POLITECNICO DI TORINO
Repository ISTITUZIONALE

LExUS: LiDAR Outlier Exposure for 3D Unknown Segmentation in Autonomous Driving

Original
LExUS: LiDAR Outlier Exposure for 3D Unknown Segmentation in Autonomous Driving / Marinai, A., Rai, S.N., Masone, C., Tommasi, T.. - In: IEEE ROBOTICS AND AUTOMATION LETTERS. - ISSN 2377-3766. - (In corso di stampa). [10.1109/LRA.2026.3756298]

Availability:
This version is available at: 11583/3018312 since: 2026-11-23T09:01:01Z

Publisher:
IEEE

Published
DOI:10.1109/LRA.2026.3756298

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©9999 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

27 November 2026

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2026.3756298

LExUS: LiDAR Outlier Exposure for 3D Unknown Segmentation in Autonomous Driving

Alessandro Marinai[1], Shyam Nandan Rai [1], Carlo Masone[1], and Tatiana Tommasi[1]

***Abstract*—LiDAR panoptic segmentation for driving scenes is challenged by the presence of instances of unknown classes. We introduce LExUS, a complete pipeline for Outlier Exposure (OE) that is designed to leverage auxiliary outliers provided as sparse point clouds drawn from real or CAD-generated synthetic data. The outliers are augmented preserving plausible point density and injected into the scene via an occupancy map to avoid collisions with scene objects. A scene-based intensity assignment ensures consistent reflectivity, preventing obvious learning shortcuts. The mixed samples are then used to fine-tune the model with a novel loss that combines distance-based weighting and hard-point mining to provide a stronger learning signal. Experiments on the STU benchmark demonstrate that LExUS surpasses all existing solutions, with massive improvements both on pixel-level metrics (-21.37 FPR@95, +48.95 AP) and on object-level ones (+21.96 RecallQ, +16.61 RQ, +16.83 UQ). Code is available at https://github.com/vandal-lab/LExUS**

***Index Terms*—Semantic Scene Understanding, Object Detection, Segmentation and Categorization, Intelligent Transportation Systems**

I. INTRODUCTION

AUTONOMOUS vehicles require an accurate and comprehensive understanding of their surroundings to operate safely in partially structured and highly dynamic environments. LiDAR sensors play a key role in this process by providing capabilities that complement RGB cameras. Unlike vision-based sensors, LiDAR is largely insensitive to illumination changes, from low-light to extremely bright conditions, and directly acquires long-range three-dimensional measurements. These characteristics support reliable depth perception and geometric scene understanding, which are essential for robust environment modeling and real-time decision making. However, algorithms for fine-grained understanding of LiDAR data remain less mature than their image-based counterparts. Most notably, panoptic segmentation methods for sparse point clouds, despite their recent improvements [1], [2], [3], [4], are mostly designed under a closed-set hypothesis. This means assuming that all semantic categories encountered at test time are known during training, which is unrealistic. Conversely, RGB-based methods have evolved towards an open set formulation [5], [6], enabling models not only to segment known categories but also to detect and appropriately handle previously unseen classes.

Manuscript received: April 13, 2026; Revised September 19, 2026; Accepted October 30, 2026. This paper was recommended for publication by Editor Hyungpil Moon upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by Politecnico di Torino, HPC@POLITO and Teoresi Group. Tatiana Tommasi acknowledges the EU project ELSA-European Lighthouse on Secure and Safe AI, under the EU's Horizon Europe Research and Innovation Programme (grant agreement no. 101070617)

[1]Politecnico di Torino, Turin, Italy `first.last@polito.it`

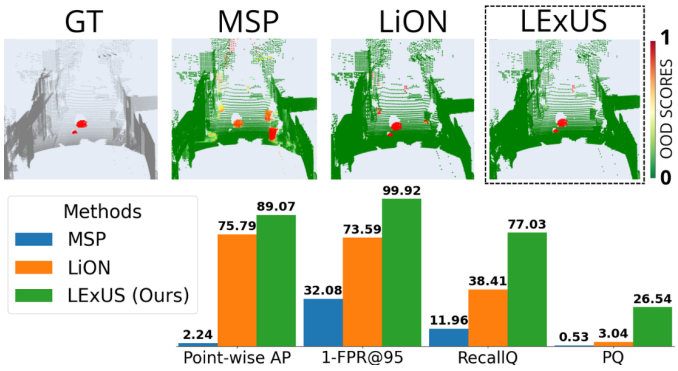

Fig. 1: **Comparison of unknowns segmentation in 3D point clouds**. Results on the STU dataset, comparing three different methods built on the same Mask4Former base model: i) the MSP [7] standard post-hoc baseline; ii) the OE method LiON [8]; iii) our LExUS using MSP as scoring function. **Top:** qualitative results. The base MSP misses one unknown object. It also segments many false positives, as well as LiON. **Bottom:** quantitative metrics; FPR@95 is shown as $1 - \text{FPR@95}$ for consistency (higher is better).

New evidence of this algorithmic gap was presented with the recent release of STU [7], a benchmark for 3D LiDAR anomaly segmentation built on a newly curated dataset. Crucially, in STU the out-of-distribution (OOD) objects are carefully selected to be atypical for a driving scene and they vary greatly both in size and kind, making it a challenging and realistic benchmark. With it, Nekrasov et al. [7] have conducted extensive tests combining post-hoc uncertainty-based methods [9], [10], [11], [12] with Mask4Former [4], a state-of-the-art method for LiDAR panoptic segmentation. Their results show that, albeit effective with RGB images, the naive application of these post-hoc methods to LiDAR data yields unsatisfactory results in detecting unknowns (see Fig. 1). These findings are indicative of the lack of large-scale pretrained feature extractors for sparse point clouds, as in contrast to 2D vision, and primarily emphasize the need for methods tailored to the 3D modality. **In this work we show that it is possible to improve LiDAR OOD recognition by developing a new Outlier Exposure (OE) pipeline.**

Outlier Exposure is a popular technique in 2D open-set segmentation [13], [14], [15], [16], [17], which encourages models to produce low-confidence predictions for unknown inputs. In images, this is typically implemented by cutting and pasting objects from auxiliary datasets, such as COCO [18], into in-distribution (ID) driving scene samples to expose the model to OOD instances during training. Adapting OE to LiDAR is largely unexplored due to two key obstacles: the lack

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 23,2026 at 08:52:32 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2026.3756298

of a standardized source of LiDAR OOD objects analogous to COCO, and the inherent difficulty of inserting objects into sparse point clouds compared to dense images.

A few exploratory works have suggested reusing portions of the inlier point cloud as outliers [19], [20]. To our knowledge, the only prior method demonstrating the use of auxiliary data for OE on LiDAR is LiON [8], which synthesizes outliers using CAD models from ShapeNet [21]. LiON inserts CAD objects into a scene with unconstrained translations and rotations and then simulates LiDAR beam interactions. While this approach allows flexible simulation of emission patterns, it precludes the use of outlier data captured by real LiDAR sensors as an auxiliary source. Moreover, LiON's pasting strategy ignores the semantic context of the scene, potentially positioning outliers on top of existing objects. Such unrealistic placements can create trivial shortcuts, allowing the model to distinguish outliers from the background without learning meaningful features. In addition, LiON disregards the intensity channel of the point cloud during pasting, removing a critical cue and forcing the model to rely solely on geometric information.

