Engineering Applications of Artificial Intelligence 163 (2026) 112793

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Research paper

An improved graph attention network for semantic segmentation of industrial point clouds in automotive battery sealing nail defect detection

Wei Pan [a,*], Yuhao Wu [a,b,c], Wenming Tang [d], Qinghua Lu [b,c,**], Yunzhi Zhang [b,c,**]

[a] Department of R&D, OPT Machine Vision, No. 66 Xingfa South Road, Chang'An Town, Dong Guan, 523860, Guang Dong, China
[b] School of Mechatronic Engineering and Automation, Foshan University, Foshan, 528000, Guang Dong, China
[c] Guangdong Provincial Key Laboratory of Industrial Intelligent Inspection Technology, Foshan University, Foshan, 528000, Guang Dong, China
[d] School of Intelligent Manufacturing and Equipment, Shenzhen Institute of Information Technology, No. 2188, Longxiang Avenue, Longgang District, Shenzhen, 518172, Guang Dong, China

ARTICLE INFO

Keywords:
Defect detection
Automotive battery
Sealing nails
Point cloud semantic segmentation
Graph attention

ABSTRACT

Accurate defect detection in automotive battery sealing nails is vital for safety and reliability. Traditional methods combine two-dimensional (2D) vision for localization with three-dimensional (3D) vision for measurement, resulting in complex workflows and reduced efficiency. We propose Local Graph Attention for Semantic Segmentation (LGASS), an end-to-end 3D point cloud segmentation model. LGASS processes raw point cloud data from structured-light systems, performing simultaneous defect localization and geometric quantification in a single stage. By leveraging a graph attention mechanism in an encoder–decoder architecture, LGASS captures local geometric features and long-range dependencies, excelling on industrial metallic surfaces. Experiments show LGASS achieves 99.47% Overall Accuracy (OA), 92.37% mean Accuracy (mAcc), and 79.23% mean Intersection over Union (mIoU), offering a robust solution for automated sealing nail inspection.

1. Introduction

Sealing nails are critical components in automotive battery systems, acting as laser-welded plugs for electrolyte injection holes. However, welding may introduce defects such as bursts, molten beads, and pinholes that compromise safety and durability (Xu et al., 2023). Due to the highly reflective and textureless nature of metallic surfaces, conventional two-dimensional (2D) imaging techniques often struggle to detect such anomalies, requiring frequent lighting re-tuning and suffering from high false detection rates.

Three-dimensional (3D) point cloud inspection has emerged as a promising alternative, offering direct access to geometric cues like surface normals and curvature, which are particularly informative for identifying subtle surface irregularities (Wang et al., 2019a). However, many state-of-the-art segmentation networks (e.g., Swin3D (Yang et al., 2023), Point Transformer (Zhao et al., 2021)) rely heavily on RGB-D or intensity inputs, limiting their robustness when only geometric data is available—an often-encountered constraint in industrial scenarios due to specular reflections and noisy sensors.

To address these limitations, we propose **LGASS** (Local Graph Attention for Semantic Segmentation), a geometry-driven, end-to-end framework designed to detect surface defects directly from raw 3D point clouds (coordinates + normals). Unlike prior works such as GAC-Net or Point Transformer, which rely on appearance features or global context, LGASS introduces two novel modules tailored for industrial surface inspection:

- **Local Graph Attention Filter (LGAF):** An attention mechanism that prioritizes differences in normal vectors across local neighborhoods, enabling the detection of subtle geometric anomalies.
- **Spatial Attention Graph Pooling (SAG-Pooling):** A pooling strategy that selectively retains geometrically critical points during downsampling to preserve structural integrity.

To evaluate the method, we construct a new industrial dataset containing real-world point clouds of battery sealing nails across five defect types. LGASS achieves state-of-the-art performance in this geometry-only setting, with 99.47% overall accuracy (OA), 92.37% mean accuracy (mAcc), and 79.23% mean IoU (mIoU).

The objective of this study is to demonstrate that a purely geometry-based deep learning pipeline can effectively segment subtle surface defects in industrial point clouds, without reliance on color or intensity cues. Compared to traditional two-stage methods—where 2D

* Corresponding author.
** Correspondence to: Guangyun Road 33, Shishan Town, Nanhai District, Foshan City, 528225, Guangdong Province, China
E-mail addresses: vpan@foxmail.com (W. Pan), qhlu@fosu.edu.cn (Q. Lu), zhangyz@fosu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.engappai.2025.112793
Received 6 June 2025; Received in revised form 8 October 2025; Accepted 13 October 2025
Available online 21 October 2025

image-based defect detection is followed by 3D localization—the proposed LGASS offers an end-to-end solution that avoids error accumulation across stages, reduces reliance on fragile 2D vision under reflective conditions, and improves adaptability for industrial deployment.

In summary, our contributions are as follows:

1. We propose a novel normal-difference attention operator (LGAF) and a spatial pooling mechanism (SAG-Pooling) designed for geometry-only segmentation.
2. We build and release a high-resolution sealing nail defect dataset with five realistic defect classes and class imbalance.
3. We achieve state-of-the-art results under geometry-only input, outperforming GACNet, PointNet++, and Point Transformer.
4. We provide code and dataset online for reproducibility.[1]

The remainder of the paper is organized as follows. Section 2 reviews related literature. Section 3 details the LGASS framework. Section 4 presents the dataset and evaluation results. Section 5 concludes the paper.

2. Related work

This section surveys prior work in four areas: point-based architectures, voxel-based methods, graph-based and attention-enhanced networks, and geometric sensing for industrial inspection. We focus on these four categories as they encompass the dominant paradigms in 3D point cloud analysis relevant to industrial surface inspection tasks.

2.1. Point-based architectures

Point-based networks process raw 3D points directly. PointNet (Qi et al., 2017a) introduced shared multilayer perceptrons (MLPs) and symmetric pooling to achieve permutation invariance. PointNet++ (Qi et al., 2017b) builds on this by grouping points hierarchically to capture local context. PointNeXt (Qian et al., 2022) improves training strategies and scaling for higher accuracy. PointCNN (Li et al., 2018) learns an $\mathcal{X}$-transform to order points for standard convolution, while PAConv (Xu et al., 2021) dynamically assembles convolution kernels based on local geometry. These methods achieve strong results on general benchmarks but typically assume additional color or intensity channels, limiting robustness when only geometric data is available.

2.2. Voxel-based methods

Voxelization discretizes 3D space into regular grids and applies 3D CNNs. VoxelNet (Zhou and Tuzel, 2018) encodes each voxel's points via PointNet before applying a 3D CNN for end-to-end detection. SECOND (Yan et al., 2018) uses sparse 3D convolutions to process only occupied voxels, reducing computation. PV-RCNN (Shi et al., 2020) combines voxel-level and point-level features for high-precision detection. While effective, voxelization introduces quantization error and high memory costs, making it challenging for high-resolution industrial scans.

2.3. Graph-based and attention-enhanced networks

Graph networks represent point clouds as graphs, with nodes as points and edges capturing local relationships. DGCNN (Wang et al., 2019b) constructs a dynamic kNN graph per layer and applies EdgeConv to learn local geometry. RGCNN (Te et al., 2018) uses regularized graph convolutions for segmentation. GAT (Veličković et al., 2017) introduces self-attention on graphs, enabling learnable neighbor weighting. GAC (Wang et al., 2019a) tailors attention to point clouds by combining feature similarity and euclidean distance. Adaptive Graph Convolution (Zhou et al., 2021) and Attentive Filtering (Zhang et al., 2023) further refine graph topology and mask neighbor features. These methods excel at modeling fine geometric details but often rely on RGB-D inputs or intensity cues, reducing performance when only coordinates and normals are present.

Transformer-based architectures have been adapted to point clouds. Point Transformer (Zhao et al., 2021) computes self-attention among point features with relative positional encoding. PCT (Guo et al., 2021) and Stratified Transformer (Lai et al., 2022) introduce multi-scale attention to capture both local and global context. Swin3D (Yang et al., 2023) employs windowed attention for computational efficiency. While these models achieve state-of-the-art results on RGB-D datasets like S3DIS (Armeni et al., 2016) and ScanNet (Dai et al., 2017), their reliance on color or intensity limits robustness in metallic industrial environments.

2.4. Geometric sensing for industrial inspection

Recent work has explored specialized sensing and geometric analysis for defect detection. Li et al. (2025) used photometric stereo to detect subtle defects on battery top covers by analyzing reflectance-based normal distortions. Pan et al. (2024) proposed a convex analysis method to measure assembly gaps from 3D point clouds, demonstrating high precision in industrial settings. Lu et al. (2024) developed a generative model with asynchronous structured light to enhance 3D reconstruction quality under challenging lighting. These approaches focus on measurement or reconstruction rather than semantic segmentation.

2.5. Industrial defect inspection on point clouds

In the specific domain of automotive battery sealing nails, Xu et al. (2023) introduced a few-shot learning method combining 2D defect localization with 3D alignment. While it reduces annotation cost, it remains a multi-stage pipeline rather than an end-to-end 3D solution. To our knowledge, no existing work directly segments sealing nail defects in a single stage using only geometric input.

Summary and Positioning. While prior research has made notable progress in 3D semantic segmentation using point-based, voxelized, and attention-driven approaches, these methods often assume access to RGB-D or intensity channels and perform poorly when only geometric data is available. Moreover, industrial inspection tasks such as sealing nail defect detection require models that can operate under noisy, textureless conditions with high class imbalance. Our proposed LGASS addresses these gaps by (i) leveraging normal-difference attention to emphasize local geometric anomalies, and (ii) preserving spatial saliency through attention-guided pooling. This geometry-exclusive, single-stage framework is thus well-suited to the demands of industrial surface defect inspection.

