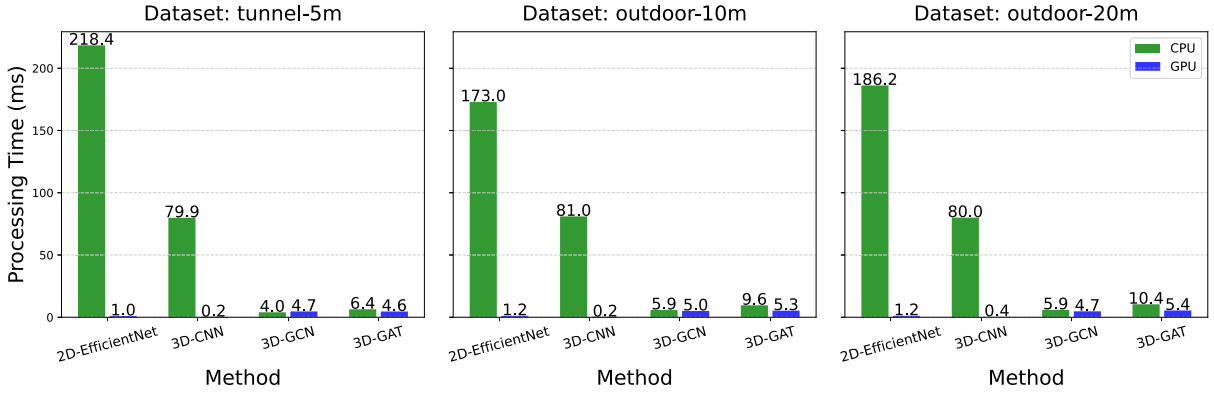


Fig. 21. Inference time comparison of contaminant detection models on NVIDIA Jetson AGX Xavier using GPU and CPU.

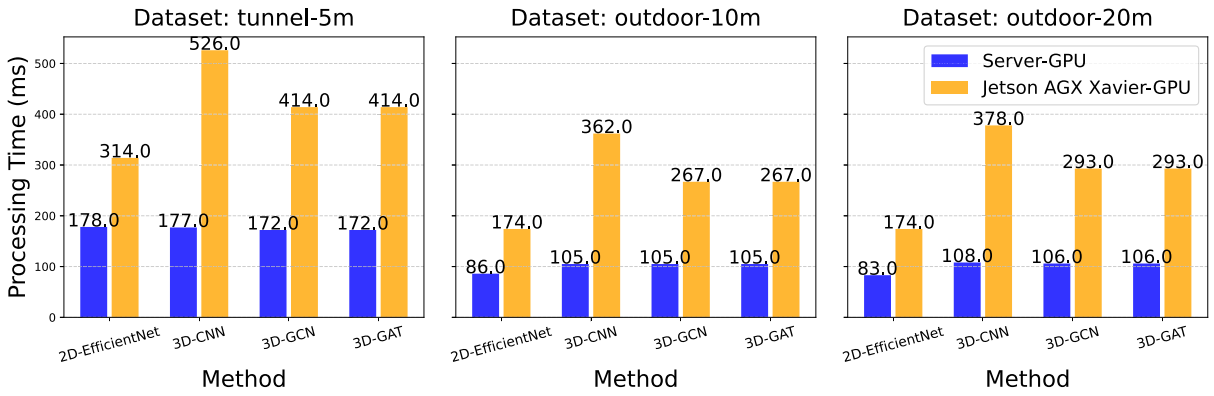


Fig. 22. Comparison of total contaminant detection time, including point cloud transformation and inference, across several models running on GPUs in a server and the Jetson AGX Xavier.

servers consistently outperform Jetson in execution speed across all algorithms. However, Jetson provides sufficient performance for real-time applications, even for computationally intensive models like 3D-GAT and 2D-EfficientNet.

The proposed 3D-GAT demonstrates competitive processing times across all methods, as shown in Fig. 22. On a server platform, it processes frames in 100-200ms. In edge computing on the Jetson platform, the denses dataset (tunnel-5m) requires up to 400ms due to higher point cloud density. For outdoor datasets with longer distances, such as outdoor-10m and outdoor-20m, processing times for GAT range from 267-414ms. These results were obtained without specific optimizations, suggesting that further reductions in processing time are possible, particularly by optimizing the implementation of point cloud transformation into graph representation. The results demonstrate that 3D-GAT, despite its slightly higher computational complexity, provides superior 3D spatial analysis for contaminant detection compared to baseline approaches. While the server achieves greater efficiency, Jetson's performance reliably supports real-time applications, making it suitable for edge deployment. These findings highlight the trade-off between computational complexity and execution speed, with both platforms offering practical solutions for real-world use cases.

