ISPRS Journal of Photogrammetry and Remote Sensing 221 (2025) 20–43

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

GEPT-Net: An efficient geometry enhanced point transformer for shield tunnel leakage segmentation

Jundi Jiang [a,1], Yueqian Shen [a,1,*], Jinhu Wang [b], Jinguo Wang [a], Chenyang Zhang [a], Jingyi Wang [a], Vagner Ferreira [a]

[a] School of Earth Sciences and Engineering, Hohai University, Nanjing 211100, China
[b] Institute for Biodiversity and Ecosystem Dynamics (IBED), University of Amsterdam, Amsterdam 1012 WX, the Netherlands

ARTICLE INFO

Keywords:
Leakage segmentation
Shield tunnel
Geometry enhanced point transformer
Point cloud
Serialization

ABSTRACT

Subway shield tunnels have emerged as the preferred solution for urban transportation due to their convenience and safety. Constructed using prefabricated concrete segments, these tunnels exhibit structural stability. However, the segment joints and bolt holes are prone to groundwater infiltration under prolonged external stress, potentially compromising the lifespan of the shield tunnels. Consequently, effective detection methods are imperative to ensure the safe operation of these tunnels. Accurate data acquisition and precise extraction of leakage features are critical for detecting leakages in subway tunnels. This research introduces Efficient Geometry Enhanced Point Transformer Network (GEPT-Net), an innovative point cloud semantic segmentation network designed specifically for detecting tunnel leakage. GEPT-Net leverages the observation that leakages predominantly occur at segment joints and bolt holes, characterized by distinct geometric features and lower intensity. The network incorporates Fast Point Feature Histograms (FPFH) to effectively capture these geometric features from the input data. Additionally, we introduce a point cloud serialization technique utilizing space-filling curves, enabling the network to perceive a greater number of points within the same computational power, thereby balancing efficiency and accuracy. The Geometry Enhanced Channel Attention (GECA) Block is introduced to enhance the interaction between FPFH feature channels and intensity channels, enhancing the precise localization of leakage areas. Furthermore, the Lovasz Hinge Loss is employed to address the issue of extreme class imbalance. We constructed a tunnel leakage point cloud dataset, named S3DIS_leakage, comprising approximately 1,600 m between two stations, to train and evaluate the performance of our network. Experimental results demonstrate that GEPT-Net achieves superior performance in tunnel leakage semantic segmentation, attaining approximately 85 % mean Intersection over Union and 89 % accuracy for leakage classes, outperforming cutting-edge 2D and 3D networks by at least 12 %. Moreover, GEPT-Net maintains a remarkable balance between segmentation accuracy and computational efficiency, rendering it viable for practical engineering applications. This study not only establishes a robust approach for tunnel leakage detection but also paves the way for future research on the comprehensive segmentation of shield tunnel components. The proposed framework is available from the following github repository: https://github.com/jdjiang312/GEPT-Net.

1. Introduction

In recent years, the rapid development of tunnel infrastructure has made tunnels a critical component of global public transportation systems. Shield tunnels, built from prefabricated concrete segments, are particularly prone to leakage. Influenced by groundwater and external stressors, these leakages pose serious risks to both the lifespan and structural integrity of the tunnels (Stałowska et al., 2022). Therefore, the efficient and precise detection of such leakages is crucial for the effective maintenance of these essential structures.

Traditional methods of tunnel leakage detection have been conducted through manual inspections, where inspectors are tasked with

* Corresponding author.
E-mail addresses: jdjiang@hhu.edu.cn (J. Jiang), y.shen_lidar@hhu.edu.cn (Y. Shen), jinhu.wang@hotmail.com (J. Wang), wang_jinguo@hhu.edu.cn (J. Wang), zhangcy@hhu.edu.cn (C. Zhang), wang_jy@hhu.edu.cn (J. Wang), vagnergf@hhu.edu.cn (V. Ferreira).
[1] Jundi Jiang and Yueqian Shen are the co-first authors.

https://doi.org/10.1016/j.isprsjprs.2025.01.032
Received 13 July 2024; Received in revised form 26 January 2025; Accepted 27 January 2025
Available online 3 February 2025

identifying the specific ring numbers of leakages (Huang et al., 2020). However, this approach is beset with several inefficiencies caused by limited inspection windows, high labor costs, and a considerable margin for human error. Alternatively, image-based methods have been employed to address these challenges, leveraging visual data to detect leakages. Nevertheless, these methods are not without their drawbacks, including reliance on stable lighting conditions and susceptibility to image quality variations. The introduction of LiDAR technology has revolutionized this conventional process. With its exceptional precision, efficiency, and ability to collect large-scale data, LiDAR has proven to be a vastly superior alternative for tunnel monitoring. LiDAR technology operates by emitting laser pulses towards a target and capturing the reflected echoes to generate three-dimensional (3D) point cloud data. Each point in this dataset includes a reflection intensity value, which is critical for identifying leakage areas since water tends to absorb laser pulses, thereby reducing the reflection intensity at those points. Additionally, each point contains precise 3D coordinate information, facilitating the accurate localization of leakages.

Mobile Laser Scanning (MLS) systems have particularly gained attention in monitoring subway tunnels, offering significant advancements in various areas including crack detection (Stałowska et al., 2022; Feng et al., 2023a), convergence deformation detection (Cui et al., 2019; Yang and Xu, 2021), misalignment detection (Du et al., 2022), and leakage detection (Ji et al., 2022; Zhang et al., 2022; Liu et al., 2022a). These applications demonstrate the technology's pivotal role in enhancing the accuracy and reliability of tunnel assessments, ultimately facilitating a more systematic approach to infrastructure maintenance. However, the immense volume of 3D point cloud data collected by MLS systems often amounts to hundreds of millions of points per kilometer of tunnel, which makes leakage detection a complex task. To efficiently process this data, researchers typically project point cloud data along the tunnel vault in a planar form, creating a reflection intensity map based on intensity properties (Liu et al., 2022a). This map transforms the volumetric data into a format that is more manageable and amenable to analysis with computer vision techniques. However, this approach relies on manual parameter adjustments, which might lead to inconsistencies in detection results. Moreover, the process demands a high level of expertise and considerable time investment, posing further challenges in operational setting. This underscores the need for continued advancements in automated analysis methods to enhance the reliability and efficiency of tunnel leakage detection using MLS data.