3. Methods

In classic semantic segmentation datasets, objects within the same category typically exhibit high similarity, simplifying classification tasks. However, the sealing nail dataset presents a unique challenge: objects within the same defect category often display diverse geometric shapes, complicating feature learning. Despite this, defect types can be effectively differentiated using positional and height data, since nearly all defect areas exhibit significant variations in height and normal vectors. Within the defect region, only the highest and lowest points possess normal vectors perpendicular to the XZ-plane, as shown by N_1 in Fig. 1. The differences in normal vectors between adjacent points serve as a key indicator for defect identification. Consequently, in this study, we use the normal vector information from the raw point cloud as the primary network input, enhancing sensitivity to normal vector changes and improving feature learning efficiency.

[1] https://github.com/WillPanSUTD/LGANet

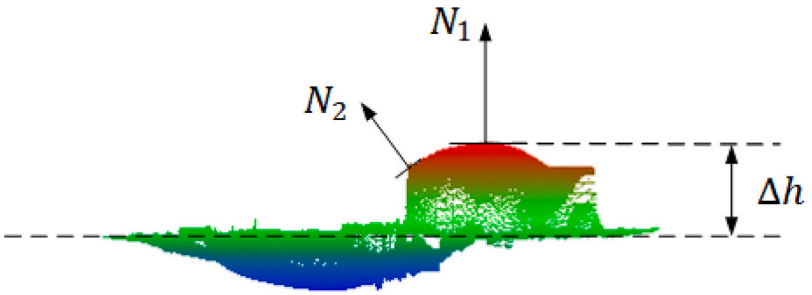

Fig. 1. Cross section of a sample defect with the normal vector and height information of a point.

3.1. Graph construction

We define the input of our model as $X = \{x_i \in \mathbb{R}^F, i = 1, 2, \ldots, N\}$, where x_i represents the feature of the ith point in the input point cloud, with feature dimension F, and N is the total number of input points. In this study, we use the surface normal vectors $(n_{x,i}, n_{y,i}, n_{z,i})$ of the raw point cloud as input, and set $F = 3$.

To effectively capture local structure, we transform the point cloud into a directed graph $G = (V, E)$, where each point corresponds to a vertex in $V = \{1, 2, \ldots, N\}$, and edges $E \subset V \times V$ are determined by the k-nearest neighbor (k-NN) relationships in Euclidean space (Fig. 2). Formally,

$$E = \{(i, j) \mid \|p_i - p_j\|_2 \leq \|p_i - p_{j(k)}\|_2, \quad \forall i, j \in V\}, \tag{1}$$

where p_i and p_j denote the 3D coordinates of points i and j, and $p_{j(k)}$ is the kth nearest neighbor of i, and $\|p_i - p_j\|_2$ represents the Euclidean distance between points i and j. This directed graph construction ensures that each point connects to its k nearest neighbors, capturing local geometric context.

3.2. Local graph attention filter

We designed a novel operator for extracting local features from point clouds, as illustrated in Fig. 3. Unlike the graph attention convolution of GACNet, which computes attention solely based on feature similarity, our operator calculates the correlation between vertex and edge features to generate attention weights. These weights are then used to aggregate convolution-based features accordingly.

Inspired by previous studies on self-attention mechanisms in 3D point clouds (Zhao et al., 2021), we define the correlation between a central point and its neighbors using the following Eq. (2).

$$C_{ij} = \phi(x_i) - \psi(x_i - x_j), \quad \text{for each pair } (i, j) \in \mathcal{E} \tag{2}$$

Here, x_i represents the vertex feature of the central point i, x_j is the vertex feature of neighbor j, and $x_i - x_j$ denotes the edge feature between them. The functions ϕ and ψ are learnable linear projection functions implemented as multilayer perceptrons (MLPs), which project the vertex and edge features into a metric space, respectively. The correlation matrix $C \in \mathbb{R}^{N \times N \times c}$ captures feature-level differences between a point and its neighbors, where N is the number of points and c is the feature dimension.

However, feature similarity alone is insufficient for learning effective representations in dense industrial point clouds. Spatial and geometric structural information also play a crucial role. As the receptive field expands in deeper layers, irrelevant distant neighbors may introduce noise. To mitigate this, we reformulate the adjacency matrix based on Euclidean distances:

$$\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{3}$$

$$\sigma_p = M_p(p_i - p_j) \tag{4}$$

Where $A \in \mathbb{R}^{N \times N}$ is the distance-weighted adjacency matrix defined in Eq. (5), D is the diagonal degree matrix of A (i.e. $D_{ii} = \sum_j A_{ij}$), and $\hat{A}$ is the normalized adjacency matrix. The δ_p represents the spatial encoding, where p_i and p_j are the spatial coordinates of points i and j, and M_p is an MLP that embeds the relative spatial position into the attention computation.

$$A_{ij} = \begin{cases} e^{-\|p_i - p_j\|^2}, & \text{if } (i, j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Here, p_i and p_j are spatial coordinates, D is the diagonal degree matrix, and M_p is an MLP used to embed relative position into the attention computation.

The attention weight is then computed as:

$$e_{ij} = M_a(C_{ij} \cdot \hat{A}_{ij}) \tag{6}$$

Here, M_a is a multilayer perceptron that maps the combined correlation and adjacency information to a scalar attention score. The attention weight is normalized using the softmax function:

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{(i,l) \in \mathcal{E}} \exp(e_{il})} \tag{7}$$

Finally, the output feature for point i is calculated as:

$$x_i' = \sum_{j=1}^{k} a_{ij} \odot (M_v(x_i - x_j) + \sigma_p), (i, j) \in \mathcal{E} \tag{8}$$

Where $\odot$ denotes element-wise multiplication, M_v is an MLP that transforms the edge feature, The summation is over the k nearest neighbors in the graph.

In designing LGAF, we emphasize differences in surface normals between neighboring points, as these are strong indicators of defects like pinholes or bursts—especially on reflective metallic surfaces where color cues are unreliable. Surface normals provide robust geometric descriptors for detecting such irregularities.

In defect detection tasks, each defect type typically exhibits distinctive morphological characteristics. As illustrated in Fig. 4, although the depth of defective regions often deviates significantly from the reference plane, the coordinates of points near defect boundaries show only minimal variation between normal and defective areas. This poses challenges for networks that rely solely on coordinate-based features, as such subtle geometric variations may not be easily captured.

To address this, LGAF is specifically designed to emphasize differences in surface normals between neighboring points—features that more directly characterize local surface irregularities, such as pinholes or bursts commonly observed on metallic surfaces like sealing nails. Compared to raw coordinates, normal vectors offer more discriminative and robust geometric cues for identifying these defects.

As shown in Fig. 4, points p_0 and p_1 lie close to each other along a defect boundary and exhibit negligible coordinate differences. However, their surface normals differ significantly in direction, reflecting the underlying geometric anomaly. By leveraging these normal vector differences as input features, the LGAF operator assigns higher attention weights to points near defect boundaries, thereby facilitating more efficient and accurate learning by the network.

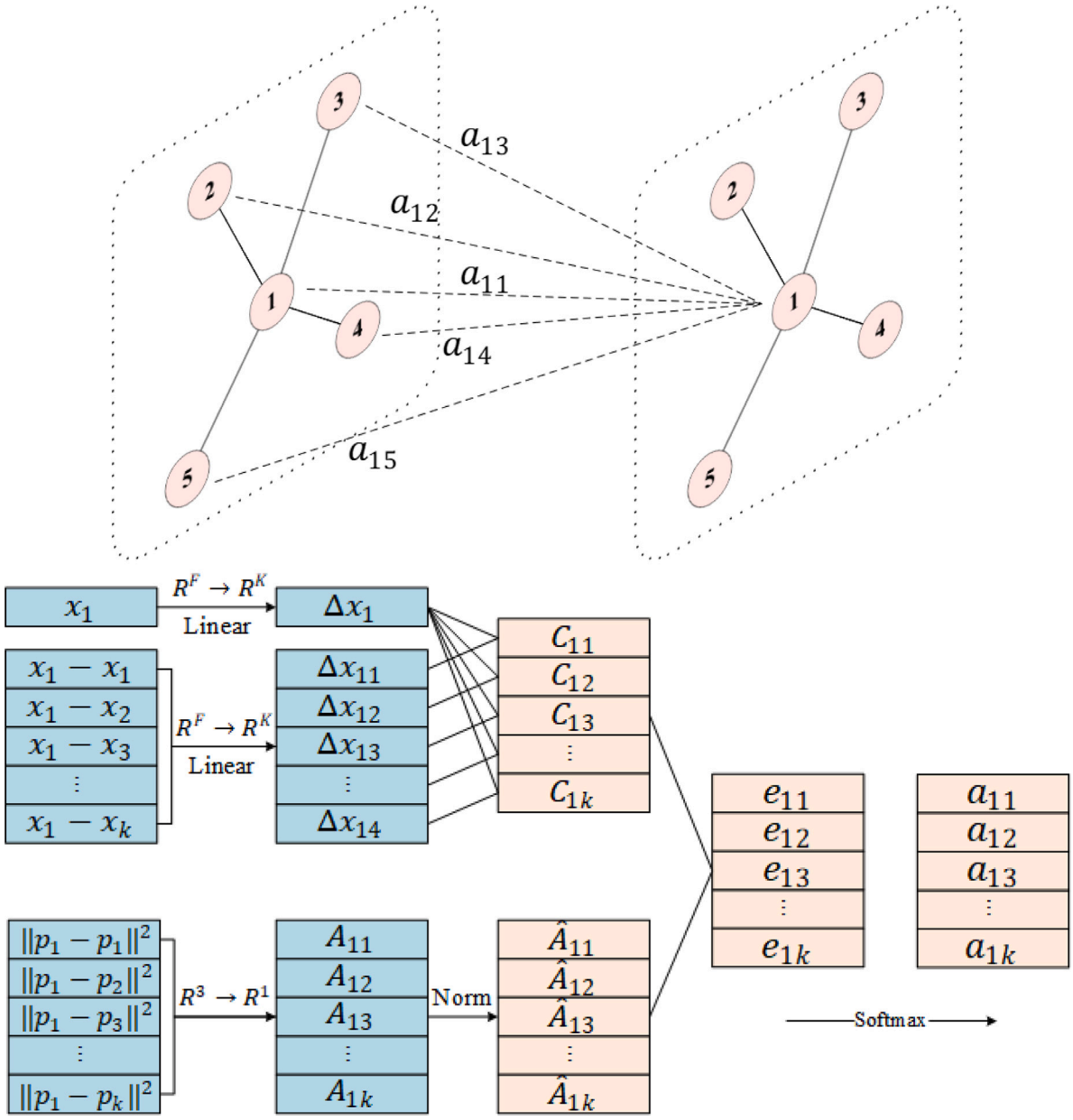

Fig. 2. The LGAF operation. LGAF calculates the output by aggregating the features of each vertex and edge.