Here we introduce our LiDAR Outlier Exposure for 3D Unknown Segmentation (**LExUS**), a novel OE framework for LiDAR data in driving scenes that overcomes these limitations and introduces other improvements:

- **auxiliary outliers**: LExUS is designed to use sparse point cloud outliers, leveraging both real data and synthetic data from CAD models;
- **augmentations**: it applies transformations that vary outlier size and distance from the sensor while carefully preserving a point density consistent with range and avoiding geometrically implausible point clusters;
- **occupancy-aware insertion**: outliers are placed using an occupancy map to prevent unrealistic overlaps and clipping that could create learning shortcuts;
- **intensity assignment**: each pasted point's intensity is adjusted based on scene context and distance, preventing auxiliary data from being trivially distinguishable and enabling use of channels missing in synthetic data;
- **novel loss**: the model is fine-tuned with a novel loss that focuses on the most challenging OOD points.

We validate LExUS and all its individual components via extensive experiments on STU [7] and on SemanticKITTI [22]. Notably, it achieves consistently better performance than previous methods, with particularly strong gains on point-level and object-level OOD metrics (see Fig. 1).

II. Related Work

LiDAR 3D panoptic segmentation. Panoptic segmentation unifies semantic and instance segmentation by predicting for each point a class label and, for countable categories (*things*), an instance identity [23]. For LiDAR data this task has been tackled with different approaches: range-view methods, like RangeNet++ [24] and PointPillars [25], project the point cloud to a spherical image; Bird Eye View (BEV) methods like Panoptic-PolarNet [1] project the point cloud to an XY-plane grid. Other methods rely on spatial partitions of the scene to encode the point cloud more efficiently, such as DS-Net [3], which uses the cylindrical representation of Cylinder3D [26], and Efficient LPS [2], which uses a voxelized representation. More recently, mask-transformer architectures have been explored: Mask4Former [4] extends the mask-query transformer-based paradigm from 2D [27], [28] and indoor 3D [29] settings to outdoor scene LiDAR 3D panoptic segmentation.

While these existing works focus on a closed-set formulation, we propose a new Outlier Exposure pipeline that enables a reliable recognition of unknowns.

Segmentation of unknowns in LiDAR. Compared to the RGB domain, relatively few works investigate unknown object segmentation in LiDAR data [30], [19], [31], [20], [8], [7], and even fewer explore OE for sparse 3D point clouds. Early approaches repurpose inlier data as outliers: REAL [19] performs OE by using downscaled and upscaled versions of inliers, while REL [30] raises and rotates ground points to generate anomalies. However, these methods produce highly artificial outliers and may discard potentially useful training data. LiON [8] is, to the best of our knowledge, the first method to inject synthetic anomalies into LiDAR scenes using an auxiliary dataset, relying on CAD models from ShapeNet [21]. This strategy is tied to CAD-based objects and cannot directly leverage real LiDAR data as outliers. Moreover, it lacks mechanisms to ensure plausible object placement, which may lead to clipping, and it ignores the intensity channel of the point cloud. The fine-tuning objective of LiON includes an OE loss which encourages the model to abstain from predicting known classes on outlier samples by introducing an additional OOD class score and a point-wise abstention mechanism.

In this work, we propose an OE pipeline tailored for LiDAR data that addresses existing methods limitations. In addition to overcoming the constraints of LiON, our approach introduces dedicated augmentation strategies and a LiDAR-aware fine-tuning loss. We further show that the proposed pipeline is compatible with existing post-hoc scoring methods widely used in the 2D domain, and largely improves the recognition of unknown objects in 3D scenes.

III. 3D Outlier Exposure with LExUS

In this section we first introduce the problem setting of open-set panoptic segmentation for LiDAR data from driving scenes (Sec. III-A), and then present our OE solution in two parts: i) how the outliers are selected, processed and injected in the scene (Sec. III-B), and ii) how we fine-tune the model using these mixed samples (Sec. III-C).

A. Problem Setting

We consider a vehicle equipped with a rotating LiDAR that produces at each rotation cycle a scan of the environment $\mathcal{X} = \{\mathbf{x}^{(j)}\}_{j=1}^{N}$ with N points. The generic j-th point $\mathbf{x}^{(j)} \in R^4$ is a quadruplet, characterized by its position $\mathbf{p}^{(j)} \in \mathbb{R}^3$ relative to the sensor, expressed in Cartesian coordinates, and by the intensity $\iota^{(j)} \in \mathbb{R}$ of the reflected beam. Each point is associated with a semantic label $y^{(j)} \in \{1, 2, \cdots, |\mathcal{C}_{\text{in}}| + 1\}$, indicating one of the predefined *inlier* categories in the set $\mathcal{C}_{\text{in}}$ (e.g., *vehicle*, *pedestrian*, *traffic sign*), with an additional

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 23,2026 at 08:52:32 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2026.3756298

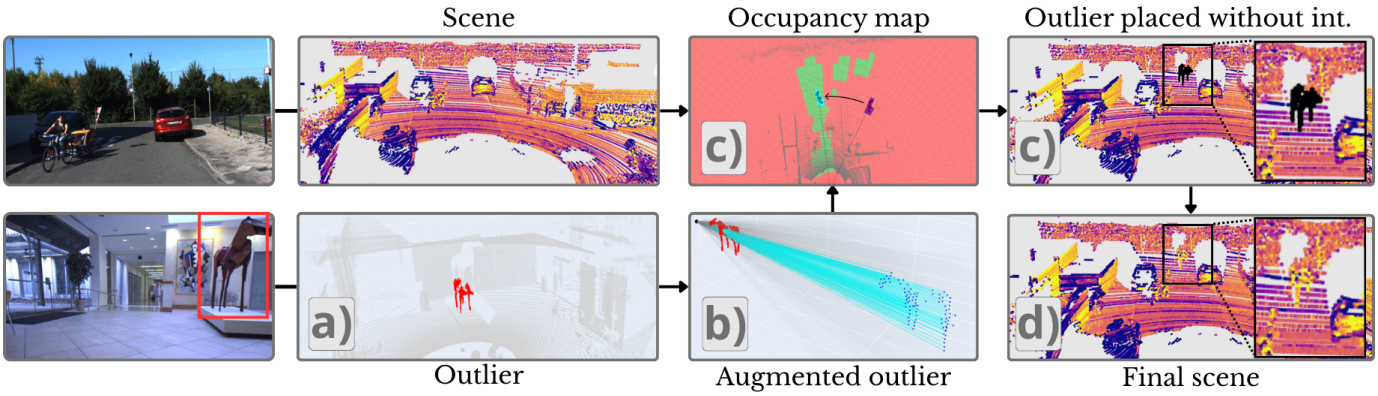

Fig. 2: **LExUS anomaly pasting pipeline.** a) An out-of-distribution (OOD) object is extracted from an auxiliary dataset. b) Augmentations are applied to the object c) The original scene is checked for overlap using an occupancy map. If an overlap is detected, the object is rotated until a overlap-free placement is found. (d) Intensity values are created based on the scene.

label assigned to unknown objects ($|\mathcal{C}_{\text{in}}| + 1$). Furthermore, each point is assigned an index $z^{(j)} \in \mathcal{N}$ for the countable classes (i.e., *things*), enabling instance-level differentiation. The task of *open-set* LiDAR panoptic segmentation aims to find a mapping $f_\theta(\mathcal{X}) = \{(\hat{y}^{(j)}, \hat{z}^{(j)})\}_{j=1}^{N}$, where θ are learnable parameters. It is optimized on a labeled dataset $\mathcal{D}_{\text{in}} = \{(\mathcal{X}_i^{\text{in}}, \mathcal{Y}_i^{\text{in}}, \mathcal{Z}_i^{\text{in}})\}_{i=1}^{N_{\text{in}}}$ that contains scans $\mathcal{X}_i^{\text{in}}$ with label $\mathcal{Y}_i^{\text{in}}$ and instance $\mathcal{Z}_i^{\text{in}}$ annotations, and without unknowns.