Deep learning techniques have achieved remarkable success across various domains, particularly excelling in image-based tasks such as object detection and semantic segmentation (Cheng et al., 2021). These models have substantial potential to revolutionize tunnel leakage detection, offering improved precision and efficiency. However, tunnels present a complex environment due to their segmented construction and the constant stress exerted from multiple directions, making areas such as segment joints, intra-segment joints, and bolt holes prone to leakage. These areas often show clear geometric disparities when compared to the smooth surface of the tunnel lining. Despite the remarkable capabilities of image-based deep learning networks, they exhibit inherent limitations when applied to tunnel leakage detection. 2D image-based networks, in particular, struggle to fully utilize the rich spatial information provided by point cloud data, which is crucial for capturing the nuanced geometric differences essential for identifying leakages. When these networks process point cloud intensity images in two dimensions, they often fail to discern between actual leakages and ancillary structures that may reflect similar intensity due to their material properties, thus leading to inaccurate segmentations. These challenges restrict traditional 2D image-based deep learning networks from effectively handling the unique demands of tunnel environments. As a result, there is an increasing shift towards the development of specialized deep learning architectures that are designed to exploit the 3D spatial data from point clouds. These advanced networks aim to leverage the full spectrum of geometric information available, improving the accuracy and reliability of leakage detection in tunnel infrastructures.

Although point cloud-based deep learning networks offer distinct advantages for tunnel leakage detection, they still face significant challenges: 1) Data Volume and Complexity: MLS systems produce an enormous volume of point cloud data, where each point encapsulates a wealth of spatial and intensity information. Managing this data requires significant computational resources and processing time, which places a considerable burden on the design and scalability of these models. Efficient data handling and processing strategies are critical to making these models viable for real-world applications. 2) Network Design and Computational Limits: Designing deep learning networks capable of handling extensive receptive fields is crucial for extracting rich local features that are essential for accurate leakage detection. However, achieving this is computationally intensive. These networks must strike a delicate balance between computational efficiency and the accuracy of leakage detection, which often restricts their practical application. 3) Lack of Training Data and Class Imbalance: The development of robust deep learning models for tunnel leakage detection is further complicated by the scarcity of publicly accessible tunnel 3D point cloud datasets. Moreover, existing datasets typically exhibit severe class imbalances, with no-leakage points vastly outnumbering leakage points. This imbalance complicates the training process and can potentially skew the performance of models towards the majority class. To effectively address these challenges, we propose GEPT-Net, an efficient and effective deep learning network specifically designed for tunnel leakage detection. To the best of our knowledge, GEPT-Net is the first 3D point cloud-based deep learning network designed specifically for this purpose. The contributions of this work are as follows:

1) Geometric features are incorporated to enhance the accuracy of tunnel leakage. Tunnel leakage typically occurs in areas with distinct geometric features. Therefore, a Geometric Feature Channel Aggregation (GECA) block is specifically designed to enhance the interaction between geometric feature channels and intensity channels. This integration enables the proposed model to more precisely identify areas with lower intensity and prominent geometric features, reducing false positives.

2) A novel point cloud serialization method is introduced to organize scattered discrete points sequentially along space-filling curves. This arrangement allows each patch processed by the network to incorporate a greater number of points, effectively expanding the receptive field. This approach enhances model performance while maintaining computational efficiency.

3) Addressing class imbalance with S3DIS_leakage datasets incorporating Lovasz Hinge Loss. The S3DIS_leakage dataset is created specifically for tunnel leakage detection. However, this dataset is characterized by a significant class imbalance. To counteract this imbalance in the GEPT-Net, we integrated the Lovasz Hinge Loss. This loss function is particularly effective at mitigating the effects of the extreme class imbalance, enhancing the ability of the model to accurately distinguish between leakage and non-leakage areas.

2. Related work

Leakage segmentation can be broadly categorized into two primary approaches based on the type of data being processed: the segmentation of 2D images and the segmentation of 3D point clouds. The following sections will discuss these approaches in detail.

2.1. Leakage segmentation of 2-D images

2.1.1. Optical image-based methods

Traditional shield tunnel leakage detection typically relies on manual inspections, which are subjective, time-consuming, and labor-intensive (Yuan et al., 2013). The inspection windows available in subway tunnels are notably limited, considerably reducing the efficiency of manual inspections. However, the continuous development in

computer vision technology has led to significant efforts to automate defect detection in shield tunnels, enhancing both accuracy and efficiency.

Leakage segmentation of 2-D images can be categorized into two types, with one prevalent method utilizing RGB cameras. This approach generally employs linear array charge-coupled device (CCD) cameras to capture images of tunnel linings, which are then processed using various image techniques to detect leakages. Techniques such as edge detection, adaptive thresholding, and region-growing algorithms are commonly used involving three key steps: contrast enhancement, mathematical morphological processing, and information extraction via linear filters (Xue and Li, 2018). However, these traditional methods rely heavily on manual parameter tuning, leading to inconsistencies in results. They require significant expertise and time and often struggle with variations in lighting and structural conditions within tunnels, sometimes leading to missed or false detections. Traditional methods for tunnel defect detection have encountered numerous limitations, prompting the increasing adoption of deep learning technology in this field over recent years. Deep learning technology has revolutionized tunnel leakage detection, achieving the state-of-the-art result. This advancement is highlighted by the development of a fully convolutional network (FCN) model, specifically tailored for this application (Feng et al., 2023b; Xue and Li, 2018). Furthering this progress, Zhao et al. (2020) presented a method using a Mask Region-based Convolutional Neural Network (Mask R-CNN) for segmenting moisture marks on shield tunnel linings. Building on these developments, Xue et al. (2021) further enhanced the performance of Mask R-CNN by introducing strategies to address three distinct error types and implementing four optimization measures. Despite these improvements, challenges, as pixel segmentation results often include excessive noise, which results in insufficient engineering requirements. Accordingly, Xue et al. (2021) implemented transfer learning and a cascading strategy to enhance detection accuracy. Nevertheless, methods like CNNs (Xue et al., 2022; Protopapadakis and Doulamis, 2015) are constrained by their local receptive fields, which restrict their ability to capture long-range dependencies and global features critical for accurate leakage detection. To overcome this limitation, Geng et al. (2023) proposed the atrous channel pyramid attention network (ACPA-Net) specifically for rail tunnel lining leakage segmentation. This network strengthens the representation ability of ACPA-Net by explicitly modeling the dependencies between feature channels. In a further effort to refine the universality and segmentation performance for identifying leakage areas in tunnels, Wu et al. (2023) have developed an advanced approach that incorporates an improved Generative Adversarial Network and Swin Transformer model. This combination leverages the strengths of both technologies to enhance detection accuracy and adaptability across varying tunnel environments.

However, these methods for tunnel defect detection are notably sensitive to lighting conditions. In real-world scenarios, numerous interferences within subway tunnels, sharing similar colors and shapes with leakages, complicate the detection process. This often leads to insufficient detection accuracy for practical applications, as these methods struggle to distinguish actual leakages from false positives effectively.