8. Conclusion

This study introduces a model-agnostic safety system integrated with the perception stack. The safety system incorporates LiDAR contaminant detection using graph representations and Graph Attention Networks (GAT) for robust identification. Extensive experiments on unseen, dynamic environment datasets of contaminated point cloud frames reveal the adverse impact of contaminants on object detection. Our method achieves an F1-score of 0.975 for contamination detection in unseen real-world scenarios across varying locations, times, and LiDAR types, demonstrating high sensitivity. For model-agnostic evaluation, the proposed approach is applicable to both point-based and voxel-based object detection models, achieving a minimum catastrophic mistake reduction rate of 97 %. It effectively minimizes critical errors while maintaining low operational costs through an adjustable aggressiveness threshold. Deployed on the NVIDIA Jetson AGX Xavier, the system supports real-time performance with inference times around 5 milliseconds and complete processing in approximately 200 milliseconds. This approach significantly enhances sensor data reliability and represents a key advancement toward robust, resilient, and reliable autonomous vehicle perception systems.

CRediT authorship contribution statement

Grafika Jati: Conceptualization, Methodology, Data curation; **Martin Molan:** Methodology, Data curation, Writing – original draft; **Francesco Barchi:** Visualization, Investigation; **Andrea Bartolini:** Supervision; **Giuseppe Mercurio:** Validation, Funding acquisition; **Andrea Acquaviva:** Supervision, Writing – review & editing.

Data availability

All data and code for ANZIL will be publicly available at: https://gitlab.com/ecs-lab/anzil and the dataset used is partly derived from LIDAROC: https://gitlab.com/ecs-lab/lidaroc.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

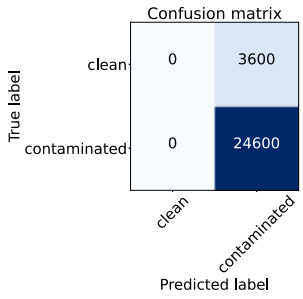
This research is partly supported by EU NRRP (PNRR) (g.a. DM 352/2022). This work is partly supported by SERICS - SEcurity and Rights in the CyberSpace - PE00000014 - J33C22002810001 funded by the European Union - NextGenerationEU - MUR Directorial Decree n. 1556 of 11 October 2022. This work is partly supported by EdgeAI KDT-JU European project (101097300). We would like to thank FEV for their support and for providing the hardware used for data acquisition.

Appendix A. Contaminant detection result and discussion

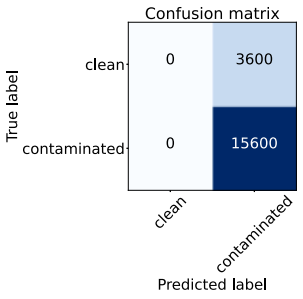
A.1. Contaminant detection confusion matrix

Confusion matrix of the baseline contaminant detection using 1D-KNN, 1D-SVM, 2D-EfficientNet, 3D-CNN, and 3D-GCN in six scenarios training-testing dataset.

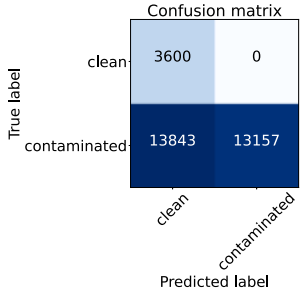
Figs. A.23–A.27 and Tables B.6–C.14.



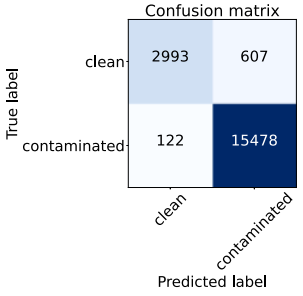
(a) Scenario 1: Trained on *tunnel* − 5*m*, tested on *outdoor* − 10*m*.



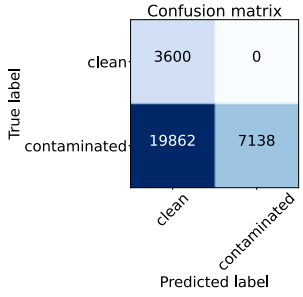
(b) Scenario 2: Trained on *tunnel* − 5*m*, tested on *outdoor* − 20*m*.



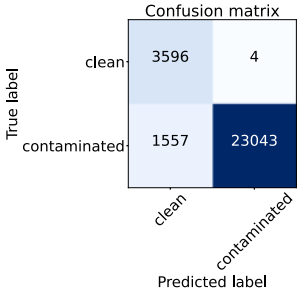
(c) Scenario 3: Trained on *outdoor* − 10*m*, tested on *tunnel* − 5*m*.



(d) Scenario 4: Trained on *outdoor* − 10*m*, tested on *outdoor* − 20*m*.