Outlier exposure strategies improve the model's capability to recognize unknowns by supplementing the training dataset $\mathcal{D}_{\text{in}}$ with a set $\mathcal{D}_{\text{aux}} = \{\mathcal{X}_i^{\text{aux}}\}_{i=1}^{N_{\text{aux}}}$ of instances that do not belong to any category in $\mathcal{C}_{\text{in}}$. Mixed examples can thus be created by injecting instances from $\mathcal{D}_{\text{aux}}$ into samples from $\mathcal{D}_{\text{in}}$, and then used to fine-tune the model that was previously pre-trained on $\mathcal{D}_{\text{in}}$. The goal of this fine-tuning step is to improve the model's capability to discriminate between known and unknown classes [13].

B. Outliers Injection

Our procedure to inject outliers in a LiDAR scan consists of multiple steps (see Fig. 2) described in the following.

a) Auxiliary data import. Selecting the source of auxiliary data $\mathcal{D}_{\text{aux}}$ is the first crucial step for OE. In LExUS, we consider $\mathcal{D}_{\text{aux}}$ to be provided as a collection of sparse-point clouds from different objects (see Fig. 2a). These auxiliary outliers can be both synthetic, created from CAD models by simulating the beams hitting them, and real, recorded with physical LiDAR sensors. This general formulation allows LExUS to leverage the very large number of CAD object models available online, while also relying on a smaller quantity of high-quality real LiDAR recordings.

Our definition of the auxiliary data source is different from LiON [8], which is the only other existing approach for 3D LiDAR outlier exposure based on an auxiliary dataset. In fact, LiON uses a set $\mathcal{D}_{\text{aux}}$ made exclusively of 3D CAD models from ShapeNet [21] and it is incompatible with using real LiDAR instances. While LiON's insertion mechanism (see [8] for details) is effective for dense synthetic models, it requires applying arbitrary translations and rotations to the 3D objects before simulating the LiDAR beams. Such unconstrained transformations are not applicable to real LiDAR point clouds,

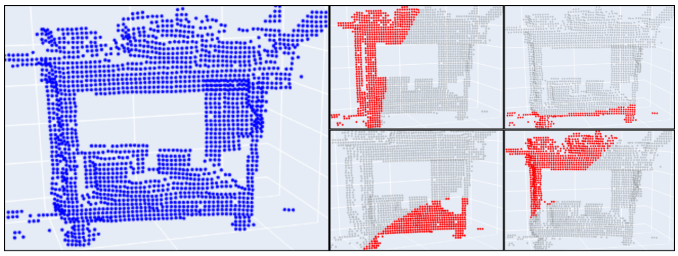

Fig. 3: **Part augmentation.** Original object (blue) and augmented versions (red) with different sampled directions $\mathbf{u}$.

where point visibility and density are inherently tied to the object's pose relative to the sensor. Treating a sparse point cloud as a rigid object and relocating it in the scene would produce physically inconsistent artifacts (e.g., incorrect point density at a given distance), ultimately introducing unrealistic outliers and enabling trivial learning shortcuts. In the next steps we discuss how our insertion strategy solves this problem.

b) Outliers augmentation. The outliers in $\mathcal{D}_{\text{aux}}$, particularly when derived from real LiDAR data, are typically obtained from whole objects captured at relatively close range. For example, in the UT Campus dataset [32], a real-world data source that we use to build $\mathcal{D}_{\text{aux}}$ (see Sec. IV), most instances correspond to large objects (e.g., table, dumpster etc.) placed 8-16 meters away from the sensor. This may introduce a bias, potentially limiting the model's ability to generalize. To mitigate this issue, we propose two complementary augmentation strategies that increase the diversity of outliers in terms of both size and spatial distribution relative to the sensor.

Our first augmentation produces smaller anomalies by removing portions of an object according to the sensor viewpoint. In this way, we expand the set $\mathcal{D}_{\text{aux}}$ with more challenging anomalies that are represented by fewer points. To introduce this *part augmentation*, let us assume to start from one selected outlier $\mathcal{X}_i^{\text{aux}} = \{\mathbf{x}_i^{\text{aux},(j)}\}_{j=1}^{n_i} \in \mathcal{D}_{\text{aux}}$. We uniformly sample a random direction $\mathbf{u} \in \mathbb{R}^3$ and a retention ratio $\rho \in (0, 1)$, which determines the proportion of points to keep. For each point $\mathbf{x}_i^{\text{aux},(j)}$ we evaluate the alignment between its position $\mathbf{p}_i^{\text{aux},(j)}$ and $\mathbf{u}$ via the cosine similarity

$$\alpha_j = \frac{\mathbf{p}_i^{\text{aux},(j)} \cdot \mathbf{u}}{\|\mathbf{p}_i^{\text{aux},(j)}\|_2 \, \|\mathbf{u}\|_2} \,. \tag{1}$$

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 23,2026 at 08:52:32 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2026.3756298

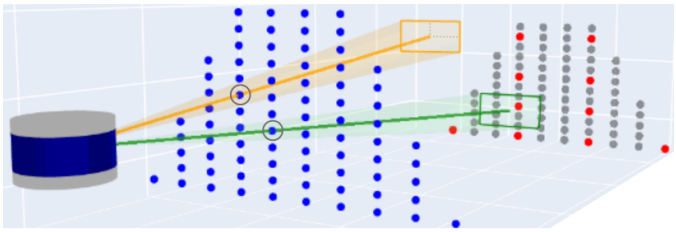

Fig. 4: **Distance augmentation**. Blue points denote the original object, while red points indicate those retained after translation. The rectangles define the regions considered by each ray according to Eq. (3): they appear rectangular due to the use of separate ϵ values for the azimuth and elevation angles. The green ray intersects a point within its region, whereas the orange ray does not detect any.

Finally, we retain the ρ-fraction of points with highest α_j (see Fig. 3). As a result, the outlier is reduced in size while maintaining plausible pattern and density of the points, contrary to the naive down-scaling of objects in REAL [19].