2.1.2. Lidar-based methods

An alternative method based on point cloud intensity maps offers a significant advantage by not being constrained by lighting conditions. LiDAR scans have become a preferred technology, rapidly replacing traditional optical imagery for tunnel defect detection. This method is expected to accurately detect and locate leakage areas in tunnels. LiDAR scanning systems typically emit near-infrared wavelengths and receive the reflected echoes (Tan et al., 2016). The emitted light is easily absorbed by water, making the point cloud intensity values in leakage areas significantly lower than in non-leakage areas. This characteristic is highly effective for detecting leakages in complex tunnel environments. Transforming 3D point clouds into a 2D intensity image is a critical step before detection. This process typically involves projecting the point cloud onto a standard cylindrical or ellipsoidal surface and then unfolding it along the arch of the tunnel to form a 2D plane. The result is a point cloud intensity map based on the intensity information of each point. During this process, it is essential to correct the intensity values of point clouds to mitigate the effects of varying surface reflectivity and material properties. Enhancing the contrast between leakage points and the surrounding tunnel surface is crucial for accurate detection. Xu et al. (2018) and Wu et al. (2023) addressed the effects of distance and incident angle on point cloud intensity values by modeling and correcting the intensity information before using the corrected data to detect water leakage regions in tunnels. However, this method relies on threshold settings, making it challenging to handle complex environments and diverse leakage situations. Additionally, the correction process may introduce errors, affecting the accuracy of detection results. To overcome these limitations, deep learning methods are now employed for leakage detection. Deep learning approaches can automatically learn features from the data, making them more robust to variations and capable of achieving higher accuracy and reliability in identifying tunnel leakages. An improved FCN based on VGG-19 was constructed to achieve accurate tunnel leakage detection using intensity images (Cheng et al., 2021). Liu et al. (2022a) proposed a new residual network module (Res2Net) integrated with a cascade structure to form a unified model that extracts the multiscale features of water leakages. This model fully considers the characteristics of water leakages and grades the residual connections within a single residual block. Although these network structures have significantly improved detection accuracy compared to previous methods, the features they learned are limited and still have potential for further exploration in terms of accuracy. Zhang et al. (2023) proposed a novel data augmentation method based on causal inference and established a Dempster–Shafer (D-S) evidence-based feature fusion model to combine local features with global features from different CNN models, thereby enhancing detection accuracy. An attention mechanism is also employed to enhance feature learning. RDES-Net aims to capture the spatial features of leakage data and facilitate meaningful interactions between different channels (Guo et al., 2024). However, none of the aforementioned methods have addressed the issue of sample imbalance, which is very common in tunnel leakage datasets and affects the results. In response to this issue, Zhang et al. (2022) constructed a pixel-weight cross-entropy loss combined with a dual attention module to address the class imbalance issues and improve segmentation performance.

This category of methods aims to avoid the exponential growth in the computational load that occurs with increasing dimensions of 3D point cloud data. It also seeks to prevent the need for more complex network structures due to the inherent characteristics of point cloud data, such as disorder, density inconsistency, non-structural randomness, and sparsity. To address these challenges, point cloud intensity images are utilized for leakage segmentation based on 2-D deep learning networks. However, leakage segmentation based on 2D point cloud intensity images offers limited learning capabilities in complex tunnel environments. Firstly, complex auxiliary facilities, due to their material properties, can produce point cloud intensity information from LiDAR signal reflections that are similar to leakage, potentially resulting in incorrect segmentation outcomes. Secondly, shield tunnels, assembled from prefabricated concrete segments, are subject to longitudinal settlement, convergence deformation, and joint dislocation over time due to external stresses (Xie et al., 2021; Lin et al., 2022). This structural characteristic allows groundwater to permeate the concrete surface through gaps between rings, within rings, and bolt holes. Consequently, leakage typically occurs at the ring gaps and bolt holes of shield tunnels, where geometric features are distinctly different from other concrete surfaces of the tunnel. Thus, methods based on 2D images have not fully leveraged the advantages of 3D point cloud data in capturing these geometric features to identify leakage.

2.2. Leakage segmentation of 3-D point clouds

2.2.1. Traditional methods

Compared to on-site manual inspections and photogrammetry-based leakage detection methods, LiDAR technology has become a popular choice for tunnel defect detection tasks. Numerous studies have highlighted the effectiveness of LiDAR in this field. Wang et al. (2022b) integrated the 3D Otsu method and K-nearest neighbors for leakage detection from point clouds. The proposed approach can achieve detection accuracy comparable to 2-D methods. Building on this, another effort has been made to detect leakage and classify various objects in tunnels using 3D point clouds. This effort developed an auto-optimized lazy learning approach named the Bayesian optimized K-nearest neighbor method, which addresses the issue of sample imbalance (Wang et al., 2022a). However, this method, designed for multi-class tasks, faces less extreme sample imbalance issues and may struggle to tackle the more extreme imbalance found in binary tasks specifically targeting leakage detection.

2.2.2. Deep learning methods

Owing to the ongoing development of deep learning technologies, point cloud deep learning techniques have become highly mature. In recent years, many scholars have developed effective networks to handle the complexities of point cloud data. Due to the irregular and unordered nature of point cloud data, performing convolution operations directly on point clouds is infeasible. However, attention mechanisms have shown significant advantages. For example, Point Transformer (Guo et al., 2021; Zhao et al., 2021; Wu et al., 2022; Wu et al., 2024), Sparse U-Net (Liu et al., 2022b), and Minkowski U-Net (Choy et al., 2019) have demonstrated impressive performance in capturing both local and global features efficiently. These advancements have paved the way for more accurate and robust solutions in various applications, including tunnel leakage detection. Ji et al. (2022) presented an encoder-decoder deep learning method for multi-class object segmentation, including leakage. This method utilized normalized point clouds and constructed voxels as the input for the proposed encoder-decoder architecture, effectively learning the characteristics of point clouds. Another novel approach focused on segmenting classical elements of tunnels, including leakage, aiming to enable the network to learn robust global features and complex local distribution patterns. In this work, the global and local feature encoding block and global correlation modeling, combined with the attention mechanism, were proposed to learn the local and global representations, allowing the model to build a global awareness of each point (Li et al., 2023). However, the aforementioned methods are designed to detect multiple categories within tunnels, which require a significant amount of time and labor for manual labeling. To the best of our knowledge, there is currently no point cloud deep learning network specifically designed for tunnel leakage detection.

In conclusion, 3-D point cloud-based methods for leakage detection in tunnels still face several substantial challenges: (1) Data Volume. The LiDAR system generates an immense of data, with hundreds of millions of points recorded per kilometer of tunnel scanned. This demands significant computational resources and considerable time overhead; (2) Computational limitations. It is challenging to design point cloud deep learning networks that can capture a large receptive field to capture detailed local features. These networks often struggle to balance computational efficiency with high detection accuracy. Despite their efforts, optimizing this balance remains an elusive goal; (3) Sample imbalance. Point cloud datasets specifically for tunnel leakage suffer from severe sample imbalance, a common issue across point cloud applications. This imbalance tends to skew models towards dominant background features rather than focusing on the crucial classes for accurate detection, resulting in compromised accuracy. These challenges restrict the full utilization of point cloud deep learning networks, which prevents them from capitalizing on the inherent benefits of point cloud data, especially when compared to photogrammetry-based leakage detection methods. In response, we will propose innovative solutions aimed at overcoming these challenges and improving the application of 3D point clouds in tunnel leakage detection.