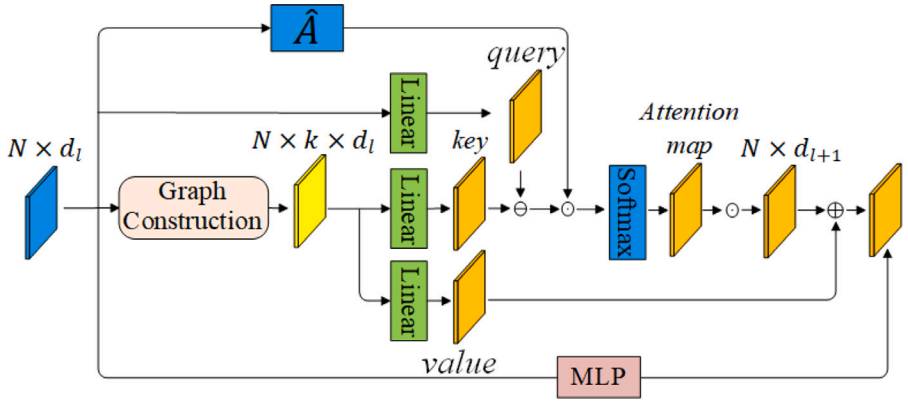

Fig. 3. The detailed structure of the LGAF block.

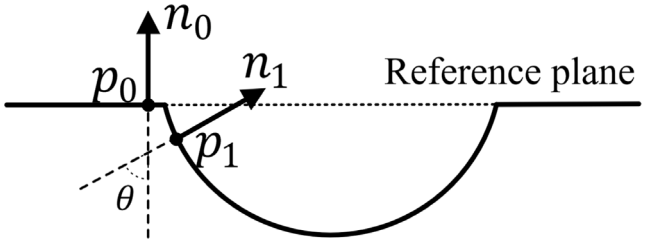

Fig. 4. The principle of leveraging surface normal changes for defect detection. (a) Defect cross-section: Although points p_0(normal surface) and p_1(defect boundary) are spatially close, the local surface orientation, represented by their normal vectors (arrows), changes drastically. It is evident that when $\theta \in [0, \pi]$, a larger value of θ results in a greater value of $\|n_1 - n_0\|^2$, where $\|n_1 - n_0\|^2 = 2(1 - cos\theta)$. (b) In our LGAF operator, this directional difference is quantified by the difference vector $n_1 - n_0$. A large difference magnitude (indicating low cosine similarity) directly contributes to a higher correlation score C_{ij} and, consequently, a higher attention weight a_{ij}, enabling the network to focus on critical boundary regions.

3.3. Spatial attention graph pooling

In contrast to GACNet, which employs standard graph pooling during downsampling and overlooks spatial feature significance, we propose a Spatial Attention Graph Pooling (SAG-Pooling) module that emphasizes spatial context. This allows the network to retain geometry-critical points during hierarchical subsampling. The structure is illustrated in Fig. 5.

SAG-Pooling assigns weights to neighboring points based on their spatial positions. The mechanism operates through four sequential steps that combine spatial relationships and geometric characteristics to identify the most salient points for retention during pooling.

First, the coordinates of the central point p_i and its neighbor p_j are projected into a shared feature space using learnable linear functions φ_q and φ_k. This transformation allows the model to represent spatial relationships in a format suitable for attention computation.

An MLP(M_w) then processes the difference between these projected features to compute a preliminary attention score $\hat{r}_{ij}$. This score represents the raw spatial importance between points i and j before normalization. The scores are subsequently normalized across all neighbors using a softmax function, ensuring they sum to 1 and can be interpreted as probability distributions:

$$\hat{r}_{ij} = M_w\left(\varphi_q(p_i) - \varphi_k(p_j)\right), (i,j) \in \mathcal{E} \tag{9}$$

$$r_{ij} = \frac{\exp(\hat{r}_{ij})}{\sum_{(i,l)\in\mathcal{E}} \exp(\hat{r}_{il})} \tag{10}$$

Our approach combine these spatial attention weights with geometric information derived from surface normals. For each point pair (i, j), we compute the Euclidean distance between their surface normal vectors n_i and n_j. This normal difference $\|n_i - n_j\|$ quantifies the local geometric variation, with larger values indicating more significant surface orientation changes.

The final score β_j is obtained by multiplying the spatial attention weight r_ij with this normal difference:

$$\beta_j = r_{ij} \cdot \|\vec{n}_i - \vec{n}_j\|, \quad \text{for each pair } (i,j) \in \mathcal{E} \tag{11}$$

This multiplicative combination ensures that points receive high scores only when they are both spatially important and located in geometrically distinctive regions (e.g., defect edges or ridges).

The top-k selection process is implemented by first ranking all points in the point cloud in descending order based on their computed β scores, which reflect each point's contribution or distinctiveness. A predetermined pooling ratio k (typically between 0.5 and 0.8) then determines the proportion of points to retain; specifically, the top $(k \times N)$ points are selected, where N is the total number of points in the current point cloud. Only the features of these selected points are propagated to the next network layer, while the features of lower-ranked points are discarded. This selective retention mechanism effectively preserves geometrically distinctive points that are critical for tasks such as defect detection or shape analysis, while progressively eliminating less informative points to optimize computational efficiency.

The final pooled feature of point i is then:

$$F_i = \sum_{(i,l)\in\mathcal{E}} r_{ij} \odot \varphi_v(F_{ij}) \tag{12}$$

Here, ϕ_v is a linear projection applied to the feature representation F_{ij} of neighboring point j.

The key advantage of SAG-Pooling is its ability to preserve spatially informative features during point reduction. This is particularly important for small or subtle defects, where standard pooling methods might discard crucial geometric cues. By applying spatial attention, we ensure that the most relevant structural information is retained for accurate segmentation.

3.4. Graph convolution

We employed GCN blocks in the upsampling stage to aggregate point features. When only normal vectors are input to the model, spatial information is likely to be neglected during the upsampling stage. Previous research (Zhou et al., 2021) tackled this by designing dynamic convolution kernels to model the link between spatial and feature information. Instead, we adopted a simpler strategy: using positional encoding to model spatial information and highlighting it during the upsampling process. The standard graph convolution formula is as follows (Wang et al., 2019b):

$$x_i' = \triangle\, h_\Theta(x_i, x_i - x_j) \tag{13}$$

Where $\triangle$ denotes a feature aggregation operation for graph features, and h_Θ is a non-linear function to project the vertex and edge into metric space. Specifically, max pooling is used for feature aggregation, and non-linear functions are implemented through convolution operations and non-linear activation functions. The general formula of graph convolution is as follows:

$$x_i' = \max_{j\in\{1,2,\ldots,k\}} \mathrm{ReLU}\left(\theta \cdot x_i + \varphi \cdot (x_i - x_j)\right) \tag{14}$$

Unlike standard graph convolutions, we adopted a simple method to model spatial information in graph convolution. Specifically, we used a multilayer perceptron (MLP) to project coordinate information into the feature space and added this to the point features. Then, we formalized the graph convolution in our model as follows:

$$x_i' = \max_{j\in\{1,2,\ldots,k\}} \mathrm{ReLU}\left(\theta \cdot x_i + \varphi \cdot (x_i - x_j) + \sigma\right) \tag{15}$$

$$\sigma = M_n(p_i - p_j) \tag{16}$$

Where p_i and p_j represent the positions of the corresponding points in the coordinate space, M_n indicates the applied multilayer perceptron.