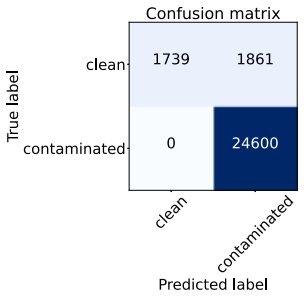


(e) Scenario 5: Trained on *outdoor* − 20*m*, tested on *tunnel* − 5*m*.

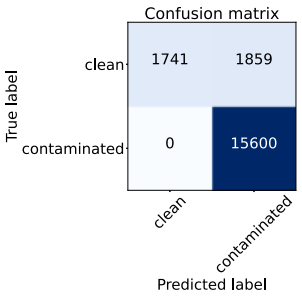


(f) Scenario 6: Trained on *outdoor* − 20*m*, tested on *outdoor* − 10*m*.

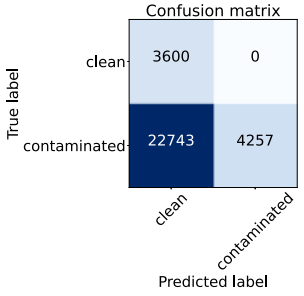
Fig. A.23. Confusion matrix of the proposed contaminant detection using 1D-KNN in six scenarios training-testing dataset.



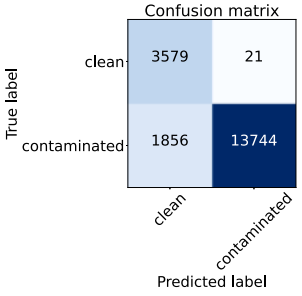
(a) Scenario 1: Trained on *tunnel* − 5*m*, tested on *outdoor* − 10*m*.



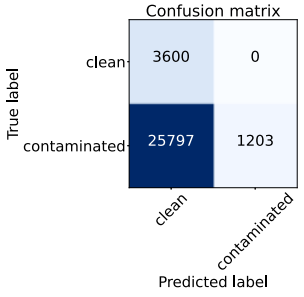
(b) Scenario 2: Trained on *tunnel* − 5*m*, tested on *outdoor* − 20*m*.



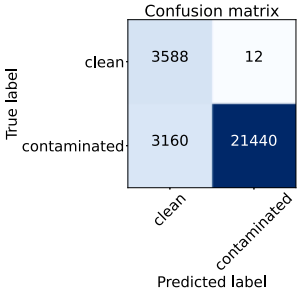
(c) Scenario 3: Trained on *outdoor* − 10*m*, tested on *tunnel* − 5*m*.



(d) Scenario 4: Trained on *outdoor* − 10*m*, tested on *outdoor* − 20*m*.

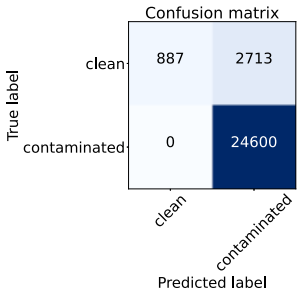


(e) Scenario 5: Trained on *outdoor* − 20*m*, tested on *tunnel* − 5*m*.

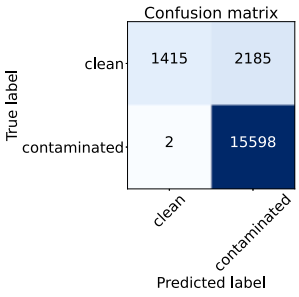


(f) Scenario 6: Trained on *outdoor* − 20*m*, tested on *outdoor* − 10*m*.

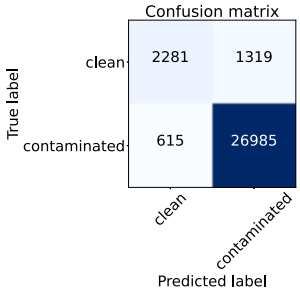
Fig. A.24. Confusion matrix of the proposed contaminant detection using 1D-SVM in six scenarios training-testing dataset.



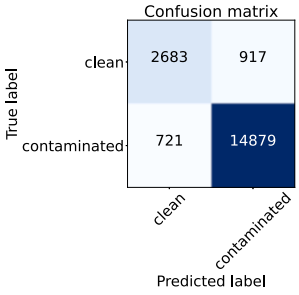
(a) Scenario 1: Trained on *tunnel* − 5*m*, tested on *outdoor* − 10*m*.



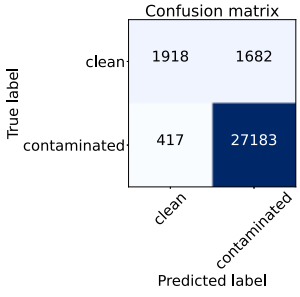
(b) Scenario 2: Trained on *tunnel* − 5*m*, tested on *outdoor* − 20*m*.