2.3. Summary

Based on the analysis presented, it is evident that traditional leakage detection methods fail to satisfy the accuracy requirements for practical applications and suffer from various shortcomings, such as reliance on parameter settings. In contrast, deep learning methods can effectively circumvent these limitations, presenting the potential for high-precision tunnel leakage detection. However, networks based on photogrammetry do not fully utilize the spatial features of 3D point cloud data, despite the significant geometric differences between leakage and non-leakage areas within tunnels. On the other hand, point cloud deep learning methods face challenges such as extreme sample imbalance, low efficiency, and limited capabilities in feature extraction. These challenges impeded the development of a dedicated 3D point cloud deep learning network specifically designed for tunnel leakage detection. Although 3D point cloud deep learning technology has advanced significantly, it still fails to fully address the unique challenges associated with tunnel leakage detection, often leading to suboptimal outcomes. This work is dedicated to proposing innovative solutions to these prevalent issues, enhancing the application of point cloud data in detecting tunnel leakages.

3. Methodology

In this work, we propose a network for semantic segmentation of shield tunnel leakage from 3D point cloud data obtained by MLS, the overall workflow is illustrated in Fig. 1.

3.1. Data preprocessing

The internal environment of shield tunnels is extremely complex due to the presence of various internal facilities such as bolt holes, cables, grouting holes, and tracks, as well as the irregularity of the tunnel surface. The process of acquiring three-dimensional point cloud data using LiDAR often generates a large amount of noise and extraneous data unrelated to the primary objects of interest (Ji et al., 2022). Direct analysis of raw point cloud data tends to introduce errors. Therefore, data preprocessing is essential to eliminate noise points and irrelevant auxiliary facilities on the tunnel surface to obtain clean data for deep learning networks.

The first step in data preprocessing involves removing outliers, realized by the Statistical Outliers Removal algorithm (Rusu et al., 2008b). After outlier removal, it is necessary to eliminate points representing unwanted and occluded tunnel auxiliary facilities. These points, due to their similar reflection intensity to leakages, can lead to inaccurate leakage detection. This step typically relies on geometric features (Bao et al., 2024), point cloud intensity thresholds, and methods such as elliptical cylinder fitting. Given that auxiliary facilities typically maintain a certain distance from the tunnel lining surface, we employed ellipse fitting to remove them in this research. However, shield tunnels experience deformation due to the substantial stress they endure (Cui et al., 2019). Directly fitting raw point clouds of shield tunnels to ellipse can inadvertently remove points representing the tunnel surface, leading to imprecise results. Therefore, we initially divided the raw point clouds of shield tunnels into several slices (c_i) along the direction of the tunnel at fixed length intervals l. Subsequently, we applied the Random Sample Consensus (RANSAC) algorithm to each segment of the point cloud to fit ellipses, obtaining the equation of the ellipse, as shown in Equation (1), where a, b represents the semi-major and semi-minor axis of the ellipse respectively. We then identified auxiliary facility points within each c_i by comparing their distances from the center of the ellipse to the designed tunnel radius.

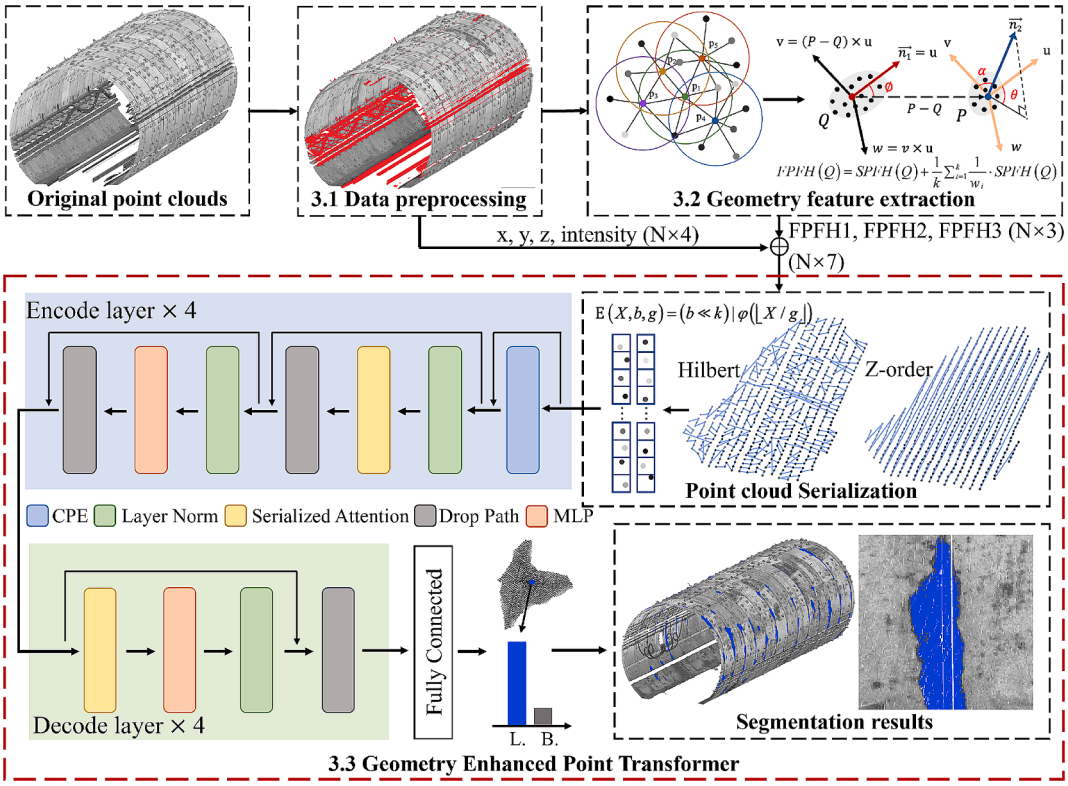


Fig. 1. The overall flowchart of our GEPT Net to segment tunnel leakage.