3.5. Network architecture

Our LGASS model adopts a U-Net style encoder–decoder structure for semantic segmentation of sealing nails surface defects. As illustrated in Fig. 6, our model, inspired by U-Net (Ronneberger et al., 2015), comprises a classifier and a feature extractor. The encoder uses four LGAF blocks, each followed by an SAG-Pooling layer that reduces the point count to one-quarter (16,384 to 256) and doubles the feature dimension (3 to 512). The decoder employs four GCN blocks with nearest-neighbor upsampling, increasing the point count fourfold per layer and halving the feature dimension, culminating in a 6-dimensional output (one per class). Skip connections concatenate encoder and decoder features at matching resolutions to preserve local details. In the encoder, after each

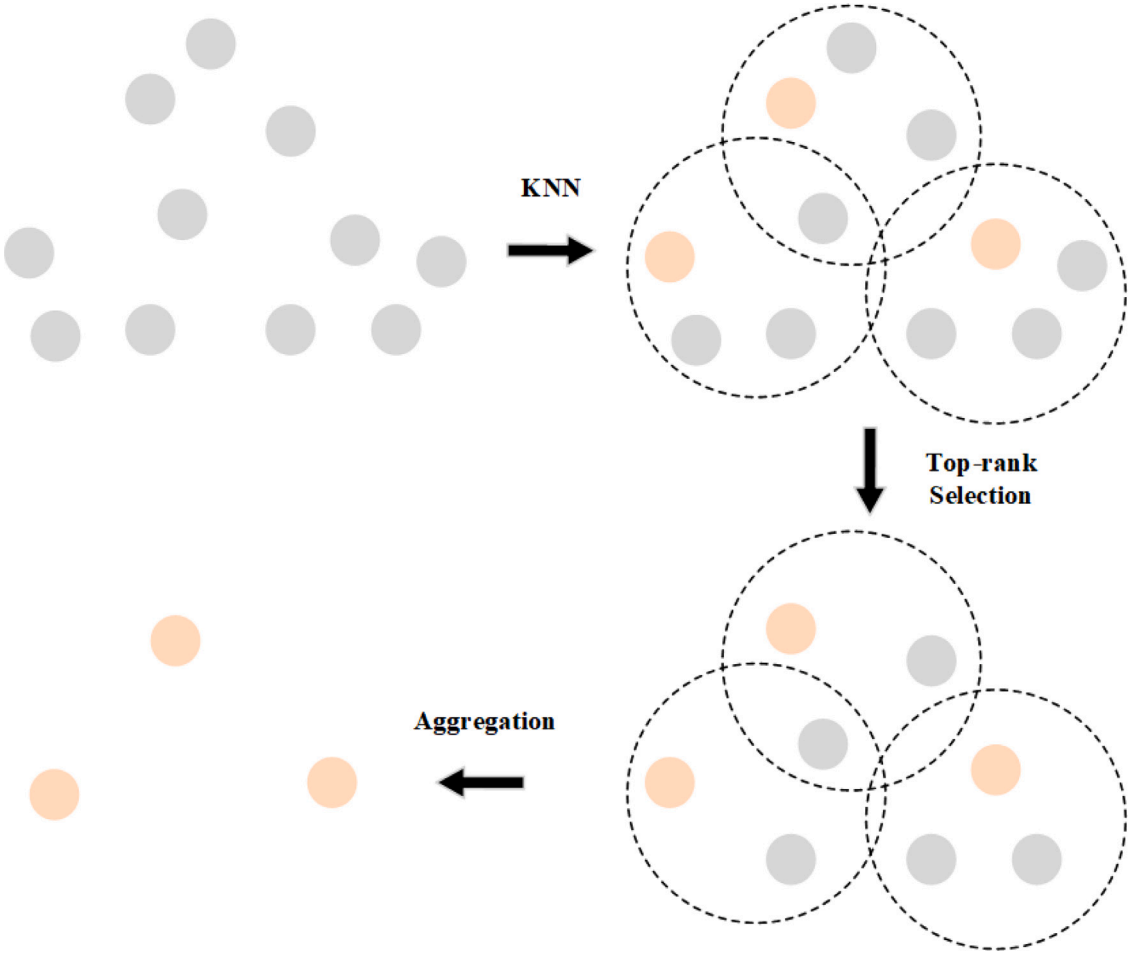

Fig. 5. The detailed structure of the SAG Pooling block.

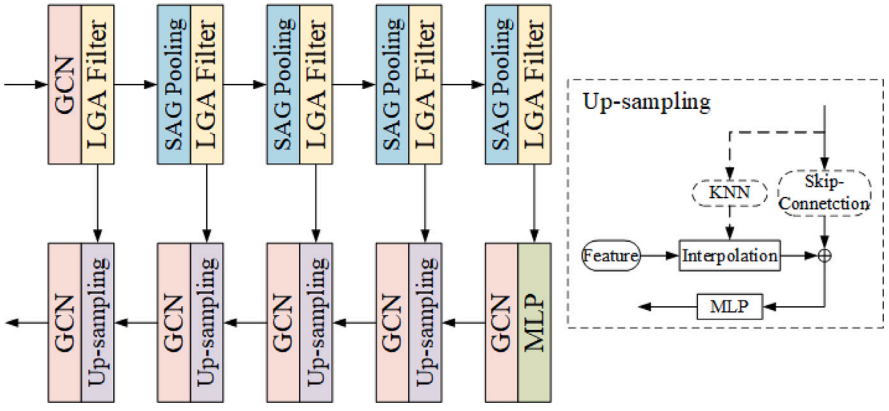

Fig. 6. The detailed structure of the LGASS.

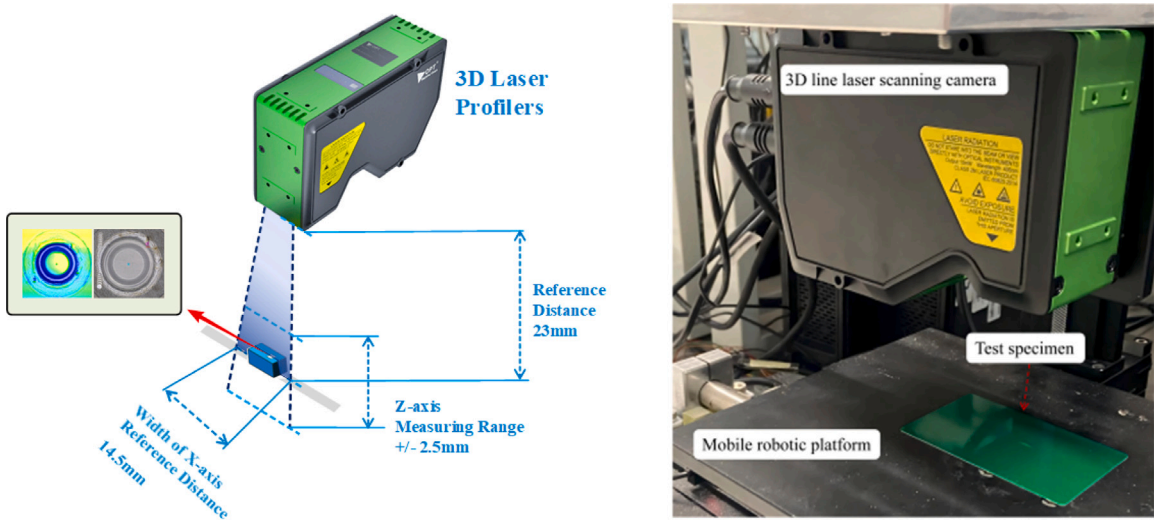

Fig. 7. Schematic diagram and dimension of the sealing nails, following the standard layout proposed in Konovalenko et al. (2022).

SAG Pooling layer, the number of output points is reduced to a quarter of the original, while the feature dimension is doubled. In the decoder, after each Up-sampling layer, the number of output points is increased to four times the original, while the feature dimension is halved.

4. Results and discussion

4.1. Dataset

4.1.1. Imaging system and dataset composition

This paper collects a point cloud dataset of automotive battery sealing nails to evaluate the practicality of the model. The 3D line laser scanning system is shown in Fig. 7. The image acquisition and imaging module consists of a 3D line laser scanning camera with a mobile robotic platform and a test specimen. The battery is mounted on the mobile robot platform, and the calibrated 3D line laser scanning camera projects laser stripes onto the sealing nail area of the battery. Through synchronized image data acquisition during planar scanning trajectories, surface data of the sealing nail specimen are obtained.

All 3D data in our dataset were acquired using a Laser Profiler as a 3D scanning system (OPT-LPC20 Laser Profiler), configured in linear scanning mode. The system uses a blue laser stripe (wavelength 405 nm) with a profile acquisition rate of 10,000 profiles per second, ensuring high fidelity on metallic reflective surfaces.

During scanning, the sealing nails were placed on a precision rotary stage to ensure uniform coverage. The scanning speed was set to 10 mm/s, and each profile captured 3000 points across the horizontal axis, resulting in a lateral resolution of 0.05 mm and vertical depth accuracy of < 0.01 mm. All scans were manually verified for completeness and noise artifacts.

The scanning mode can significantly influence point cloud quality. For example, increasing scanning speed may reduce point density, leading to loss of fine defect features, while improper lighting or sensor saturation may introduce normal estimation errors. To mitigate these effects, we optimized the exposure and laser intensity settings and discarded overexposed frames during preprocessing. The resulting point clouds contain only geometry (coordinates + normals) and are preprocessed using a modified RANSAC-based segmentation strategy to extract the sealing nail region.

Fig. 8 also depicts the real image and point cloud of the battery cover, and the marked area represents the defect of sealing nails. The definition of defects are displayed in Table 1.

Based on distinct welding methods and production processes, the power battery sealing nail point cloud dataset acquired during data collection contains two nail types: flat-top nails and recessed nails. During production, the acquisition equipment typically captures point cloud data from the sealing nail and its surrounding top cover within a rectangular area of 13 mm–16 mm in side length. This study aimed to accurately classify areas exhibiting abnormal morphological features within the seal nail weld region.

According to industry standards and regulatory guidance (Yu et al., 2023), defects such as burst, pinhole, and pit are considered unacceptable for laser-welded sealing nails due to their impact on battery safety and airtightness.

The distribution of defect samples in the dataset is shown in Table 1. The dataset comprises 271 normal samples and 658 defective samples across five defect types (e.g., burst: 170, pit: 112), with each point cloud averaging 100,000 points and a severe class imbalance (defect points 1,000 per sample). We note that all defect samples were collected from actual inspection lines and reflect naturally occurring anomalies rather than artificially created defects. Although the dataset annotations are based on visual non-destructive testing (NDT) techniques, future work could involve cross-validation with X-ray or ultrasonic inspection results to further confirm ground truth reliability.

4.1.2. Data preprocessing

The morphology of defects in sealing nails exhibits significant variation, making localization and identification of specific types difficult, and challenging to define their morphological boundaries. Furthermore, the welding area of the sealing nail is relatively small, making it hard to distinguish. To prevent interference from the battery cover or background data points in subsequent defect detection algorithms, it is necessary to isolate the sealing nail welding area targeted for inspection. To achieve this, this paper employs the Random Sample Consensus (RANSAC) algorithm as part of the preprocessing workflow for sealing nail point cloud data to extract the sealing nail welding region. The primary approach involves using the RANSAC algorithm to fit a reference plane for both the sealing nail and the battery cover patch. Subsequently, the extent of the welding region is determined based on the spatial position of the sealing nail. Fig. 9 illustrates the detailed processing workflow through the raw data of sealing nails.