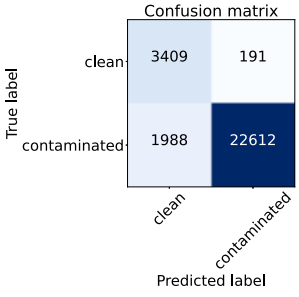
(c) Scenario 3: Trained on *outdoor*−10*m*, tested on *tunnel* − 5*m*.



(d) Scenario 4: Trained on *outdoor*−10*m*, tested on *outdoor* − 20*m*.

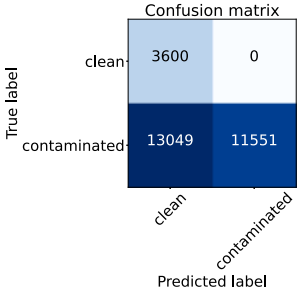


(e) Scenario 5: Trained on *outdoor*−20*m*, tested on *tunnel* − 5*m*.

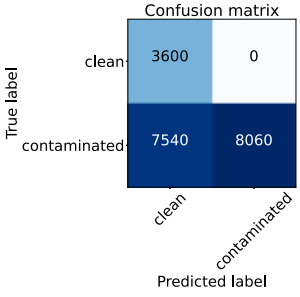


(f) Scenario 6: Trained on *outdoor*−20*m*, tested on *outdoor* − 10*m*.

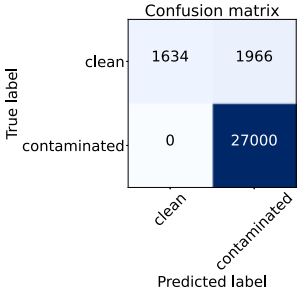
Fig. A.25. Confusion matrix of the proposed contaminant detection using 2D-EfficientNet in six scenarios training-testing dataset.



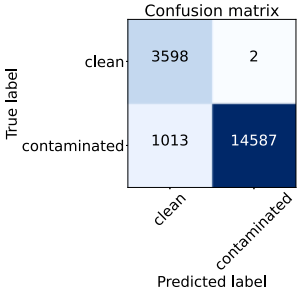
(a) Scenario 1: Trained on *tunnel* − 5*m*, tested on *outdoor* − 10*m*.



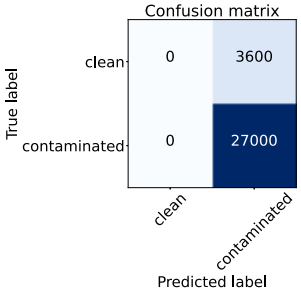
(b) Scenario 2: Trained on *tunnel* − 5*m*, tested on *outdoor* − 20*m*.



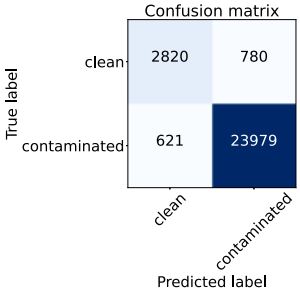
(c) Scenario 3: Trained on *outdoor*−10*m*, tested on *tunnel* − 5*m*.



(d) Scenario 4: Trained on *outdoor*−10*m*, tested on *outdoor* − 20*m*.

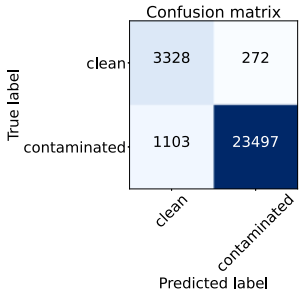


(e) Scenario 5: Trained on *outdoor*−20*m*, tested on *tunnel* − 5*m*.

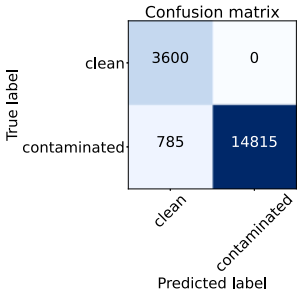


(f) Scenario 6: Trained on *outdoor*−20*m*, tested on *outdoor* − 10*m*.

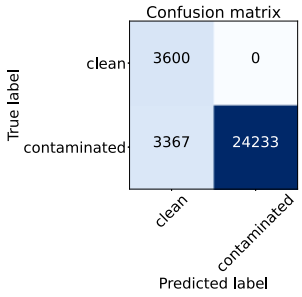
Fig. A.26. Confusion matrix of the proposed contaminant detection using 3D-CNN in six scenarios training-testing dataset.



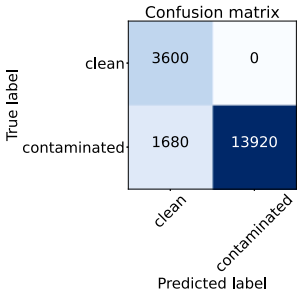
(a) Scenario 1: Trained on *tunnel* − 5*m*, tested on *outdoor* − 10*m*.