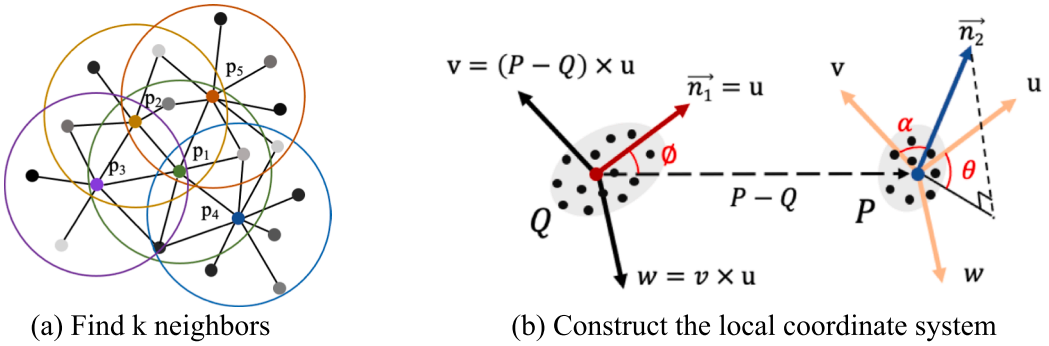
$$(x-x_0)^2/a^2+(y-y_0)^2/b^2=1 \tag{1}$$

3.2. Geometry feature extraction

Geometric feature extraction is a crucial step in point cloud processing and is widely applied in tasks such as point cloud registration and object recognition. The feature descriptors encapsulate the local geometric properties of points in the point cloud, thereby enabling the comparison of the point similarities across different datasets. Feature descriptors such as the Point Feature Histograms (PFH) (Rusu et al., 2008a), Fast Point Feature Histograms (FPFH) (Rusu et al., 2009), and Signature of Histogram of Orientation (SHOT) (Tombari et al., 2010), play a crucial role in determining point similarity by capturing the spatial relationships and surface characteristics around each point.

In this research, we utilize the FPFH algorithm to extract geometric features from tunnel point clouds. The selection of FPFH was driven by its superior computational efficiency and feature representativeness compared to alternative descriptors such as PFH and SHOT. (1) Efficiency: Unlike PFH and SHOT, which generate 125 and 352-dimensional descriptors, respectively, FPFH significantly reduces feature complexity by producing a more compact representation. This reduction in dimensionality accelerates processing time, making FPFH particularly suitable for large-scale point cloud datasets. To illustrate, when applied to a 50-meter tunnel dataset with 7 million points, FPFH completed the processing in 118 s, substantially outperforming PFH and SHOT, which required 1,623 s, and 534 s, respectively. This phenomenon can be further explained by analyzing the theoretical computational complexity of each algorithm. Assuming k nearest neighbors and N points, the theoretical computational complexity of FPFH is $O(Nk)$, which is significantly lower than that $O\left(Nk^2\right)$ of PFH and SHOT. (2) Feature Representativeness: Another key reason for selecting FPFH is its ability to maintain a balance between feature compactness and geometric expressiveness. Given that our network is designed to leverage a few well-selected geometric features, both PFH and SHOT, through detailed in their dimensionality, provide more information than necessary for our network's focus on extracting key features. Selecting a subset from such high-dimensional features could risk losing essential information. In contrast, FPFH, with its lower dimensionality, retains sufficient geometric detail for robust feature selection, making it a more effective choice for representing the local geometry of tunnel surfaces. Therefore, the selection of FPFH as input features for the deep learning network offers an optimal balance between computational efficiency and descriptive power, ensuring robust performance in detection of tunnel leakage. The FPFH begins by collecting neighboring points P within a radius r of the query point Q, establishing connections between each neighbor point and the query point, as illustrated in Fig. 2(a). FPFH links neighbor points to query points, thereby reducing the dimensionality from 125 dimensions to 33 dimensions. Furthermore, it extends the neighborhood range considered for the descriptor, ensuring effective capture of local geometric features. After establishing connections, FPFH estimates the normal vector for each query point within the neighborhood. Subsequently, a local coordinate system is defined for each pair consisting of the query point Q and its neighbor points P, as depicted in Fig. 2(b), where n_1 and n_2 are the normal vectors of Q and P, respectively. Using point P as the origin of the local coordinate system, n_2 is aligned with the U-axis, and $P-Q$ is projected onto n_1 to determine the V-axis. The W-axis direction follows the right-hand coordinate system. Angle deviations α, θ, and ϕ between normal vectors are computed using Equation (2). Each tuple $\langle\alpha,\theta,\phi\rangle$ for each query point and its neighboring points is divided into 11 bins for each feature. Combining these bins across three features results in 33 feature vectors known as the Simplified Point Feature Histogram (SPFH). For each point, its k neighbors are reassessed, and the SPFH of the neighboring points is used to weight the SPFH using Equation (3), where w_i represents the distance between the Q and the P in a given metric space.

$$\alpha=v\cdot\overrightarrow{n_2}\ ,\ \phi=u\cdot\overrightarrow{n_2} \tag{2}$$



(a) Find k neighbors (b) Construct the local coordinate system

Fig. 2. Geometry feature extraction using FPFH.

$$\theta = \arctan(w \cdot n_2, u \cdot n_2)$$

$$FPFH(Q) = SPFH(Q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{w_i} \cdot SPFH(Q) \tag{3}$$

We propose to identify tunnel leakages by analyzing geometric differences between leakage and background regions. To identify optimal geometric features, we performed rigorous feature selection. Specifically, we focused on selecting the top 5 features that exhibit the most significant differences among the 33 FPFH features compared to background and leakage points within each tunnel block. Larger differences indicate more pronounced geometric distinctions between these regions, facilitating effective differentiation using geometric features. Among the identified features, FPFH0, FPFH5, FPFH10, FPFH15, FPFH16, FPFH7, FPFH27, and FPFH28 consistently demonstrated substantial differences and ranked among the top 5 best performing FPFH features. We visualized these top 5 FPFH features for each tunnel block using a concentric ring diagram, as illustrated in Fig. 3. This diagram sequentially represents the top 5 FPFH features that effectively distinguish between background and leakage regions, with FPFH27, FPFH16, and FPFH5 frequently appearing among the selected features.

Furthermore, to identify geometric features that effectively differentiate between background and leakage regions, we calculated the values of the aforementioned eight Fast Point Feature Histograms (FPFH) for both classes within each tunnel block. Fig. 4 illustrates the disparities in data distribution for these features using boxplots. The horizontal axis represents the FPFH features, while the vertical axis shows the values for the leakage and background classes. Each column in the plot displays the value distributions for both classes under a specific FPFH feature.

The statistical distribution of FPFH features reveals that FPFH27, FPFH16, and FPFH5 exhibit distinct distributions between leakage and background conditions. Specifically, the boxplot for FPFH27 shows a concentrated distribution around the median for the leakage class, suggesting consistent and concentrated feature values. The background class also displays relatively concentrated feature values under FPFH27. The minimal overlap in FPFH27 feature values between the two classes highlights its effectiveness in distinguishing them based on geometric features. Similarly, FPFH16 and FPFH5 demonstrate a noticeable difference in data distribution between leakage and background classes, with non-overlapping interquartile ranges indicating substantial differences in most feature values.