To improve the computational efficiency of the algorithm, the seed point selection strategy of the RANSAC algorithm was improved. Considering that spatially adjacent points within the sample data are more likely to belong to the same reference plane, three points are randomly selected within a neighborhood satisfying specific conditions to serve as seed points for fitting the plane model. Specifically, a point is randomly selected from the original point cloud dataset. A k-d tree is then utilized

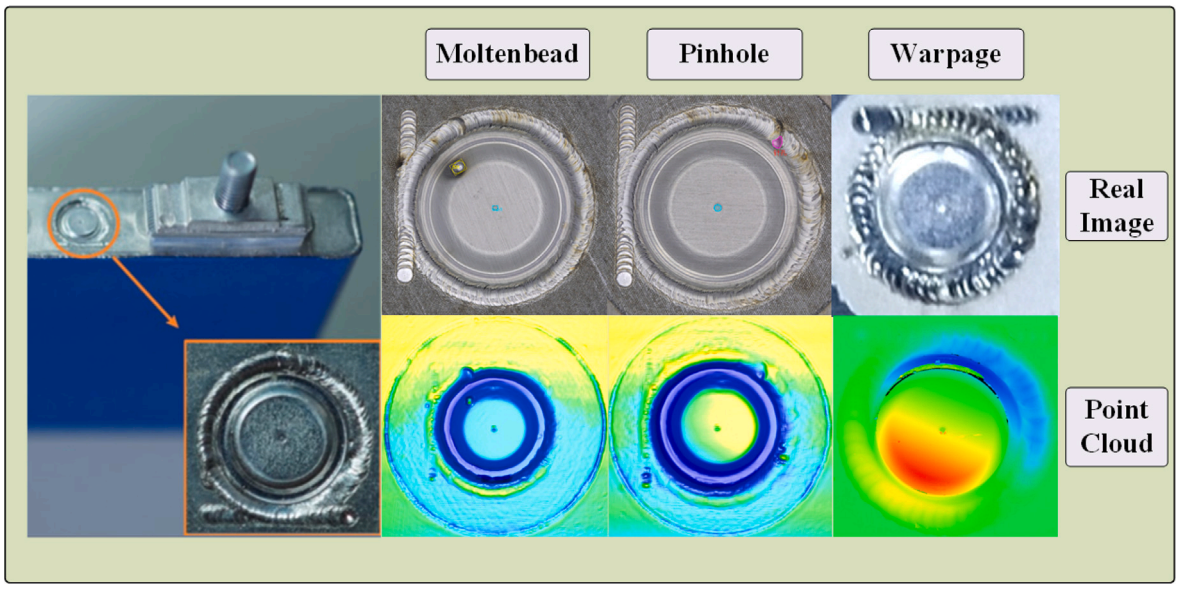

Fig. 8. Real images and point cloud representations of automotive battery sealing nails, highlighting defect examples.

Table 1
Defect definitions and corresponding quantities in the sealing nail dataset.

Category	Definition	Quantity
Normal	No visible or measurable surface defects.	271
Burst	Protrusion in the weld seam, area $\geq 0.02\,mm^2$, height $\geq 0.08\,mm$.	170
Pit	Depression in the weld seam, area $\geq 0.02\,mm^2$, depth $\geq 0.08\,mm$.	112
Warpage	Bulge along sealing nail, area $\geq 0.2\,mm^2$, height $\geq 0.1\,mm$.	132
Moltenbead	Bead-like protrusion along weld, area $\geq 0.02\,mm^2$, height $\geq 0.08\,mm$.	102
Pinhole	Small depression in weld zone, area $\geq 0.01\,mm^2$, depth $\geq 0.02\,mm$.	142

to find a set of points $S = \{p_1, p_2, \ldots, p_n\}$ lying within a spherical neighborhood of radius r around this point. The corresponding set of surface normals for these points is $N = \{n_1, n_2, \ldots, n_n\}$. Defining the reference normal vector (typically vertical) as $n_v = \{0, 0, 1\}$, if the average value γ (given by Eq. (17)) of the cosine of the angle between each point's normal and n_v exceeds a specified threshold, it indicates a high probability that points within this neighborhood belong to either the sealing nail or the battery top cover patch.

$$\gamma = \sum_{i=1}^{n} n_v \cdot n_i \tag{17}$$

The threshold for γ is set to 0.90, meaning that only neighborhoods with $\gamma > 0.90$ are considered for seed point selection. This method is chosen based on established practices in 3D point cloud processing (Zou and Pan, 2021), where threshold in the range of 0.85–0.95 are commonly used to ensure strong normal alignment indicative of planar surfaces. At this threshold, the corresponding angle between normals is approximately 25°, which provides an optimal balance between rejecting noisy points while retaining legitimate planar regions. This value is particularly suitable for industrial surfaces like sealing nails and battery covers where high planarity is expected. Subsequently, three points are randomly selected from this qualified neighborhood to serve as the initial seed points.

The flowchart of the improved RANSAC algorithm is shown in Fig. 10, which details the preprocessing for sealing nail point cloud data and the design of algorithmic hyperparameters. Additionally, 50 sealing nail samples were selected to compare the effectiveness of the proposed algorithm with region growing and Euclidean clustering algorithms. Experimental results were evaluated using precision, recall, F1-score to assess the comprehensive performance of the four algorithms. To address industrial requirements for computational efficiency, we also measured the average processing time for each algorithm on a standard computing platform (Intel i5-10700K CPU, 32 GB RAM).

Table 2
Comparison of segmentation performance of different segmentation algorithms.

Algorithms	Precision (%)	Recall (%)	F1-score (%)	Latency (s)
RANSAC	76.24	82.19	79.10	6.82
Region growing	83.10	81.57	82.33	8.73
Clustering	36.23	99.73	53.15	3.45
Ours	82.44	93.76	87.74	6.14

Data in Table 2 demonstrate that the proposed algorithm achieves the highest F1-score and accuracy, exhibiting the best comprehensive performance. Importantly, our improved RANSAC method also shows superior computational efficiency, with processing time reduced by approximately 65% compared to region growing and 80% compared to Euclidean clustering. This balance between accuracy and speed makes our method particularly suitable for industrial applications where real-time processing is critical.

The efficiency improvement can be attributed to our optimized seed point selection strategy and the integration of normal vector-based filtering, which reduces the number of iterations required for model

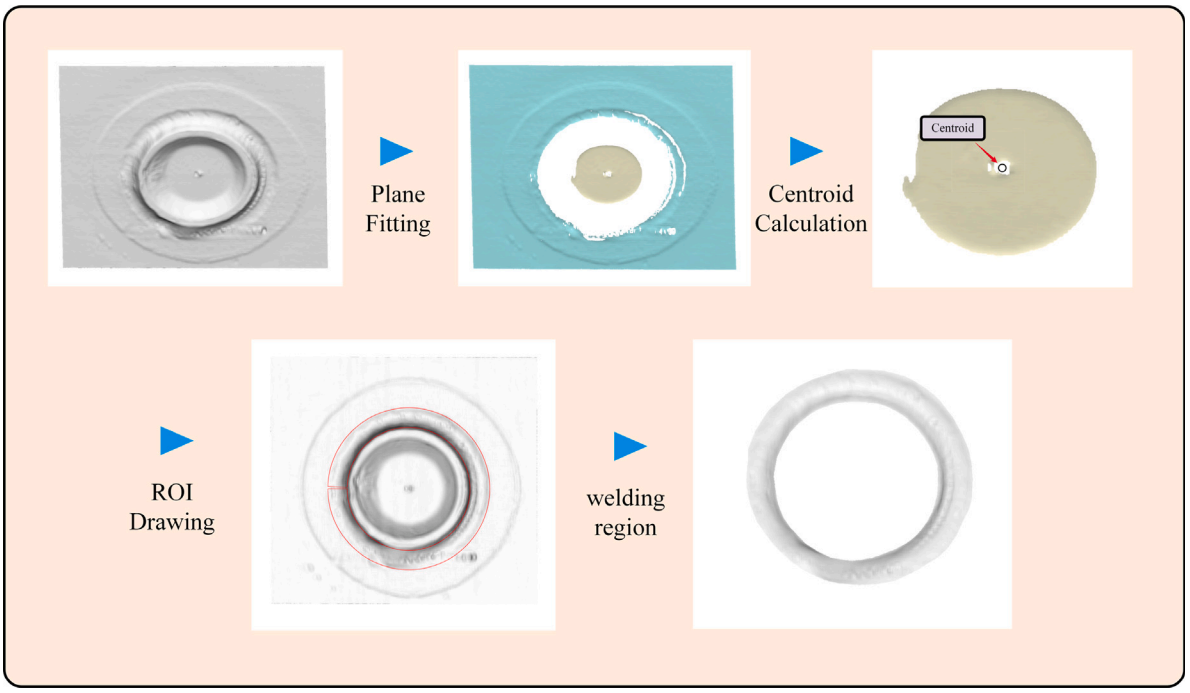

Fig. 9. Schematic diagram of the welding region extraction process. RANSAC is employed to fit the reference planes of both the sealing nail and the battery cover. Using the centroid position of the sealing nail as the geometric center, a circular Region of Interest (ROI) is drawn. This ROI precisely delineates the weld region for subsequent detection.