Our second augmentation relocates anomalies farther from the sensor while preserving physical consistency, thereby increasing the diversity of observed distances (see Fig. 2b). Given an outlier $\mathcal{X}_i^{\text{aux}} = \{\mathbf{x}_i^{\text{aux},(j)}\}_{j=1}^{n_i} \in \mathcal{D}_{\text{aux}}$, we compute its centroid $\mathbf{c} \in \mathbb{R}^3$ and define the radial direction $\mathbf{r} = \mathbf{c}/\|\mathbf{c}\|_2$. The object is then translated along this direction by a random offset δ:

$$\widetilde{\mathbf{p}}_i^{\text{aux},(j)} = \mathbf{p}_i^{\text{aux},(j)} + \delta\mathbf{r} \ . \tag{2}$$

To preserve realistic LiDAR sampling patterns, we retain only the translated points that remain consistent with the original beam directions (see Fig. 4). Specifically, a translated point is kept if it is approximately aligned with at least one ray that intersected the original object:

$$\text{keep } \widetilde{\mathbf{x}}_i^{\text{aux},(j)} \text{ iff } \exists\, \mathbf{x}_i^{\text{aux},(k)} : \left\langle \frac{\widetilde{\mathbf{p}}_i^{\text{aux},(j)}}{\|\widetilde{\mathbf{p}}_i^{\text{aux},(j)}\|_2}, \frac{\mathbf{p}_i^{\text{aux},(k)}}{\|\mathbf{p}_i^{\text{aux},(k)}\|_2} \right\rangle < \epsilon \tag{3}$$

where $j, k \in \{1, \dots, n_i\}$ and ϵ is a tolerance hyperparameter.

Both augmentations are applied to the instance $\mathcal{X}_i^{\text{aux}}$, producing the augmented outlier $\widetilde{\mathcal{X}}_i^{\text{aux}}$.

c) Occupancy-aware outlier insertion. The next step is inserting the augmented outlier $\widetilde{\mathcal{X}}_i^{\text{aux}}$ into a scene $\mathcal{X}_l^{\text{in}} \in \mathcal{D}_{\text{in}}$. With RGB images, this operation is straightforward as pasted objects are always in the foreground. However, adding instances to a LiDAR scan without any verification may cause them to collide with existing objects and produce clipping artifacts, as it happens in LiON [8] (see Fig. 5-left).

In LExUS we address this issue by first computing an occupancy map $\mathcal{O}_i$, to identify free regions where outliers can be safely inserted. We project the point cloud onto the ground plane, discarding the vertical component, and partition the resulting 2D space into bins using a cylindrical discretization (see Fig. 5-middle). A bin is considered *free* if it contains at least one point labeled as *road* and no points belonging to other inlier classes in $\mathcal{C}_{\text{in}}$. This corresponds to regions of unobstructed road surface, suitable for anomaly pasting. The resulting occupancy map enables efficient validation of feasible insertion locations (see Fig. 5-right) and implicitly accounts for occlusions, as regions behind objects do not contain visible road points.

For efficiency, we pre-compute the occupancy map for each training scan. During training, we only verify whether the bins corresponding to the pasted outlier $\widetilde{\mathcal{X}}_i^{\text{aux}}$ are free. If the check fails, we apply a relocation strategy inspired by PolarMix [33], rotating the anomaly by a small random angle around the LiDAR $\vec{Z}$ axis (see Fig. 2c), and repeat the validation. This process is iterated until a feasible placement is found. By preserving the distance from the sensor, this strategy maintains consistent point density, while rotation around the sensor also preserves the viewpoint, ensuring that the same object surfaces remain visible to the LiDAR.

Once a valid position is determined, we estimate the ground level by selecting the road point with the highest Z-coordinate among those within 0.5m radius of the object centroid. The anomaly is then vertically translated so that its lowest point aligns with this ground plane. We also enforce consistency by removing points occluded by the pasted object from the sensor's viewpoint, as done in LiON [8]. The resulting augmented scene is denoted as $\widetilde{\mathcal{X}}^{\text{OE}}$, obtained by inserting $\widetilde{\mathcal{X}}_i^{\text{aux}}$ into $\mathcal{X}_l^{\text{in}}$.

d) Scene-based intensity assignment. The previous step ensures geometric plausibility of the mixed scene $\widetilde{\mathcal{X}}^{\text{OE}}$, but it does not enforce consistency in the intensity channel. Indeed, the auxiliary instance $\mathcal{X}_i^{\text{aux}}$ may differ in intensity from the surrounding scene due to acquisition setting variations, or may lack the intensity entirely when simulated from a 3D CAD model. This mismatch should be solved to avoid the model exploiting it as a shortcut for anomaly detection.

Existing LiDAR outlier exposure methods [8], [31], [19] bypass these problems by disregarding the intensity channel. Instead, we address them through a scene-based intensity assignment strategy. Given the mixed scan $\widetilde{\mathcal{X}}^{\text{OE}}$, we first take all in-distribution (ID) instances and rank them in terms of how similar their distance from the sensor is with respect to the pasted outlier. We then take the top-ranked instance $\mathcal{X}_l^{\text{iid}} \subset \mathcal{X}_l^{\text{in}}$ and we transfer its intensity values to the outlier via a spatial traversal procedure:

1) We build k-NN graphs ($k = 8$) for $\widetilde{\mathcal{X}}_i^{\text{aux}}$ and $\mathcal{X}_l^{\text{iid}}$.
2) Using Dijkstra's algorithm we create an order of points for both instances starting from a random point.
3) The intensities of the points from $\mathcal{X}_l^{\text{iid}}$ are transferred to $\widetilde{\mathcal{X}}_i^{\text{aux}}$ by following the Dijkstra ordering, adding a small random noise to avoid copying the exact pattern.

We remark that this operation does not add significant computational overhead as Dijkstra's algorithm is applied to the individual instances $\widetilde{\mathcal{X}}_i^{\text{aux}}$ and $\mathcal{X}_l^{\text{iid}}$, rather than the full scene $\mathcal{X}_l^{\text{in}}$. The resulting mixed scan, with intensity values assigned to the pasted anomaly, is denoted as $\mathcal{X}^{\text{OE}}$. While the procedure has been described for a single anomaly, it naturally extends to multiple outliers within the same scene.

C. Fine-Tuning with Exposed Anomalies

The outlier injection pipeline described above produces mixed scans $\mathcal{X}^{\text{OE}} = \{\mathbf{x}^{\text{OE},(j)}\}_{j=1}^{N^{\text{OE}}}$ with N^{OE} points. We denote the index sets of anomalous and inlier points as

$$\begin{aligned} \mathcal{I}_{\text{ood}} &= \{j \in \{1, \dots, N^{\text{OE}}\} \mid y^{\text{OE},(j)} = |\mathcal{C}_{\text{in}}| + 1\} \\ \mathcal{I}_{\text{id}} &= \{j \in \{1, \dots, N^{\text{OE}}\} \mid y^{\text{OE},(j)} \in \mathcal{C}_{\text{in}}\} \ . \end{aligned} \tag{4}$$

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 23,2026 at 08:52:32 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2026.3756298

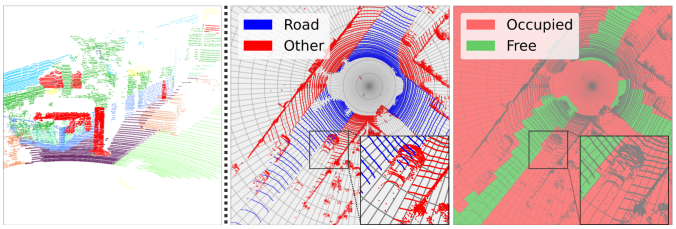

Fig. 5: **Outlier insertion**. **Left**: Anomaly insertion example with LiON [8]: the outliers (in red) are pasted on top of existing objects. **Right**: LExUS overcomes this problem with an occupancy map computed using polar bins.