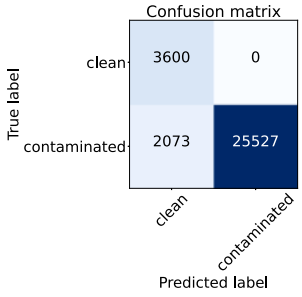
(b) Scenario 2: Trained on *tunnel* − 5*m*, tested on *outdoor* − 20*m*.



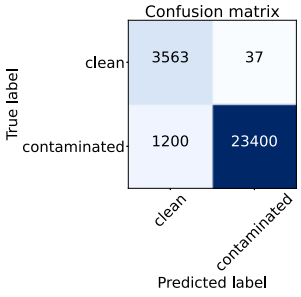
(c) Scenario 3: Trained on *outdoor* − 10*m*, tested on *tunnel* − 5*m*.



(d) Scenario 4: Trained on *outdoor* − 10*m*, tested on *outdoor* − 20*m*.



(e) Scenario 5: Trained on *outdoor* − 20*m*, tested on *tunnel* − 5*m*.



(f) Scenario 6: Trained on *outdoor* − 20*m*, tested on *outdoor* − 10*m*.

Fig. A.27. Confusion matrix of the proposed contaminant detection using 3D-GCN in six scenarios training-testing dataset.

Appendix B. Cost-benefit analysis of contaminant detection on downstream processing

Table B.6
Collision risk metric: Reduction of Catastrophic Mistakes in Object Detection PointRCNN under Cross-Scenario Contamination. Catastrophic Confidence Threshold is of 0.9.

Train Data	Test Data	Catastrophic Mistakes in Object Detection (case)	Catastrophic Mistakes prevent by Contaminant Detection (case)	Benefit: Reduction Rate of Catastrophic Mistakes
outdoor-20m	outdoor-10m	604	604	100 %
	tunnel-5m	2639	2638	100 %
outdoor-10m	outdoor-20m	804	790	98 %
	tunnel-5m	2639	2551	97 %
tunnel-5m	outdoor-10m	604	604	100 %
	outdoor-20m	804	785	98 %

Table B.7
Collision risk metric: Reduction of catastrophic mistakes in object detection pointRCNN under cross-scenario contamination. Catastrophic confidence threshold is of 0.7.

Train Data	Test Data	Catastrophic Mistakes in Object Detection (case)	Catastrophic Mistakes prevent by Contaminant Detection (case)	Benefit: Reduction Rate of Catastrophic Mistakes
outdoor-20m	outdoor-10m	629	629	100 %
	tunnel-5m	2723	2719	100 %
outdoor-10m	outdoor-20m	1681	1583	94 %
	tunnel-5m	2723	2608	96 %
tunnel-5m	outdoor-10m	629	629	100 %
	outdoor-20m	1681	1557	93 %

Table B.8
Cost of impementing contaminant detecion in PointRCNN. Catastrophic Confidence Threshold is of 0.9.

Train Data	Test Data	MisAlign-ment Cost	True Cost	Total Cost
outdoor-20m	outdoor-10m	0.84	0.01	0.85
	tunnel-5m	0.80	0.00	0.80
outdoor-10m	outdoor-20m	0.77	0.00	0.77
	tunnel-5m	0.77	0.00	0.77
tunnel-5m	outdoor-10m	0.76	0.00	0.76
	outdoor-20m	0.73	0.00	0.73

Table B.9
Cost of impementing contaminant detecion in PointRCNN. Catastrophic Confidence Threshold is of 0.7.

Train Data	Test Data	MisAlignment Cost	True Cost	Total Cost
outdoor-20m	outdoor-10m	0.84	0.01	0.85
	tunnel-5m	0.80	0.00	0.80
outdoor-10m	outdoor-20m	0.76	0.00	0.76
	tunnel-5m	0.77	0.00	0.77
tunnel-5m	outdoor-10m	0.76	0.00	0.76
	outdoor-20m	0.73	0.00	0.73

Table B.10
Collision Risk Metric: Reduction of Catastrophic Mistakes in Object Detection SECOND under Cross-Scenario Contamination. Catastrophic Confidence Threshold is of 0.9.

Train Data	Test Data	Catastrophic Mistakes in Object Detection (case)	Catastrophic Mistakes prevent by Contaminant Detection (case)	Benefit: Reduction Rate of Catastrophic Mistakes
outdoor-20m	outdoor-10m	0	0	0 %
	tunnel-5m	1	1	100 %
outdoor-10m	outdoor-20m	0	0	0 %
	tunnel-5m	1	0	0 %
tunnel-5m	outdoor-10m	0	0	0 %
	outdoor-20m	0	0	0 %

Table B.11
Collision Risk Metric: Reduction of Catastrophic Mistakes in Object Detection SECOND under Cross-Scenario Contamination. Catastrophic Confidence Threshold is of 0.7.