The remaining five features, however, exhibit unsuitable characteristics for distinguishing between background and leakage geometric features. Their similar data distributions and numerous outliers indicate unstable patterns. Therefore, to balance computational efficiency and

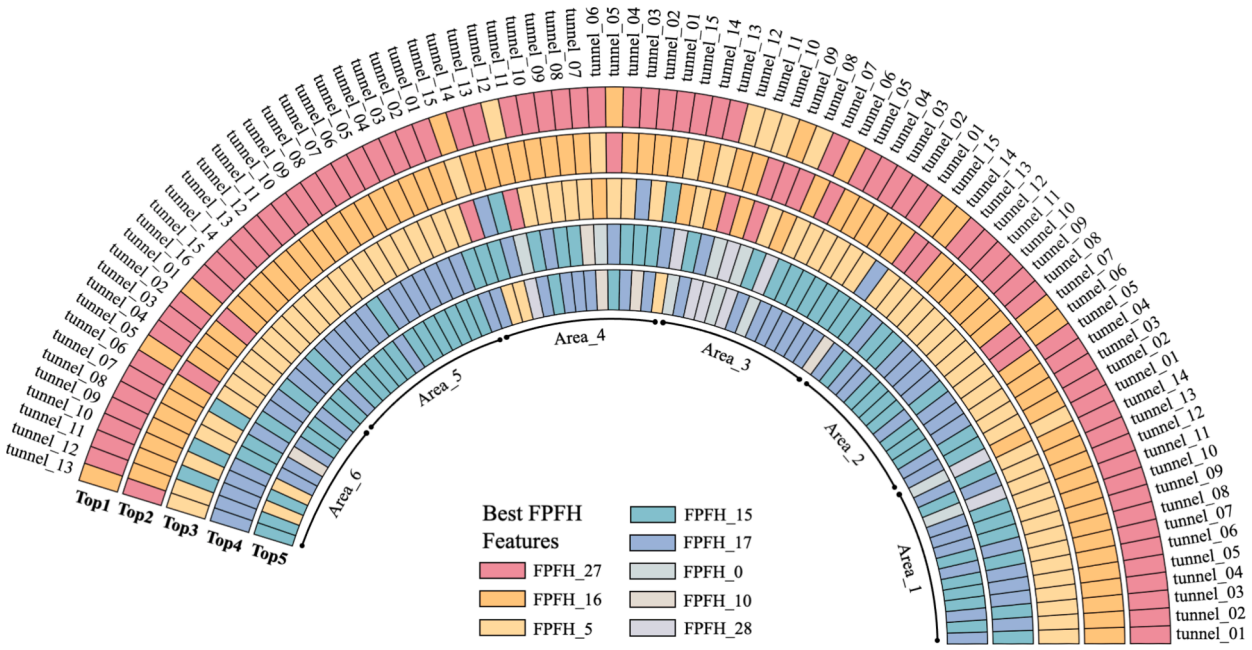


Fig. 3. Concentric Rings Diagram of the Top 5 FPFH Features. In the concentric rings, from the outermost ring to the innermost ring indicates the top5 FPFH features that best distinguish between the background and leakage regions, the best FPFH features for a total of 88 tunnel blocks in 6 Areas of the dataset were calculated in the concentric ring.

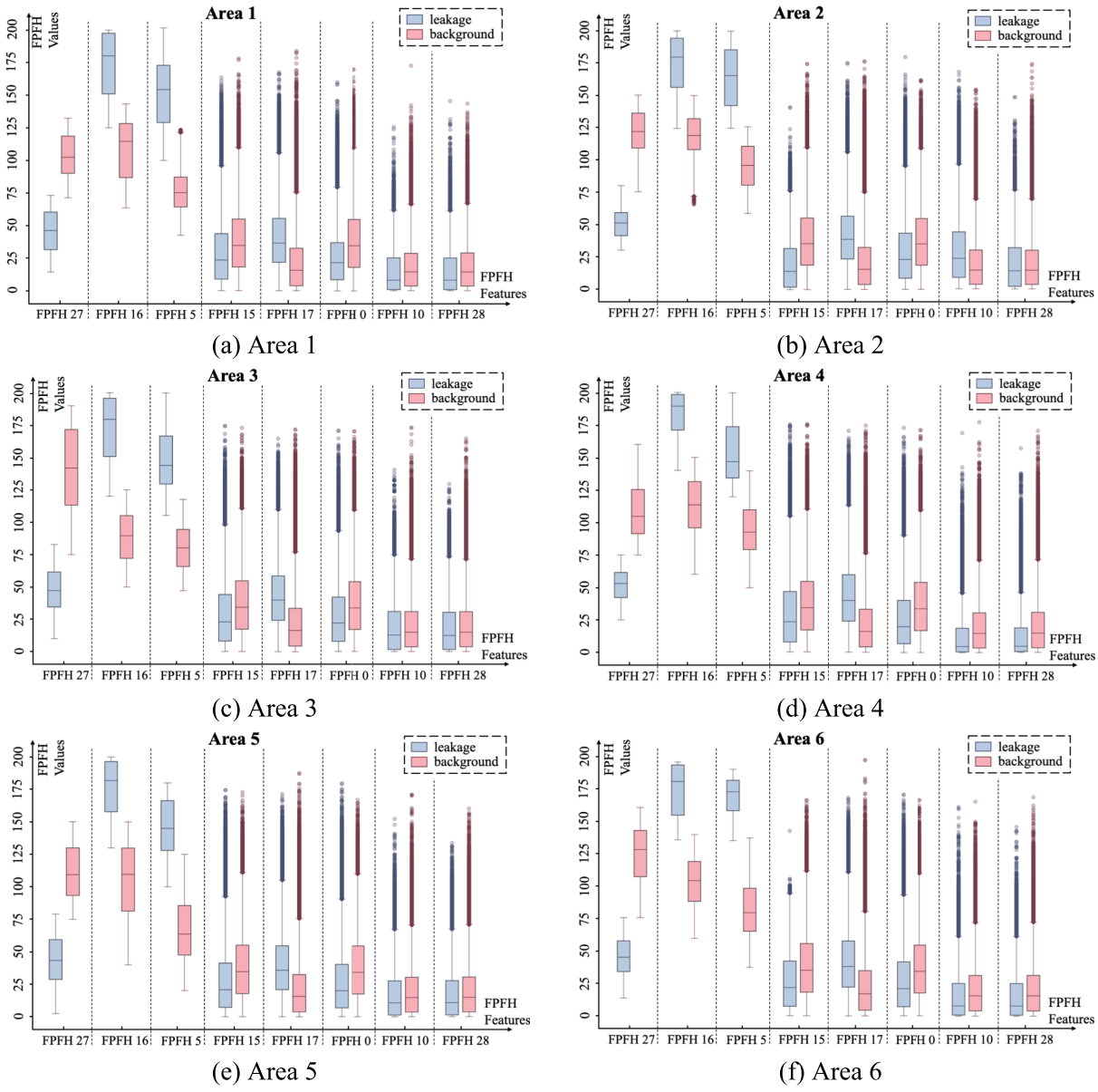


Fig. 4. FPFH feature boxplot for leakage and background classes. In a boxplot, the horizontal line within the box represents the median of the data distribution, indicating the central tendency of the data. The upper and lower boundaries of the box represent the upper quartile (Q3) and lower quartile (Q1), respectively, suggesting the range within which the central 75 % and 25 %of data points lie. The interquartile range is a measure of variability or dispersion within the data, quantified as the distance between Q3 and Q1. The whiskers extending from the box indicate the maximum and minimum data that are not considered outliers. Outliers, depicted as circles, represent data points that are significantly different from the rest of the dataset.