By construction, $\mathcal{I}_{\text{ood}} \cup \mathcal{I}_{\text{id}} = \{1, \ldots, N_i^{\text{OE}}\}$.

LExUS uses $\mathcal{X}^{\text{OE}}$ to fine-tune the model f_θ originally trained only on inlier data, without unknowns. For our study, the base model f_θ is Mask4Former [4]. During fine-tuning we retain its standard training loss $\mathcal{L}_{\text{M4F}}$ [4] and supplement it with an additional loss that aims to improve the separation between in-distribution and out-of-distribution classes. It is implemented as a weighted cross-entropy:

$$\mathcal{L}_{\text{OE}} = -\frac{1}{|\mathcal{I}_{\text{ood}}|} \sum_{u \in \mathcal{I}_{\text{ood}}} \underbrace{\alpha^{(u)} \log(s^{(u)})}_{\ell_u^{ood}} - \frac{1}{|\mathcal{I}_{\text{id}}|} \sum_{v \in \mathcal{I}_{\text{id}}} \underbrace{\alpha^{(v)} \log(1 - s^{(v)})}_{\ell_v^{id}}, \tag{5}$$

where u, v are point indexes and $s^{(j)} \in [0, 1]$ is a per-point bounded anomaly score obtained by applying a sigmoid activation to the OOD score from a post-hoc method. The weighting terms consist of an inverse class-frequency balancing $1/|\mathcal{I}|$, and a distance-based factor α. The former prevents the function from being overly influenced by inlier points in $\mathcal{I}_{\text{id}}$, which are typically far more numerous than anomalies in $\mathcal{I}_{\text{ood}}$. The latter accounts for LiDAR points becoming sparser as distance from the sensor increases. We set

$$\alpha^{(j)} = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \cdot \left(\min\left(1, \frac{d^{(j)}}{d_{\max}}\right) \right)^{\gamma}, \tag{6}$$

where $d^{(j)}$ indicates the distance of the j-th point from the sensor, γ is a scaling factor, and $\alpha_{\min}$, $\alpha_{\max}$ and $d_{\max}$ are hyperparameters that bound the range of variations and prevent excessively large weights for far-range measurements. We fix these values to $\alpha_{min} = 1$, $\alpha_{max} = 2$, $d_{max} = 50$m.

Although effective, the proposed loss may be dominated during training by the large number of already correctly classified points. To address this, we incorporate Online Hard Example Mining (OHEM) [34] into Eq. (5), restricting backpropagation to a fixed fraction of the hardest samples. Concretely, we select the top-$\beta\%$ points $u \in \mathcal{I}_{ood}$ and $v \in \mathcal{I}_{id}$ based on their contributions to $-\ell_u^{\text{ood}}$ and $-\ell_v^{\text{id}}$.

IV. Experiments

Training/Evaluation Datasets. We conduct experiments on:

- **STU Dataset** [7] is a benchmark for 3D LiDAR unknown segmentation that provides a test set with sequences acquired with an OS1-128 Ouster LiDAR sensor and staged anomalies. The benchmark uses a combination of SemanticKITTI [22], Panoptic-CUDAL [35], and two new LiDAR sequences as training set.
- **SemanticKITTI** [22] is collected using a Velodyne HDL-64E and is labeled with 19 known classes. It does not have a test split with unknown objects, but we follow prior works [19], [8] in training on 18 classes and considering the remaining one (*other-vehicle*) as anomalous.

Auxiliary Datasets. The outliers $\mathcal{D}_{\text{aux}}$ are sourced from:

- **ShapeNet** [21], a large-scale dataset of 3D CAD models covering a wide range of object categories, and it is used to synthetically generate anomalous objects.
- **UT Campus Dataset** [32], a real-world LiDAR dataset collected by a mobile robot equipped with an OS1-128 Ouster sensor, capturing diverse outdoor and indoor campus environments and a wide variety of object categories. We rely on the provided 3D bounding box annotations and extract object point clouds.

For both datasets we ensure that the selected OOD categories are fully disjoint from the inlier semantic classes used during training, so that the model is never exposed to known classes as anomalies. We also avoid overlaps with the anomaly categories present in the test sets, preventing evaluation bias.

Implementation Details. We first train the vanilla Mask4Former on the training split, without OE. Then we fine-tune it with LeXUS, updating all layers. Fine-tuning is performed for 12 epochs using the AdamW optimizer with a learning rate of 1×10^{-4} and a batch size of 6 distributed on 2 NVIDIA GeForce RTX 4090. Anomaly pasting is applied on-the-fly during training, using the pipeline described in Sec. III-B. Anomaly scores are obtained with MSP [9], unless otherwise specified. Validation scenes are constructed by pasting anomalies into validation scans, following the same procedure used for training. Model selection is performed by monitoring the anomaly binary AP on the validation set. To avoid biased validation scores, anomaly objects used for validation are disjoint from those used in training.

Evaluation Metrics. For STU we compute the point-level and object-level metrics defined in the original paper [7]. Inlier performance is reported on the SemanticKITTI validation set, to assess the potential degradation of inlier prediction caused by OE. For SemanticKITTI OOD performance, we report AUPR and AUROC, as in [19], [8].

Baselines. We compare LExUS to different families of methods. First, we consider baselines *without OE*, equipping the pre-trained Mask4Former with post-hoc uncertainty scoring strategies [12], [11], [10], [9], [36], [37], [38]. Second, we include methods (Void Classifier [39], REAL [19], REL[31]) that perform *OE without auxiliary data*, re-purposing training (inlier) points as outliers. Finally, we compare to LiON [8], as the only other existing method that performs *OE with auxiliary data*. Since LiON is the strongest competitor, we report results using both its original Cylinder3D backbone [26], and our Mask4Former-based reimplementation to provide a direct and fair benchmark.

A. Main Results

Unknown results on STU. Table I reports results on the

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 23,2026 at 08:52:32 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2026.3756298

TABLE I: **Results on STU**. M4F stands for Mask4Former [4], C3D stands for Cylinder3D [26]. Methods that use C3D, denoted with †, are designed for semantic segmentation, not panoptic, hence their inlier results cannot be computed. We use $*$ to indicate models whose implementation was not released, so the results are copied from their original paper, which lacks inlier performance. Best results are highlighted in **bold**.