Train Data	Test Data	Catastrophic Mistakes in Object Detection (case)	Catastrophic Mistakes prevent by Contaminant Detection (case)	Benefit: Reduction Rate of Catastrophic Mistakes
outdoor-20m	outdoor-10m	1	1	100 %
	tunnel-5m	1	1	100 %
outdoor-10m	outdoor-20m	7641	6967	0.91 %
	tunnel-5m	1	0	0 %
tunnel-5m	outdoor-10m	1	1	100 %
	outdoor-20m	7641	6465	0.85 %

Table B.12
Cost of impementing contaminant detecion in SECOND. Catastrophic Confidence Threshold is of 0.9.

Train Data	Test Data	MisAlignment Cost	True Cost	Total Cost
outdoor-20m	outdoor-10m	0.84	0.01	0.85
	tunnel-5m	0.79	0.00	0.79
outdoor-10m	outdoor-20m	0.75	0.00	0.75
	tunnel-5m	0.77	0.00	0.77
tunnel-5m	outdoor-10m	0.76	0.00	0.76
	outdoor-20m	0.72	0.00	0.72

Table B.13
Cost of impementing contaminant detecion in SECOND. Catastrophic Confidence Threshold is of 0.7.

Train Data	Test Data	MisAlignment Cost	True Cost	Total Cost
outdoor-20m	outdoor-10m	0.84	0.01	0.85
	tunnel-5m	0.80	0.00	0.80
outdoor-10m	outdoor-20m	0.68	0.00	0.68
	tunnel-5m	0.78	0.00	0.78
tunnel-5m	outdoor-10m	0.76	0.00	0.76
	outdoor-20m	0.66	0.00	0.66

Appendix C. Contaminant detection in dynamic environment

Table C.14
Testing result of LiDAR contaminant detection in unseen dynamic dataset across different training model.

Dataset	Location	Condition	nSample	Contaminant Model	Acc	Pre	Rec	F1-score
KITTI	Urban and Rural Area	Day and Clean	1700	Tunnel-5m	1.000	1.000	1.000	1.000
				Outdoor-10m	0.999	1.000	0.999	0.999
				Outdoor-20m	0.988	1.000	0.989	0.994
LIDAROC_road_clean	Urban Area	Day and Clean	1700	Tunnel-5m	1.000	1.000	1.000	1.000
				Outdoor-10m	0.997	1.000	0.997	0.997
				Outdoor-20m	0.816	1.000	0.816	0.899
LIDAROC_road_mud	Urban and Flyover road	Day and Mud in Light Rain	1700	Tunnel-5m	0.951	1.000	0.951	0.975
				Outdoor-10m	1.000	1.000	1.000	1.000
				Outdoor-20m	1.000	1.000	1.000	1.000

References

Baccari, S., Hadded, M., Ghazzai, H., Touati, H., & Elhadef, M. (2024). Anomaly detection in connected and autonomous vehicles: A survey, analysis, and research challenges. *IEEE Access*, *12*, 19250–19276. https://doi.org/10.1109/ACCESS.2024.3361829

Beemelmanns, T., Zhang, Q., Geller, C., & Eckstein, L. (2024). Multicorrupt: A multi-modal robustness dataset and benchmark of liDAR-camera fusion for 3d object detection. In *IEEE intelligent vehicles symposium (iv)* (pp. 3255–3261). https://doi.org/10.1109/IV55156.2024.10588664

Benedek, C., Majdik, A., Nagy, B., Rozsa, Z., & Sziranyi, T. (2021). Positioning and perception in LIDAR point clouds. *Digital Signal Processing*, *119*, 103193. https://doi.org/10.1016/j.dsp.2021.103193

Bogdoll, D., Nitsche, M., & Zöllner, J. M. (2022). Anomaly detection in autonomous driving: A survey. In *The IEEE/CVF conference on computer vision and pattern recognition* (pp. 4488–4499).

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). Nuscenes: A multimodal dataset for autonomous driving. In *The IEEE/CVF conference on computer vision and pattern recognition* (pp. 11621–11631).

Chen, Y., Wang, Z., Zou, L., Chen, K., & Jia, K. (2022). Quasi-balanced self-training on noise-aware synthesis of object point clouds for closing domain gap. In *European conference on computer vision* (pp. 728–745). Springer.

Dong, Y., Kang, C., Zhang, J., Zhu, Z., Wang, Y., Yang, X., Su, H., Wei, X., & Zhu, J. (2023a). Benchmarking robustness of 3D object detection to common corruptions. In *The IEEE/CVF conference on computer vision and pattern recognition* (pp. 1022–1032).