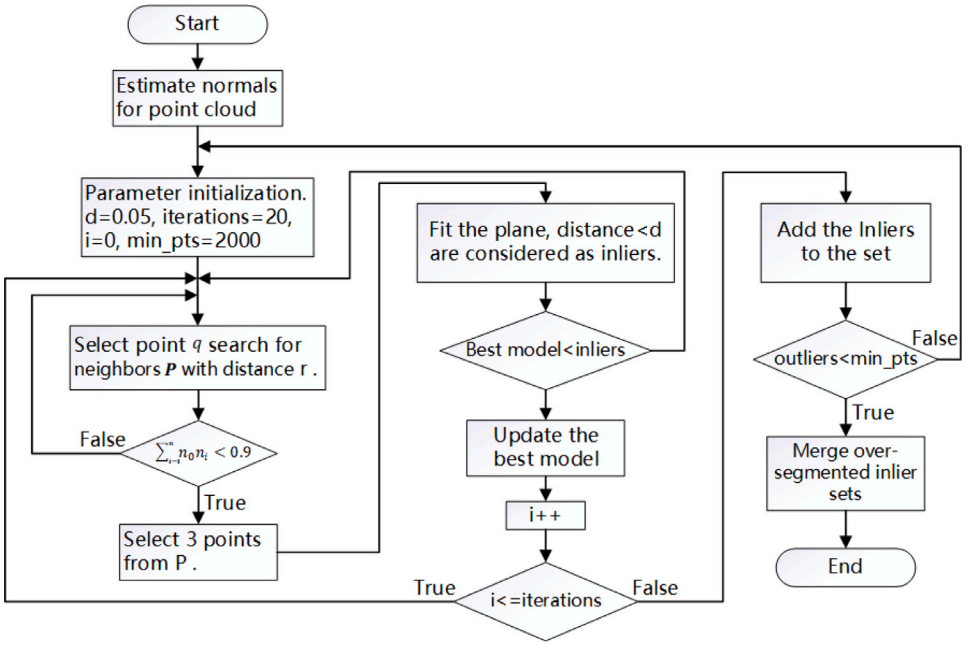

Fig. 10. Flowchart of the Improved RANSAC Algorithm.

convergence. This aligns with findings in RANSAC optimization literature, where preprocessing and intelligent sampling can significantly accelerate computation without sacrificing accuracy

As illustrated in Fig. 11, the segmentation results of the sealing nail point cloud data using the RANSAC algorithm demonstrate that this method effectively extracts the target sealing nail point set from the samples under inspection. By calculating the centroid of the point set, the positions of the geometric centers for both the sealing nail and the welding area can be confirmed, thereby enabling the extraction of the welding region. The computational efficiency achieved through our improvements ensures that this process can be completed within practical time constraints for industrial inspection systems.

4.2. Loss function

The loss function, also known as the cost function, is utilized in machine learning and deep learning to quantify the discrepancy between a model's predictions and the ground truth values. Its primary role is to measure prediction error, guiding the model to adjust its parameters during training to improve prediction accuracy. The loss

Fig. 11. Example of sealing nail region segmentation using the RANSAC algorithm.

function value iteratively decreases during training, guiding the model to gradually align predictions closer to the true values. When the loss function decreases and stabilizes, it indicates that the training process is largely complete. A smaller, stable loss value signifies higher prediction precision.

Detecting defects in sealing nail point clouds requires segmenting multiple types of defect regions, falling under the category of a multi-class semantic segmentation problem. Class imbalance must therefore be addressed. If data points belonging to a specific class constitute a large proportion of the dataset, the semantic segmentation network tends to learn parameter sets biased towards predicting that dominant class during iterative training. This occurs because the dominant class contributes more significantly to the overall loss function value. In this scenario, the network learns the overrepresented classes adequately but fails to sufficiently learn the characteristics of defects represented by fewer samples. This compromises the detection accuracy, thereby adversely affecting the overall learning performance of the model.

To mitigate the interference from classes with a higher proportion of data points on defect segmentation, this paper optimizes the calculation formula of the loss function. The optimization involves assigning corresponding weighting coefficients to different categories within the loss function. The method for generating these weighting coefficients is given by Eq. (18).

$$\hat{w} = \left[\left(\frac{n_{max}}{n_1} \right)^{1/2}, \left(\frac{n_{max}}{n_2} \right)^{1/2}, \ldots, \left(\frac{n_{max}}{n_k} \right)^{1/2} \right] \tag{18}$$

$$w_i = \frac{\hat{w}_i}{\sum_{i=1}^{k} \hat{w}_i} \tag{19}$$

where n_{max} represents the number of data points in the category with the largest proportion in the data set, n_1, n_1, …, n_k represents the number of data points in each category, and k is the number of defect categories. Finally, the optimized loss function is defined as follows:

$$Loss = -\sum_{i=1}^{k} w_i log \frac{\exp(x_i)}{\sum_{l=1}^{k} \exp(x_l)} y_i \tag{20}$$

where y_i is the true label, x_i is the predicted score of the ith category of the observed sample x, and after normalization by the softmax function, it is the probability that the observed sample x belongs to category i.

4.3. Experimental setup

The experimental environment is configured as follows: operating system Ubuntu 22.04, GPU NVIDIA GeForce RTX 4070, PyTorch 2.2.1, and CUDA 12.1. The model performance metrics include mean Accuracy (mAcc), mean Intersection over Union (mIoU), and Overall Accuracy (OA). The training hyperparameter settings are shown in Table 3.

Table 3
Experimental setting.

Parameter	Value
Epoch	300
Batch size	8
Number of points	16 384
Optimizer	AdamW
LR	0.01
LR decay	Multi step

4.4. Comparison result

We compared LGASS against state-of-the-art methods on the sealing nail dataset (Table 4). To ensure statistical reliability, all experiments were repeated 5 times with distinct random seeds (42, 123, 456, 789, 1024). Reported results are averaged values with standard deviations calculated across runs. To fully capture the variability, we also report the maximum and minimum values observed over these runs. LGASS achieves 99.43 ± 0.05% OA (range: 99.40%–99.48%), 92.16 ± 0.08% mAcc (range: 92.08%–92.20%), and 79.26 ± 0.35% mIoU (range: 78.97%–79.61%), surpassing PTv3 (75.13% mIoU) and OA-CNN (74.04% mIoU) by more than 4% in mIoU. The narrow ranges between minimum and maximum values across all metrics confirm the stability and consistency of LGASS under different initialization conditions. This improvement stems from its focus on normal-based features, which are robust to the reflective surfaces of sealing nails, unlike color-reliant methods. For example, LGASS excels at detecting subtle defects like pinholes (65.23% IoU), where geometric cues are critical. In Table 4, our method outperforms several previous methods (such as Ptv3 (Wu et al., 2024), OA-CNN (Peng et al., 2024)) in the sealing nail defect detection task. The rendered visualizations are shown in Fig. 12.

The robustness of LGASS on metallic surfaces stems from its reliance on purely geometric features, particularly surface normals, which remain relatively stable across lighting conditions. By computing attention weights based on normal differences, LGAF enhances feature representation at geometric discontinuities. Furthermore, SAG-Pooling ensures that sharp surface variations are retained during feature downsampling, making the model less susceptible to over-smoothing and missing fine defects such as pinholes or microcracks.

4.5. Ablation experiments

We conduct ablation experiments on the sealing nail dataset to isolate the contributions of LGAF and SAG-Pooling (Table 5).

- **Experiment I (Baseline):** Uses GCN blocks for local feature learning and standard graph pooling.
- **Experiment II:** Replaces GCN with LGAF (no SAG-Pooling).

Table 4
Accuracy comparison of different methods on the sealing nail dataset.

Model	OA	mAcc	mIoU	Burst	Pit	Moltenbead	Warpage	Pinhole
GACNet	96.05	87.84	69.41	56.02	67.83	54.54	86.96	59.94
PointNet	83.29	72.83	46.21	32.27	34.96	19.97	65.37	36.04
PointNet++	89.33	84.65	62.67	47.91	59.32	42.37	81.64	46.53
PTv1	98.77	90.96	72.83	68.54	62.83	58.37	89.67	58.37
PTv3	99.03	**94.96**	75.13	**77.32**	67.98	60.44	89.17	56.21
OA-CNN	98.72	91.13	74.04	64.89	66.91	64.87	88.37	60.04
LGASS(Ours)	**99.43**	92.16	**79.26**	75.82	**72.76**	**70.45**	**91.84**	**65.23**