	Method	OE data	Model	Point-Level OOD			Object-Level OOD					Inliers Panoptic		
				AUROC↑	FPR@95↓	AP↑	RecallQ↑	SQ↑	RQ↑	UQ↑	PQ↑	SQ_{ID}↑	RQ_{ID}↑	PQ_{ID}↑
No OE	Deep Ensemble [12]	×	M4F	86.74	58.05	5.17	16.75	84.49	10.43	14.16	8.81	**81.0**	**70.4**	**60.7**
	MC Dropout [11]	×	M4F	61.51	82.37	0.11	2.25	**86.72**	1.95	2.14	1.86	80.1	69.7	60.0
	Max Logit [10]	×	M4F	84.53	81.49	0.95	26.14	83.06	2.13	21.71	1.77	80.1	69.7	60.0
	MSP [9]	×	M4F	87.45	67.92	2.24	11.96	84.88	0.62	10.15	0.53	80.1	69.7	60.0
	RbA [36]	×	M4F	66.38	100.0	0.81	24.04	83.28	3.23	20.02	2.69	80.1	69.7	60.0
	MaxEntropy [37]	×	M4F	88.24	38.96	0.98	26.77	78.45	1.16	21.00	0.91	80.1	69.7	60.0
	MaxEnergy [38]	×	M4F	90.83	51.09	2.88	8.87	82.72	0.52	7.34	0.43	80.1	69.7	60.0
OE w./o. $\mathcal{D}_{aux}$	REAL [19]†	Scaled Inliers	C3D	79.18	77.34	2.53	9.36	84.72	0.59	7.93	0.50	–	–	–
	Void Classifier [39]	*Unlabeled*	M4F	85.99	78.60	3.92	17.64	84.40	8.19	14.89	6.91	78.9	69.0	57.8
	REL [31]*	Floor Raise	M4F	96.26	21.69	10.17	33.98	83.39	11.64	28.34	9.71	–	–	–
OE w. $\mathcal{D}_{\text{aux}}$	LiON [8]†	ShapeNet	C3D	96.53	26.41	33.33	50.80	76.66	2.66	38.94	2.04	–	–	–
	LiON [8]	ShapeNet	M4F	97.02	21.45	40.12	52.15	75.89	3.47	44.23	2.63	74.9	65,1	54.9
	LExUS	ShapeNet	M4F	98.90	0.55	77.31	66.88	76.72	8.10	49.97	6.21	79.1	69.1	58.0
	LExUS	UT	M4F	99.82	0.13	86.38	71.10	76.80	22.69	54.61	17.43	79.4	69.3	58.4
	LExUS	UT + ShapeNet	M4F	**99.85**	**0.08**	**89.07**	**77.03**	78.98	**33.60**	**60.84**	**26.54**	79.3	69.3	58.3

STU benchmark. Inlier-only methods struggle to recognize unknowns across most metrics. OE methods without auxiliary data yield mixed performance; while REL improves AUROC and FPR@95, it achieves limited object-level gains (results taken from their manuscript, as code is unreleased). LiON, the only competitor using auxiliary data for OE, greatly outperforms all other baselines. Yet, LExUS surpasses it across all point- and object-level metrics. By jointly utilizing the ShapeNet and UT datasets, our method exploits both the diversity of synthetic data and the quality of real outliers. This results in top performance, substantially outperforming LiON in FPR@95 (-21.37), AP (+48.95), RQ (+21.96), UQ (+16.61), and PQ (+16.83). Interestingly, LiON and LExUS underperform on the SQ metric compared to the other baselines. Since SQ only evaluates correctly detected anomalies, this highlights a favorable trade-off: our substantially stronger detection capability (many more true positives) comes at the cost of marginally less precise boundary alignment.

Inliers results on STU. Table I (last three columns) also contains the results on the inlier classes of the STU benchmark. All baselines without OE match the performance of the vanilla Mask4Former trained only on inlier classes, as they do not involve any fine-tuning. As expected, methods that incorporate OE exhibit a slight degradation. However, in the case of LExUS this drop is negligible, especially when contrasted with the substantial gains in OOD metrics.

OOD results on SemanticKITTI. To asses the performance in anomaly segmentation on SemanticKITTI, we follow the protocol from [19], [8], using the *other-vehicle* category as outlier and training the base model only on the remaining 18 classes. LExUS outperforms all baselines in both AUPR and AUROC (see Tab. II).

TABLE II: **Unknowns segmentation on SemanticKITTI**, using the *other-vehicle* class as anomaly. M4F stands for Mask4Former [4], C3D stands for Cylinder3D [26]. Best results are highlighted in **bold**.

	Method	OE data	Model	AUPRC↑	AUROC↑
No OE	MSP [9]	×	M4F	7.51	72.53
	MaxLogit [10]	×	M4F	6.53	69.41
OE w./o. $\mathcal{D}_{aux}$	REAL [19]	inliers	C3D	20.08	84.90
OE w. $\mathcal{D}_{aux}$	LiON [8]	ShapeNet	C3D	44.68	92.69
	LiON [8]	ShapeNet	M4F	44.99	95.28
	LExUS	ShapeNet	M4F	47.51	97.32
	LExUS	UT	M4F	**48.00**	**97.55**
	LExUS	UT + ShapeNet	M4F	47.94	97.48

B. Ablations and component-specific analyses.

LExUS with different scoring methods. To demonstrate the generality of our framework, we evaluate LExUS on STU using different post-hoc OOD scoring functions. Results in Tab. III show consistent improvements in unknown segmentation across all settings, with the only exception being the SQ metric, as already observed in Tab. I. These results also confirm that performance on inlier classes is preserved compared to the base model without OE.

Component-wise comparison with LiON. To demonstrate the importance of both our outlier injection strategy and the loss function $\mathcal{L}_{\text{OE}}$, we conduct experiments replacing each component with its counterpart from LiON, the only other existing LiDAR-based OE method that leverages auxiliary data. To make a fair comparison, we use the same base model (Mask4Former) and the same source of outlier (ShapeNet, to adhere closely to the original implementation of LiON) for all experiments. The results shown in Tab. IV indicate that our outlier injection and our fine-tuning loss $\mathcal{L}_{\text{OE}}$ separately provide significant improvements over the full LiON. Moreover, the gains are complementary, with our fine-tuning helping on RecallQ, SQ, UQ and our outlier injection pipeline providing an adavantage on RQ, PQ and a major boost in AP. Finally when combining the two components into our full LExUS, the individual improvements are reinforced yielding more favorable scores on all metrics.

Ablation on outlier injection components. Table V ablates our outlier injection pipeline. Occupancy-aware pasting (O) proves most critical, with its removal causing the sharpest

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 23,2026 at 08:52:32 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2026.3756298

TABLE III: **LExUS with different scoring functions.** Results computed on STU. LExUS uses UT as a source for $\mathcal{D}_{\text{aux}}$.

Post-hoc	LExUS	Point-Level OOD			Object-Level OOD					Inliers Panoptic		
		AUROC↑	FPR@95↓	AP↑	RecallQ↑	SQ↑	RQ↑	UQ↑	PQ↑	SQ_{ID}↑	RQ_{ID}↑	PQ_{ID}↑
MaxEnergy [38]	×	90.83	51.09	2.88	8.87	**82.72**	0.52	7.34	0.43	80.1	**69.7**	**60.0**
	✓	**99.38** (+9.45)	**0.65** (-50.44)	**74.51** (+71.63)	**65.04** (+56.17)	75.37 (-7.35)	**8.54** (+8.02)	**49.02** (+41.68)	**6.44** (+6.01)	**80.7** (+0.6)	68.9 (-0.8)	59.0 (-1.0)
MaxEntropy [37]	×	88.24	38.96	0.98	26.77	**78.45**	1.16	21.00	0.91	**80.1**	**69.7**	**60.0**
	✓	**98.84** (+10.60)	**1.18** (-37.78)	**80.76** (+79.78)	**64.87** (+38.10)	74.89 (-3.56)	**5.90** (+4.74)	**48.58** (+27.58)	**4.42** (+3.51)	79.6 (-0.5)	69.4 (-0.3)	59.2 (-0.8)
MaxLogit [10]	×	84.53	81.49	0.95	26.14	**83.06**	2.13	21.71	1.77	**80.1**	**69.7**	**60.0**
	✓	**99.30** (+14.77)	**1.21** (-80.28)	**57.27** (+56.32)	**66.16** (+40.02)	74.64 (-8.42)	**6.01** (+3.88)	**49.38** (+27.67)	**4.48** (+2.71)	75.3 (-4.8)	68.1 (-1.6)	57.5 (-2.5)
MSP [9]	×	87.45	67.92	2.24	11.96	**84.88**	0.62	10.15	0.53	**80.1**	**69.7**	**60.0**
	✓	**99.82** (+12.37)	**0.13** (-67.79)	**86.38** (+84.14)	**71.10** (59.14)	76.80 (-8.08)	**22.69**(+22.07)	**54.61** (+44.46)	**17.43** (+16.9)	79.4 (-0.7)	69.3 (-0.4)	58.4 (-1.6)

TABLE IV: **Component-wise comparison with LiON**. Different combinations of outliers pasting and fine-tuning, from LiON and LExUS. All results are computed on STU, using ShapeNet as source for $\mathcal{D}_{\text{aux}}$ and M4F as base model.