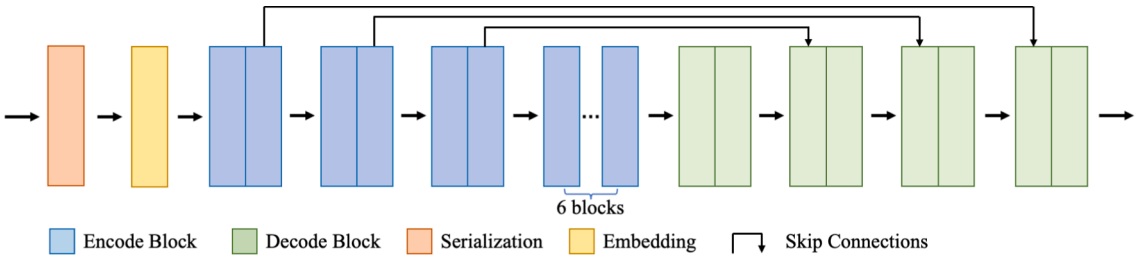
accuracy, FPFH27, FPFH16, and FPFH5 were selected as the primary geometric enhancement features from the top five FPFH features. These features will serve as inputs to the deep learning network.

3.3. Geometry enhanced point transformer

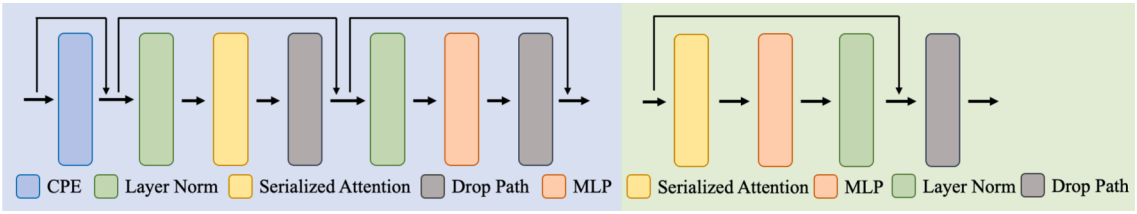
3.3.1. Network architecture

The Point Transformer is a groundbreaking architecture designed specifically for point cloud processing, employing attention mechanisms to efficiently capture both local and global contextual information. Unlike traditional CNN-based approaches, the Point Transformer operates directly on unordered point sets. This enables permutation-invariant processing and effectively models complex spatial dependencies within point clouds. The proposed GEPT-Net, a 3D deep learning network specifically designed for large-scale tunnel leakage detection, adopts Point Transformer as its backbone. GEPT-Net addresses the unique geometric features and point cloud intensity characteristic of leakage while balancing efficiency and accuracy. As illustrated in Fig. 5(a), GEPT-Net employs a U-Net architecture, which consists of *n* encoder and decoder blocks. The encoder utilizes attention mechanisms to selectively focus on informative point interactions and spatial relationships within the input point cloud. It comprises multiple layers of processing units that transform the input point cloud data into high-level feature representations. The decoder reconstructs or generates outputs based on the encoded features, involving the inverse processes of pooling and upsampling. This approach retains hierarchical information and reconstructs detailed output representations, ensuring consistency with the task objectives and effectively interpreting point cloud data.

We represent each point in 3D space with its coordinates (x, y, z), point intensity (I), and geometric features, i.e. FPFH, as the original input to our network. Therefore, an F-dimensional point cloud with n



(a) The overall network architecture of GEPT-Net



(b) Encode block (c) Decode block

Fig. 5. The network architecture of GEPT-Net. (a) Overview of the GEPT-Net architecture, featuring the encoder-decoder framework. (b) The encode block used during the encoding phase. (c) The decode block used during the decoding phase.

points, labeled as either leakage or background, can be denoted as $X = \{x_1, x_2, \cdots, x_n\} \subseteq \mathbb{R}^F$, where $F = 4 + i$, i denotes the number index of FPFH features. To enhance the capability of the network, we serialize the input point cloud, transforming unstructured point clouds into a structured format. This process is crucial for expanding the receptive field, acquiring richer local information, reducing memory usage, and improving the efficiency of network training and inference. Subsequently, an embedding layer converts the serialized point cloud into higher-dimensional feature representations. The network architecture consists of multiple stages in both the encoding and decoding phases. Each stage is composed of blocks that include Conditional Position Encoding, multi-head attention mechanisms, flash attention, and MLP. These blocks work together to extract local and global features from the input point cloud. Each layer updates the feature representation of each point, gradually capturing higher-level geometric and semantic information. In the decoding phase, the network uses a series of upsampling operations. These operations leverage features from the encoder and progressively merge high-level features with low-level detail features to restore the details of the original point cloud. The decoder also aggregates features from the previous decoder stage with skip-connected features from the corresponding encoder stage. This effectively retains feature information at different resolutions, enhancing the quality of point cloud reconstruction and segmentation accuracy.

Below is a detailed description of the implementation and functionality of each module in GEPT-Net.

3.3.2. Point cloud serialization

To harness the simplicity and efficiency inherent in structured data, point cloud serialization has been introduced to convert unstructured point clouds into a structured format. This serialization process involves encoding the 3D points and subsequently sorting the results codes. The sorting procedure reorganizes according to a specified space-filling curve, ensuring that neighboring points in the data structure are proximate in physical space. Here we should demonstrate that point cloud serialization does not involve physically reordering points within the point cloud itself. Instead, it focuses on establishing mappings between points generated during the serialization process. These mappings are achieved through indices or encoding schemes that denote spatial adjacency among points in the point cloud. Mathematically, it can be represented as a mapping function defined in Equation (4):

$$\varphi(X) = (\lfloor X/g \rfloor), \mathbb{Z}^n \Rightarrow \mathbb{Z}^m \tag{4}$$

where $\lfloor * \rfloor$ denotes downward rounding, g represents the grid size for projecting points onto a grid, n is the dimensionality of space, and m represents the dimensionality of the mapped high-dimensional space.