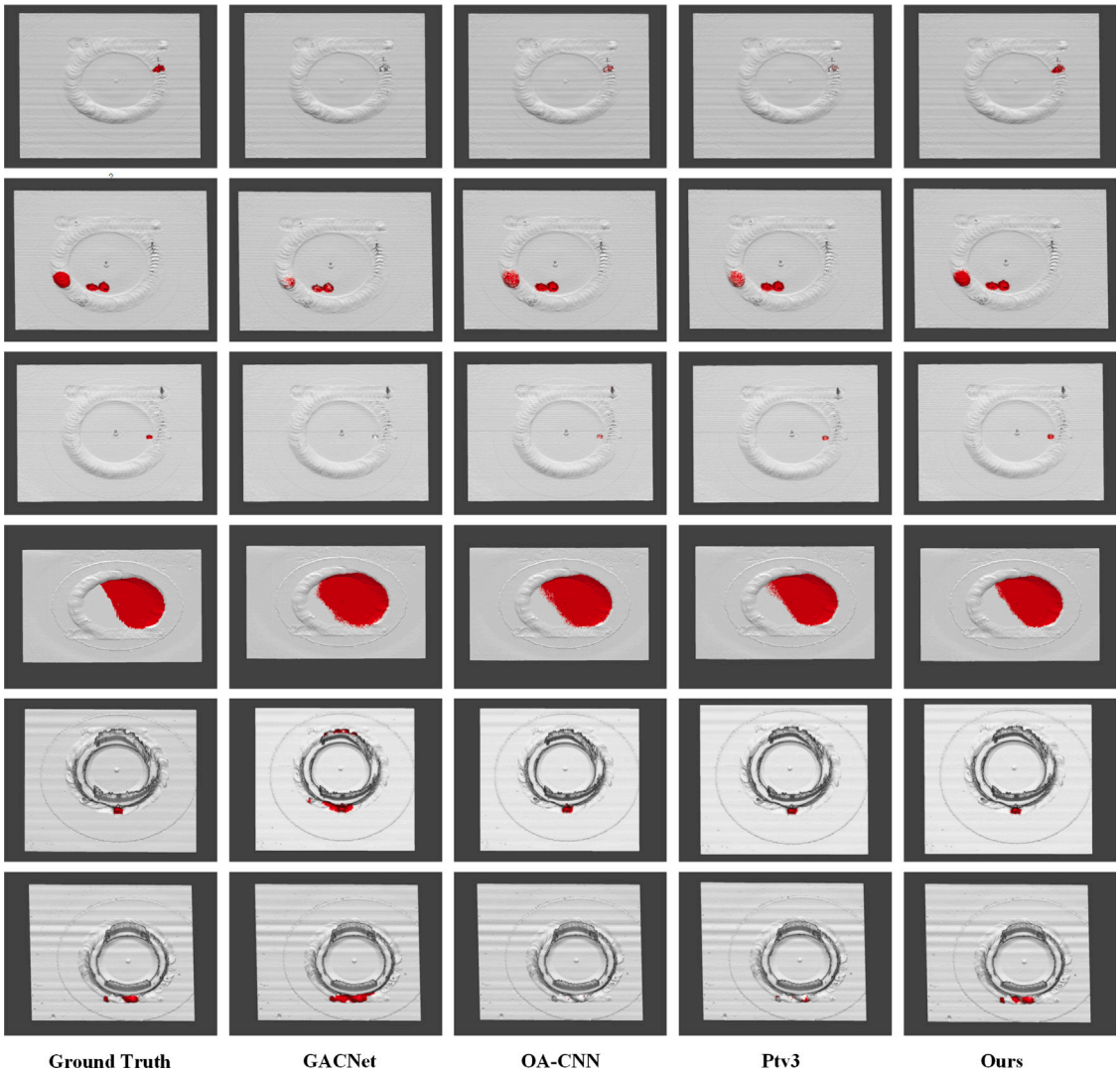

Fig. 12. Visualization of the semantic segmentation results in defects of sealing nail.

Table 5
Ablation results on the sealing nail dataset.

ID	LGAF	SAG-Pooling	OA (%)	mIoU (%)
I	✗	✗	99.36	76.67
II	✓	✗	99.39	78.86
III	✗	✓	99.31	77.13
IV	✓	✓	**99.43**	**79.26**

- **Experiment III:** Uses GCN blocks with SAG-Pooling.
- **Experiment IV (Full LGASS):** Integrates both LGAF and SAG-Pooling.

From Table 5, we observe:

1. **LGAF improves local feature extraction:** Replacing GCNs with LGAF (Experiment II vs. I) leads to a 2.19% increase in mIoU, demonstrating that capturing normal differences significantly enhances defect sensitivity.
2. **SAG-Pooling preserves spatial context:** Replacing standard pooling with SAG-Pooling (Experiment III vs. I) yields a 0.46% mIoU gain, confirming that spatially aware pooling retains critical geometric information.
3. **Combined effect:** Integrating both LGAF and SAG-Pooling (Experiment IV) achieves a 0.40% additional mIoU improvement over Experiment II and a 2.59% gain over Experiment I, validating the complementary benefits of both modules.

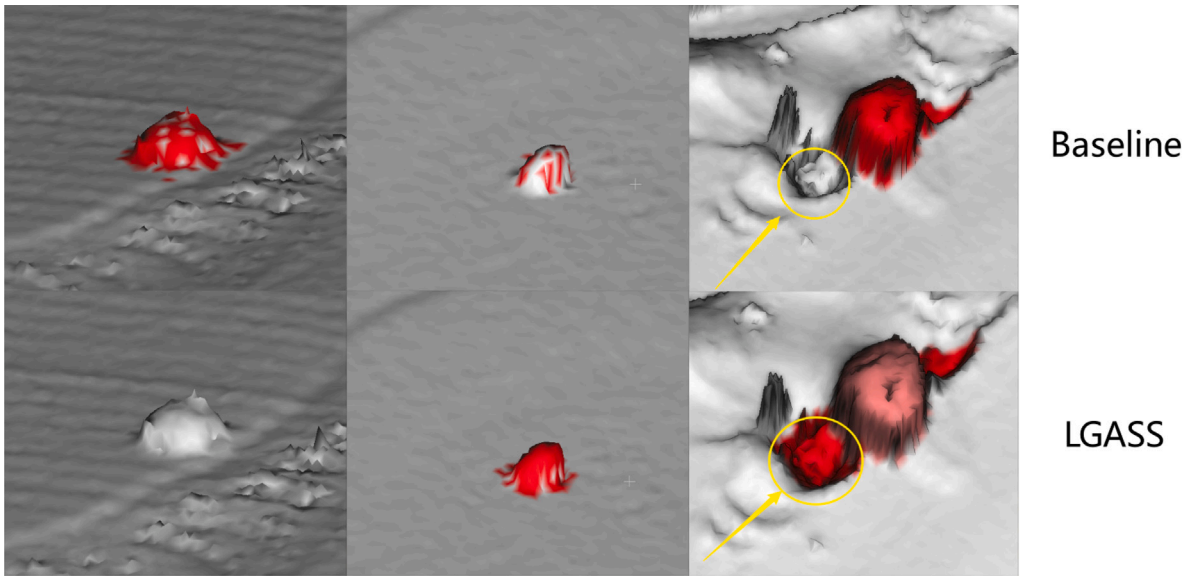

Fig. 13. Comparative visualization of defect segmentation performance between baseline and LGASS.

Table 6
Performance comparison across different modules of LGASS.

Method	Module	Inference time (ms)	Total time (ms)
LGASS	LGAF	106.57	220.9
	SAG-Pooling	32.91	
	GCN	70.44	
	UpSample	10.50	
	Classifier	0.48	

Table 7
Ablation on different input types (sealing nail dataset).

Input type	OA	mAcc	mIoU
Coordinates only	99.42	90.92	76.94
Coordinates + Normals	99.51	91.43	78.31
Normals only	99.47	92.37	79.23

Table 8
Impact of weighted loss functions on performance for different defect categories.

Defect class	Baseline (%)	Weighted loss (%)	ΔIoU (%)
Burst	75.17	75.82	+0.65
Pit	73.22	72.76	−0.46
Moltenbead	68.32	70.45	+2.13
Warpage	93.07	91.84	−1.23
Pinhole	62.93	65.23	+2.3
mIoU	78.73	79.26	+0.53

We additionally provide two sets of visualized segmentation results comparing the Baseline and LGASS models. Fig. 13 demonstrates the baseline model fails to capture fine-grained defects. In contrast, our method achieves significantly more precise segmentation thanks to LGAF's powerful local geometric feature extraction and SAGPooling's feature aggregation capabilities. Fig. 14 demonstrates the OA, mAcc, mIoU, and Loss curves of both models during testing, proving that LGASS outperforms the baseline model with significantly better evaluation metrics. LGASS achieved the highest mIoU between 30 and 40 epochs, indicating that the proposed LGASS model can more effectively detect sealing nail defects. On average, LGASS processes one 131072-point sample in 220.9 ms on an RTX 4070 GPU, demonstrating efficiency suitable for real-time industrial inspection.

To address practical deployment considerations, we measured the peak GPU memory usage during inference, which reached approximately 2.8 GB on the RTX 4070 GPU (12 GB VRAM). This memory footprint indicates that our model can efficiently run on commonly deployed industrial hardware with moderate specifications, such as GPUs with 4–6 GB of VRAM, after applying standard optimization techniques like quantization or dynamic batch sizing.

For real-time performance on lower-specification industrial hardware (e.g., NVIDIA GTX 1060 with 6 GB VRAM or Jetson AGX with 4 GB VRAM), we conducted additional analysis using model compression techniques. With FP16 quantization, the memory requirement reduces to approximately 1.9 GB. This makes the model compatible with most industrial inspection systems that typically feature mid-range GPUs. The inference time on such devices is projected to be under 400 ms per sample, which meets the real-time requirement of 2–3 Hz for typical industrial conveyor belt systems (see Table 6).

We also investigate different input types. As shown in Table 7, the ablation studies reveal that using only normal vectors as input leads to better performance compared to using coordinates alone or in combination with normals. This is because normal vectors capture changes in surface orientation, which are highly discriminative for detecting defects in sealing nails. Defects such as pinholes or bursts manifest as abrupt changes in surface normals, whereas positional data alone may not sufficiently highlight these anomalies. Therefore, prioritizing normal information allows the model to focus on the most relevant features for defect detection.

Our discussion on the impact of loss function weighting demonstrates its significant effect on prediction accuracy for minority defect classes. As evidenced in Table 8, the weighted loss substantially enhances performance for minority classes (pinholes, Moltenbeads), yielding IoU gains exceeding 2%, while not adversely affecting detection performance for other defect types. This improvement occurs because the loss weight w directly amplifies gradients of minority classes during backpropagation, compelling the model to prioritize learning their features—aligning with our objective to increase sensitivity toward minority defect categories. In contrast, conventional loss functions neglect these classes due to their lower contribution to the total loss.