Pasting	Fine-Tuning	Point-Level OOD			Object-Level OOD				
		AUROC↑	FPR@95↓	AP↑	RecallQ↑	SQ↑	RQ↑	UQ↑	PQ↑
LiON	LiON	97.02	21.45	40.12	52.15	75.89	3.47	44.23	2.63
LiON	**LExUS**	99.12 (+2.1)	4.96 (-16.49)	24.01 (-16.11)	57.41 (+5.26)	77.15 (+1.26)	2.13 (-1.34)	44.29 (+0.06)	1.65 (-0.98)
LExUS	LiON	99.50 (+2.48)	0.97 (-20.48)	75.79 (+35.67)	38.41 (-13.74)	72.09 (-3.8)	4.21 (+0.74)	27.69 (-16.54)	3.04 (+0.41)
LExUS	**LExUS**	**98.90** (+1.88)	**0.55** (-20.9)	**77.31** (+37.19)	**66.88** (+14.73)	**76.72** (+0.83)	**8.10** (+4.63)	**49.97** (+5.74)	**6.21** (+3.58)

TABLE V: **Ablation on individual components of LExUS' outliers injection.** *A, O* and *I* denote respectively the part and distance augmentations, the occupancy-aware insertion and the scene-based intensity assignment. Gauss. refers to sampling point intensities from a Gaussian whose mean and std are estimated from a reference instance in the scene. Results computed on STU, using UT as source for $\mathcal{D}_{\text{aux}}$.

A	O	I	Point-Level OOD			Object-Level OOD				
			AUROC↑	FPR@95↓	AP↑	RecallQ↑	SQ↑	RQ↑	UQ↑	PQ↑
×	✓	✓	99.24	0.58	77.52	68.39	75.20	7.66	51.42	5.76
✓	×	✓	97.91	5.29	45.63	61.93	75.84	2.44	46.97	1.85
✓	✓	×	99.73	0.75	71.08	57.32	75.46	4.34	43.25	3.28
✓	✓	Gauss.	99.20	0.40	79.58	70.75	76.09	12.45	53.83	9.48
✓	✓	✓	**99.82**	**0.13**	**86.38**	**71.10**	**76.80**	**22.69**	**54.61**	**17.43**

TABLE VI: **Ablation on different components of $\mathcal{L}_{\text{OE}}$.** Results computed on STU, using UT as source for $\mathcal{D}_{\text{aux}}$.

Dist. Weigh. [γ]	OHEM [β]	Point-Level OOD			Object-Level OOD				
		AUROC↑	FPR@95↓	AP↑	RecallQ	SQ	RQ	UQ	PQ
×	×	99.44	0.67	81.28	70.56	76.38	6.33	53.66	4.83
1.0	×	99.90	0.22	80.66	70.17	76.51	14.59	53.69	11.16
1.5	×	99.78	0.26	80.85	64.98	73.56	12.24	47.80	9.01
2.0	×	99.73	0.29	80.28	64.04	72.37	8.54	49.02	6.18
1.0	0.7	99.81	0.26	80.87	59.60	73.57	11.29	43.84	8.31
1.0	0.5	99.85	0.22	79.78	65.38	73.72	14.01	48.20	10.33
1.0	0.3	**99.82**	**0.13**	**86.38**	**71.10**	**76.80**	**22.69**	**54.61**	**17.43**
1.0	0.1	99.27	0.20	85.65	67.08	74.47	16.24	49.95	12.09

metric drops. Training without augmentations (A) or completely disregarding intensity (I = × means the intensity channel was identically set to zero) causes milder degradation. Finally, replacing our intensity assignment with a naive, scene-referenced Gaussian sampling (Gauss.) still reduces object-level performance.

Ablation on the components of $\mathcal{L}_{\text{OE}}$. We investigate the role of each component of the fine-tuning loss $\mathcal{L}_{\text{OE}}$ and report results in Table VI. We first evaluate the impact of the distance-weighting term, controlled by the parameter γ (see Eq. (6)). Adding this term produces some improvements on a few metrics, most notably RQ and UQ, with the best results obtained for $\gamma = 1$. We then add the OHEM, testing different values for the fraction of hardest examples $\beta \in \{0.7, 0.5, 0.3, 0.1\}$. The best results are obtained with $\beta = 0.3$, while both overly aggressive (e.g., $\beta = 0.1$) and weak mining (e.g., $\beta = 0.7$) lead to worse performance. The best results are achieved when both components are enabled, demonstrating their complementarity.

Training and inference results

Table VII details computational costs (on a AMD Ryzen 9 7950X, 2×RTX 4090 workstation). Offline occupancy pre-computation takes ≈ 15.1 ms per scene, and online pasting adds minimal training overhead compared to the core network training. Crucially, MSP adds just 0.12 ms during inference, allowing LExUS to perform joint inlier/outlier segmentation at virtually no extra cost.

TABLE VII: Detailed Pipeline Latency Breakdown

Stage	Component	Time (ms)	Total (ms)
Offline	Occ. Map Precomputation	15.1 ± 1.3	≈ **15.1**
Train (Pasting)	Load Data	36.5 ± 7.9	≈ **84.7**
	Find Location	22.1 ± 19.0	
	Dijkstra Intensity	26.1 ± 37.0	
Train (Network)	Forward Pass	278.2 ± 18.5	≈ **787.2**
	Loss w/ OHEM	89.0 ± 6.9	
	Backward Pass	420.0 ± 44.4	
Inference	Mask4Former Forward	158.5 ± 27.1	≈ **158.6**
	MSP Overhead	0.12 ± 0.00	

V. Conclusions

This work introduces LExUS, a novel outlier exposure (OE) strategy for open-set segmentation of LiDAR point clouds in driving scenarios. It proposes a new pipeline for injecting outliers into LiDAR scenes, generating mixed samples that are more effective for OE. In addition, it includes a tailored loss function that accounts for the specific characteristics of the LiDAR modality. We empirically show that the approach is general, as it applies across different post-hoc OOD scoring functions and sources of auxiliary data. Moreover, it preserves in-distribution performance while delivering substantial improvements in unknown segmentation.