Dong, Y., Kang, C., Zhang, J., Zhu, Z., Wang, Y., Yang, X., Su, H., Wei, X., & Zhu, J. (2023b). Benchmarking robustness of 3D object detection to common corruptions. In *The IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 1022–1032).

Egelhof, J., Wolf, P., & Berns, K. (2022). Disturbance and particle detection in liDAR data. In *Annual conference of the IEEE industrial electronics society* (pp. 1–6). IEEE.

Filgueira, A., González-Jorge, H., Lagüela, S., Díaz-Vilariño, L., & Arias, P. (2017). Quantifying the influence of rain in liDAR performance. *Measurement*, *95*, 143–148.

Gao, B., Pan, Y., Li, C., Geng, S., & Zhao, H. (2022). Are we hungry for 3d liDAR data for semantic segmentation? a survey of datasets and methods. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), 6063–6081. https://doi.org/10.1109/TITS.2021.3076844

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*.

Han, B., Liu, C. L., & Zhang, W. J. (2016). A method to measure the resilience of algorithm for operation management. *IFAC-PapersOnLine*, *49*(12), 1442–1447. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016. https://doi.org/10.1016/j.ifacol.2016.07.774

Heinzler, R., Schindler, P., Seekircher, J., Ritter, W., & Stork, W. (2019). Weather influence and classification with automotive lidar sensors. In *IEEE intelligent vehicles symposium (iv)* (pp. 1527–1534). https://doi.org/10.1109/IVS.2019.8814205

Huch, S., Scalerandi, L., Rivera, E., & Lienkamp, M. (2023). Quantifying the liDAR sim-to-real domain shift: A detailed investigation using object detectors and analyzing point clouds at target-level. *IEEE Transactions on Intelligent Vehicles*, *8*(4), 2970–2982. https://doi.org/10.1109/TIV.2023.3251650

James, J. K., Puhlfürst, G., Golyanik, V., & Stricker, D. (2018). Classification of lidar sensor contaminations with deep neural networks. In *The computer science in cars symposium (CSCS)* (p. 8).

Jati, G., Molan, M., Barchi, F., Bartolini, A., & Acquaviva, A. (2024a). Tinylid: A RISC-v accelerated neural network for liDAR contaminant classification in autonomous vehicle. In *the 21st ACM international conference on computing frontiers* CF '24 (p. 249–257). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3649153.3649201

Jati, G., Molan, M., Barchi, F., Bartolini, A., Mercurio, G., & Acquaviva, A. (2024b). Lidaroc: Realistic lidar cover contamination dataset for enhancing autonomous vehicle perception reliability. *IEEE Sensors Letters*, *8*(9), 1–4. https://doi.org/10.1109/LSENS.2024.3434624

Jati, G., Molan, M., Khan, J. A., Barchi, F., Bartolini, A., Mercurio, G., & Acquaviva, A. (2024c). AutogrAN: Autonomous vehicle liDAR contaminant detection using graph attention networks. In *Companion of the 15th ACM/SPEC international conference on performance engineering* ICPE '24 Companion (p. 112–119). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3629527.3652896

Khan, M. A., Menouar, H., Abdallah, M., & Abu-Dayya, A. (2024). LiDAR in connected and autonomous vehicles - perception, threat model, and defense. *IEEE Transactions on Intelligent Vehicles*, (pp. 1–19). https://doi.org/10.1109/TIV.2024.3510787

Li, D., Lu, C., Chen, Z., Guan, J., Zhao, J., & Du, J. (2024). Graph neural networks in point clouds: A survey. *Remote Sensing*, *16*(14). https://doi.org/10.3390/rs16142518

Li, S., Wang, Z., Juefei-Xu, F., Guo, Q., Li, X., & Ma, L. (2023). Common corruption robustness of point cloud detectors: Benchmark and enhancement. *IEEE Transactions on Multimedia*, (pp. 1–12). https://doi.org/10.1109/TMM.2023.3318317

Liang, D., Zhang, C., Zhang, P., Liu, S., Li, H., Niu, S., Rao, R. Z., Zhao, L., Chen, X., Li, H., & Huo, Y. (2024). Evolution of laser technology for automotive LiDAR, an industrial viewpoint. *Nature Communications*, *15*(1), 7660. https://doi.org/10.1038/s41467-024-51975-6

Liu, L., Lu, S., Zhong, R., Wu, B., Yao, Y., Zhang, Q., & Shi, W. (2021). Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet of Things Journal*, *8*(8), 6469–6486. https://doi.org/10.1109/JIOT.2020.3043716

Matos, F., Bernardino, J., Durães, J., & Cunha, J. (2024). A survey on sensor failures in autonomous vehicles: Challenges and solutions. *Sensors*, *24*(16). https://doi.org/10.3390/s24165108