5. Conclusion

In this paper, we proposed LGASS, an efficient semantic segmentation network designed for industrial defect detection in automotive battery sealing nails. LGASS's core innovations are:

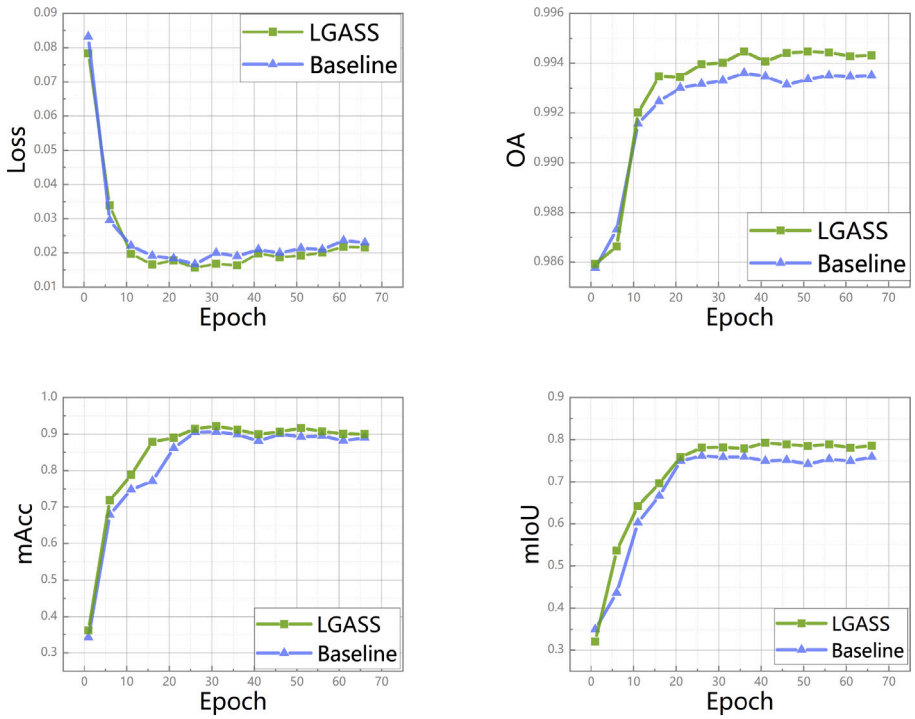

Fig. 14. Comparison of Loss, mAcc, OA, and mIoU between Baseline and LGASS during testing.

- **Local Graph Attention Filter (LGAF):** A novel attention-based operator that leverages normal vector differences to extract fine-grained local geometric features from point clouds, enhancing sensitivity to subtle surface defects.
- **Spatial Attention Graph Pooling (SAG-Pooling):** A pooling mechanism that assigns spatially adaptive weights to points during downsampling, preserving important defect cues and mitigating information loss.
- **Single-Stage End-to-End Pipeline:** An integrated network that directly processes raw 3D point clouds from structured-light scanners, enabling simultaneous defect localization and geometric quantification in one pass.

We constructed and released a specialized point cloud dataset of automotive battery sealing nails, which includes various defect types and exhibits realistic class imbalance. Extensive experiments demonstrate that LGASS outperforms existing methods—such as GACNet, PointNet++, and Point Transformer—achieving state-of-the-art metrics on the sealing nail dataset (99.47% OA, 92.37% mAcc, 79.23% mIoU). Ablation studies confirm the individual contributions of LGAF and SAG-Pooling.

Nonetheless, LGASS may face challenges with highly concave geometries, such as recessed nails with overlapping defects, where fixed k-NN graphs limit long-range dependency modeling. Additionally, minority defect classes like pinholes exhibit relatively lower IoU values due to severe class imbalance in the dataset. Such defects occur infrequently and exhibit subtle geometric variations, resulting in insufficient training instances for the model to learn discriminative features effectively. This imbalance can lead to biased learning toward majority classes, limiting generalization on under-represented defects.

To address this, practical strategies such as synthetic data augmentation via techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be employed to generate realistic minority-class samples by interpolating features between existing instances. Adaptive sampling during training, such as re-weighting loss functions to penalize minority-class misclassifications more heavily, can also enhance focus on challenging examples. Furthermore, incorporating dynamic sampling strategies that prioritize hard negatives or rare defects during batch selection could improve model sensitivity.

Future work will address these limitations via adaptive graph construction and imbalance-aware learning. Additionally, LGASS's geometry-driven approach has potential beyond sealing nails, enabling precise defect detection in automotive welds or metallic surfaces, thus enhancing manufacturing quality control. We will also integrate advanced augmentation pipelines and class-balanced sampling to boost minority-class performance in real-world deployments.

Furthermore, we acknowledge that the current study is primarily validated on our specialized dataset of automotive battery sealing nails. While this dataset captures realistic industrial scenarios and class imbalances, its scale is relatively small compared to large-scale point cloud benchmarks. This limitation may constrain the model's generalization performance across an even wider range of defect types or vastly different industrial environments. We anticipate that with the availability of larger and more diverse datasets in the future, the generalization ability of LGASS could be substantially improved through large-scale pre-training or transfer learning techniques. Such advancements would further unlock the potential of our geometry-driven approach for precise defect detection in automotive welds or metallic surfaces, thus enhancing manufacturing quality control.

CRediT authorship contribution statement

Wei Pan: Writing – review & editing, Validation, Supervision, Methodology. **Yuhao Wu:** Writing – original draft, Methodology, Investigation, Data curation. **Wenming Tang:** Writing – review & editing. **Qinghua Lu:** Validation, Project administration, Conceptualization. **Yunzhi Zhang:** Writing – review & editing, Supervision, Methodology.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Wei Pan reports financial support and article publishing

charges were provided by OPT Machine Vision. WEI PAN has patent #CN119048449A pending to Pending. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is conducted and supported by OPT Machine Vision with an industry fund Dongguan Key Research &Development Program, China (No. 20241200300122).

Data availability

I have shared the link to my data/code in the paper.

References

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3D semantic parsing of large-scale indoor spaces. pp. 1534–1543.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5828–5839.

Guo, M.H., Cai, J.X., Liu, Z.N., Wu, Y.J., Huang, J.B., 2021. PCT: Point cloud transformer. Comput. Vis. Media 7 (2), 187–199.

Konovalenko, I., Maruschak, P., Kozbur, H., Brezinová, J., Brezina, J., Nazarevich, B., Shkira, Y., 2022. Influence of uneven lighting on quantitative indicators of surface defects. Machines 10 (3), 194.

Lai, X., Liu, J., Jiang, L., Lu, J., Guo, Y., Zhang, H., 2022. Stratified transformer for 3D point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8500–8509.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: Convolution on $\mathcal{X}$-transformed points. Adv. Neural Inf. Process. Syst. 31, 820–830.

Li, H., Xie, W., Pan, W., 2025. Photometric stereo-based defect detection for lithium battery top covers. IPMV 2025, In: Seventh International Conference on Image Processing and Machine Vision, vol. 13636, SPIE, pp. 100–107.

Lu, L., Bu, C., Su, Z., Guan, B., Yu, Q., Pan, W., Zhang, Q., 2024. Generative deep-learning-embedded asynchronous structured light for three-dimensional imaging. Adv. Photonics 6 (4), 046004–046004.

Pan, W., Jiang, B.f., Tang, W.m., Wu, F.p., Li, S.p., 2024. Gap measurement method based on projection lines and convex analysis of 3D points cloud. Meas. Sci. Technol. 35 (10), 105024.

Peng, B., Wu, X., Jiang, L., Chen, Y., Zhao, H., Tian, Z., Jia, J., 2024. Oa-cnns: Omni-adaptive sparse cnns for 3d semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21305–21315.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process. Syst. 30, 5105–5114.

Qian, G., Li, Y., Peng, H., Lai, J., Wang, L., 2022. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. Adv. Neural Inf. Process. Syst. 35, 23192–23204.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer International Publishing, pp. 234–241.

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Duan, H., 2020. PV-RCNN: Point-voxel feature set abstraction for 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538.

Te, G., Hu, W., Zheng, A., Hu, Y., Wang, M., 2018. RGCNN: Regularized graph CNN for point cloud segmentation. In: Proceedings of the 26th ACM International Conference on Multimedia. pp. 746–754.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

Wang, L., Huang, Y., Hou, Y., Liu, X., 2019a. Graph attention convolution for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10296–10305.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019b. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. 38 (5), 146:1–146:12.

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H., 2024. Point transformer v3: Simpler faster stronger. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4840–4851.

Xu, M., Ding, R., Zhao, H., Shi, S., Wang, W., 2021. PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3173–3182.

Xu, C., Ye, Y., Zhang, J., Li, D., Chen, Q., 2023. A few-shot learning method for the defect inspection of lithium battery sealing nails. In: Proceedings of the 2023 4th International Conference on Computing, Networks and Internet of Things. pp. 983–989.

Yan, Y., Mao, Y., Li, B., 2018. SECOND: Sparsely embedded convolutional detection. Sensors 18 (10), 3337.

Yang, Y.Q., Guo, Y.X., Xiong, J.Y., Li, H., Zhao, M., Wang, S., 2023. Swin3D: A pretrained transformer backbone for 3D indoor scene understanding. arXiv preprint arXiv:2304.06906.

Yu, Q., Nie, Y., Peng, S., Miao, Y., Zhai, C., Zhang, R., Han, J., Zhao, S., Pecht, M., 2023. Evaluation of the safety standards system of power batteries for electric vehicles in China. Appl. Energy 349, 121674.

Zhang, N., Pan, Z., Li, T.H., Liu, Q., Wang, S., 2023. Improving graph representation for point cloud segmentation via attentive filtering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1244–1254.

Zhao, H., Jiang, L., Jia, J., Koltun, V., Li, H., 2021. Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268.

Zhou, H., Feng, Y., Fang, M., Han, X., Wang, X., 2021. Adaptive graph convolution for point cloud analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4965–4974.

Zhou, Y., Tuzel, O., 2018. VoxelNet: End-to-end learning for point cloud based 3D object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499.

Zou, X., Pan, B., 2021. Full-automatic seed point selection and initialization for digital image correlation robust to large rotation and deformation. Opt. Lasers Eng. 138, 106432.