Limitations: our method (and also other OE solutions) display a reduction in Segmentation Quality (SQ), which relates to the quality of the individual masks of segmented unknowns. This is possibly caused by the higher number of detected true positives, but further investigation may aim to improve upon this point.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 23,2026 at 08:52:32 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2026.3756298

Future works may explore different recipes to combine real and synthetic auxiliary data sources, explore the interplay of our OE strategy with newer models for LiDAR data that distill knowledge from RGB foundation models, and extend this work towards solutions that process multiple consecutive scans to exploit temporal information.

REFERENCES

[1] Z. Zhou, Y. Zhang, and H. Foroosh, "Panoptic-PolarNet: Proposal-free lidar point cloud panoptic segmentation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 13 194–13 203.

[2] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada, "EfficientLPS: Efficient LiDAR panoptic segmentation," in *IEEE Trans. on Robot. (T-RO)*, vol. 38, no. 3, 2021, pp. 1894–1914.

[3] F. Hong, H. Zhou, X. Zhu, H. Li, and Z. Liu, "LiDAR-based panoptic segmentation via dynamic shifting network," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 13 090–13 099.

[4] K. Yilmaz, J. Schult, A. Nekrasov, and B. Leibe, "Mask4Former: Masked transformer for 3D and 4D panoptic segmentation," in *IEEE Int. Conf. on Robot. and Automat. (ICRA)*, 2024, pp. 9418–9425.

[5] M. Sodano, F. Magistri, J. Behley, and C. Stachniss, "Open-world panoptic segmentation," *arXiv:2412.12740*, 2024.

[6] S. N. Rai, F. Cermelli, D. Fontanel, C. Masone, and B. Caputo, "Unmasking anomalies in road-scene segmentation," in *IEEE/CVF Int. Conf. on Comput. Vis. (ICCV)*, 2023, pp. 4037–4046.

[7] A. Nekrasov, M. Burdorf, S. Worrall, B. Leibe, and J. S. B. Perez, "Spotting the Unexpected (STU): A 3D LiDAR dataset for anomaly segmentation in autonomous driving," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025, pp. 11 875–11 885.

[8] S. Xu, P. Li, Q. Sun, X. Liu, Y. Li, S. Guo, Z. Wang, B. Jiang, R. Wang, K. Sheng, B. Zhang, L. Jiang, H. Zhao, and Y. Chen, "LiON: learning point-wise abstaining penalty for LiDAR outlier detection using diverse synthetic data," in *AAAI Conf. on Artif. Intell.*, vol. 39, no. 9, 2025, pp. 8951–8959.

[9] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv:1610.02136*, 2016.

[10] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, "Open-set recognition: A good closed-set classifier is all you need," in *Int. Conf. on Learn. Repr. (ICLR)*, 2022, pp. 10 398–10 423.

[11] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Int. Conf. on Mach. Learn. (ICML)*, vol. 48, 2016, pp. 1050–1059.

[12] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[13] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *Int. Conf. on Learn. Repr. (ICLR)*, 2019, pp. 2942–2959.

[14] M. Grcić, P. Bevandić, and S. Šegvić, "Densehybrid: Hybrid anomaly detection for dense open-set recognition," in *Eur. Conf. on Comput. Vis. (ECCV)*, 2022, pp. 500–517.

[15] S. N. Rai, F. Cermelli, B. Caputo, and C. Masone, "Mask2Anomaly: Mask transformer for universal open-set segmentation," *IEEE Trans. on Pattern Anal. & Mach. Intell. (TPAMI)*, vol. 46, no. 12, pp. 9286–9302, 2024.

[16] A. Delić, M. Grcic, and S. Šegvić, "Outlier detection by ensembling uncertainty with negative objectness," in *Brit. Mach. Vis. Conf. (BMVC)*, 2024, pp. 1–27.

[17] P. de Jorge, R. Volpi, P. K. Dokania, P. H. S. Torr, and G. Rogez, "Placing objects in context via inpainting for out-of-distribution segmentation," in *Eur. Conf. on Comput. Vis. (ECCV)*, 2024, pp. 456–473.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Eur. Conf. on Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[19] J. Cen, P. Yun, S. Zhang, J. Cai, D. Luan, M. Tang, M. Liu, and M. Yu Wang, "Open-world semantic segmentation for LiDAR point clouds," in *Eur. Conf. on Comput. Vis. (ECCV)*, 2022, pp. 318–334.

[20] R. Mohan, J. Hindel, F. Drews, C. Gläser, D. Cattaneo, and A. V., "Open-set LiDAR panoptic segmentation guided by uncertainty-aware learning," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (IROS)*, 2025, pp. 2224–2231.

[21] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "ShapeNet: An information-rich 3d model repository," *arXiv:1512.03012*, 2015.

[22] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *IEEE/CVF Int. Conf. on Comput. Vis. (ICCV)*, 2019, pp. 9297–9307.

[23] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 9404–9413.

[24] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate lidar semantic segmentation," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (IROS)*, 2019, pp. 4213–4220.

[25] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 12 697–12 705.

[26] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 9939–9948.

[27] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Advances in Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 17 864–17 875.

[28] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 1290–1299.

[29] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3D: Mask transformer for 3D semantic instance segmentation," in *IEEE Int. Conf. on Robot. and Automat. (ICRA)*, 2023, pp. 8216–8223.

[30] H. S. Miandashti and C. Brenner, "Out-of-distribution detection in LiDAR semantic segmentation using epistemic uncertainty from hierarchical GMMs," in *German Conf. on Pattern Recognit. (GCPR)*, 2025, pp. 615–628.

[31] Z. Li, Z. Xiang, J. Ao, J. West, and K. Khoshelham, "Relative energy learning for LiDAR out-of-distribution detection," *arXiv:2511.06720*, 2025.

[32] A. Zhang, C. Eranki, C. Zhang, J.-H. Park, R. Hong, P. Kalyani, L. Kalyanaraman, A. Gamare, A. Bagad, M. Esteva, and J. Biswas, "Toward robust robot 3-D perception in urban environments: The UT Campus object dataset," *IEEE Trans. on Robot. (T-RO)*, vol. 40, pp. 3322–3340, 2024.

[33] A. Xiao, J. Huang, D. Guan, K. Cui, S. Lu, and L. Shao, "PolarMix: A general data augmentation technique for lidar point clouds," *Advances in Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, pp. 11 035–11 048, 2022.

[34] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 761–769.

[35] T.-Y. Tseng, A. Nekrasov, M. Burdorf, B. Leibe, J. Stephany Berrio, M. Shan, and S. Worrall, "Panoptic-CUDAL technical report: Rural australia point cloud dataset in rainy conditions," *arXiv:2503.16378*, 2025.

[36] N. Nayal, M. Yavuz, J. F. Henriques, and F. Güney, "RbA: Segmenting unknown regions rejected by all," in *IEEE/CVF Int. Conf. on Comput. Vis. (ICCV)*, 2023, pp. 711–722.

[37] R. Chan, M. Rottmann, and H. Gottschalk, "Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation," in *IEEE/CVF Int. Conf. on Comput. Vis. (ICCV)*, 2021, pp. 5128–5137.

[38] H. Wang, W. Liu, A. Bocchieri, and Y. Li, "Can multi-label classification networks know what they don't know?" *Advances in Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, pp. 29 074–29 087, 2021.

[39] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, "The fishyscapes benchmark: Measuring blind spots in semantic segmentation," *Int. J. Comput. Vis. (IJCV)*, vol. 129, pp. 3119–3135, 2021.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 23,2026 at 08:52:32 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.