Minh Mai, N. A., Duthon, P., Salmane, P. H., Khoudour, L., Crouzil, A., & Velastin, S. A. (2022). Camera and liDAR analysis for 3D object detection in foggy weather conditions. In *12th international conference on pattern recognition systems (ICPRS)* (pp. 1–7). https://doi.org/10.1109/ICPRS54038.2022.9854073

Mirza, M. J., Buerkle, C., Jarquin, J., Opitz, M., Oboril, F., Scholl, K.-U., & Bischof, H. (2021). Robustness of object detectors in degrading weather conditions. In *Ieee international intelligent transportation systems conference (itsc)* (pp. 2719–2724). https://doi.org/10.1109/ITSC48978.2021.9564505

Pandharipande, A., Cheng, C.-H., Dauwels, J., Gurbuz, S. Z., Ibanez-Guzman, J., Li, G., Piazzoni, A., Wang, P., & Santra, A. (2023). Sensing and machine learning for automotive perception: A review. *IEEE Sensors Journal*, *23*(11), 11097–11115. https://doi.org/10.1109/JSEN.2023.3262134

Pérez-Gil, Ó., Barea, R., López-Guillén, E., Bergasa, L. M., Gomez-Huelamo, C., Gutiérrez, R., & Diaz-Diaz, A. (2022). Deep reinforcement learning based control for autonomous vehicles in CARLA. *Multimedia Tools and Applications*, *81*(3), 3553–3576.

Ren, J., Pan, L., & Liu, Z. (2022). Benchmarking and analyzing point cloud classification under corruptions. In *International conference on machine learning* (pp. 18559–18575). PMLR.

Schillgalies, M., & Sharman, P. (2022). White paper: Unveiling the Mystery of LiDAR Qualification for Automotive Applications: The four pillars you should know. Technical Report First Sensor Westlake Village, CA.

Schlager, B., Goelles, T., Muckenhuber, S., & Watzenig, D. (2022). Contaminations on lidar sensor covers: Performance degradation including fault detection and modeling as potential applications. *IEEE Open Journal of Intelligent Transportation Systems*, *3*, 738–747. https://doi.org/10.1109/OJITS.2022.3214094

Sebastian, G., Vattem, T., Lukic, L., Bürgy, C., & Schumann, T. (2021). Rangeweathernet for liDAR-only weather and road condition classification. In *IEEE intelligent vehicles symposium (iv)* (pp. 777–784). https://doi.org/10.1109/IV48863.2021.9575320

Shahbeigi, S., Robinson, J., & Donzella, V. (2024). A novel score-based liDAR point cloud degradation analysis method. *IEEE Access*, *12*, 22671–22686. https://doi.org/10.1109/ACCESS.2024.3359300

Shi, S., Wang, X., & Li, H. (2019). PointRCNN: 3D object proposal generation and detection from point cloud. In *the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.

Tang, L., Shi, Y., He, Q., Sadek, A. W., & Qiao, C. (2020). Performance test of autonomous vehicle lidar sensors under different weather conditions. *Transportation Research Record*, *2674*(1), 319–329. https://doi.org/10.1177/0361198120901681

Team, O. D. (2020). OpenPCDet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet

Vargas Rivero, J. R., Gerbich, T., Teiluf, V., Buschardt, B., & Chen, J. (2020). Weather classification using an automotive LIDAR sensor based on detections on asphalt and atmosphere. *Sensors*, *20*(15). https://doi.org/10.3390/s20154306

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International conference on learning representations*.

Yan, Y., Mao, Y., & Li, B. (2018). Second: Sparsely embedded convolutional detection. *Sensors*, *18*(10). https://doi.org/10.3390/s18103337

Zhang, C., Huang, Z., Ang, M. H., & Rus, D. (2021). LiDAR degradation quantification for autonomous driving in rain. In *IEEE/RSJ international conference on intelligent robots and systems (iros)* (pp. 3458–3464). https://doi.org/10.1109/IROS51168.2021.9636694

Zhang, Y., Carballo, A., Yang, H., & Takeda, K. (2023). Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, *196*, 146–177. https://doi.org/10.1016/j.isprsjprs.2022.12.021

Zhao, J., Zhao, W., Deng, B., Wang, Z., Zhang, F., Zheng, W., Cao, W., Nan, J., Lian, Y., & Burke, A. F. (2024a). Autonomous driving system: A comprehensive survey. *Expert Systems with Applications*, *242*, 122836. https://doi.org/10.1016/j.eswa.2023.122836

Zhao, X., Fang, Y., Min, H., Wu, X., Wang, W., & Teixeira, R. (2024b). Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective. *Expert Systems with Applications*, *236*, 121358. https://doi.org/10.1016/j.eswa.2023.121358