Specifically, the serialization of point clouds involves assigning a 64-bit integer to each point. This process allocates the trailing k bits to encode the position specified by $\varphi(X)$, while the remaining leading bits are assigned to the batch index $b \in \mathbb{Z}$ (Wu et al., 2024). This encoding is defined by Equation (5):

$$\mathbb{E}(X, b, g) = (b \ll k) | \varphi(\lfloor X/g \rfloor) \tag{5}$$

where $\mathbb{E}$ represents the Encoding function, $\ll$ denotes left bit-shift and $|$ is bitwise OR.

Through encoding, spatially adjacent points are grouped closely together, thereby maintaining the spatial locality of the point cloud data. This enhances the ability of the network to grasp the spatial structure and geometric characteristics inherent in point cloud data. Additionally, serialization maps the relationships between point clouds onto a one-dimensional space, enabling networks to treat them akin to sequential data. This simplifies the complexity of data processing, allowing networks to process more points with the same computational resources. Consequently, this expands the receptive field of the network, facilitating the capture of richer local information. We employ two spatial curves for point cloud serialization: the Hilbert curve (Hilbert and Hilbert, 1935) and the Z-order curve (Morton, 1966). These curves transform unstructured and irregular point clouds into ordered sequences while preserving spatial adjacency. The connection patterns of the Z-order curve and the Hilbert curve are illustrated in Fig. 6(a) and (b), respectively.

The Z-order curve organizes points in a "Z" shape, which connects adjacent points along a continuous path. This path may traverse various spatial locations, resulting in adjacent points on the curve covering significant distances. This serialization method proves particularly advantageous for detecting leakages that manifest as elongated shapes within annular gaps and intra-annular gaps. Given that leakage detection primarily focuses on local details within specific regions, these gaps are typically critical areas of concern. The Z-order curve effectively encodes and represents these local details. As described earlier, each patch of the point cloud input into the network has a receptive field capable of considerable distances. As illustrated by the green box in Fig. 7, the tunnel point cloud is typically collected using a cross-sectional scanning method. Consequently, the movement speed disparity between

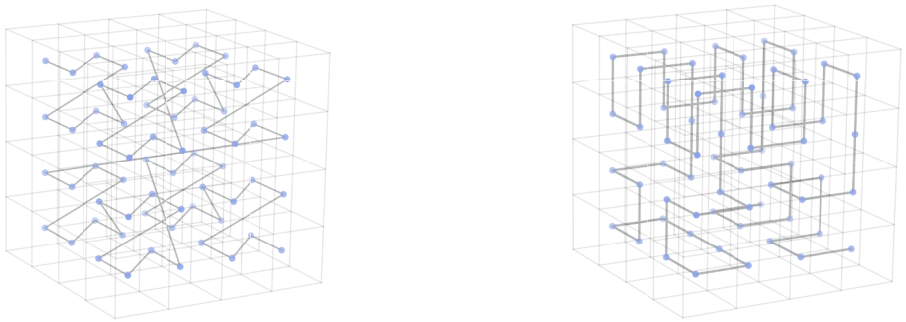


Fig. 6. 3D Space-filling curves. (a) Denotes the way Z-order curves in space filling curves connect point clouds. (b) Shows the way Hilbert curves in space filling curves connect point clouds.

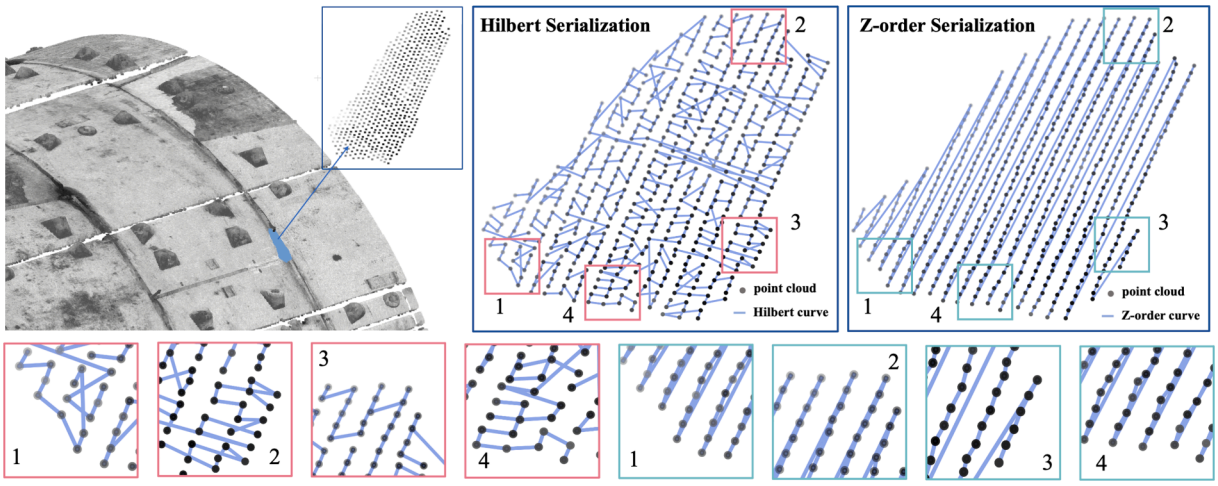


Fig. 7. Hilbert and Z-order Serialization in tunnel leakage. The red and green wireframes represent the cross-up of the Hilbert and Z-order serialization methods for the same area, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the laser scanning platform and the laser signal causes the scanning lines to form a "Z" shape, which corresponds to the connectivity pattern of the Z-order serialization. Therefore, the network can simultaneously consider local details near the leakage location and information from distant areas. This expansive receptive field capable of encompassing significant distances, enhances the accuracy and robustness of deep learning networks in detecting leakage.

The Hilbert curve is a continuous, fractal space-filling curve known for its self-similarity. This property means that the local structure of the curve resembles its overall structure, exhibiting similarity across different scales. This characteristic proves highly advantageous for extracting features from the point cloud. As illustrated by the red box in Fig. 7, the Hilbert curve maintains local connectivity patterns that reflect this self-similarity. This serialization approach allows deep learning networks to utilize the self-similarity of the curve to extract local features effectively. It facilitates robust feature representation and learning across various scales, enhancing the network's ability to discern and analyze details within the point cloud data. In summary, the Hilbert curve's self-similarity enables deep learning models to capture intricate local features crucial for accurate and comprehensive analysis of leakage detection in tunnel environments.

In summary, we begin by serializing the input point cloud at the start of the network, compactly storing spatially adjacent points together. These serialized points are then served as inputs to the subsequent embedding layer. Point cloud serialization greatly expands the receptive field of each patch inputted into the subsequent network, enabling the extraction of richer hierarchical local features. This increased receptive field enables the network to consider both local details and the global environment simultaneously, thereby enhancing detection accuracy. Moreover, this approach allows the network to read and process the data more efficiently, reducing computational overhead.

3.3.3. *Geometry enhanced channel attention block*

Tunnel leakage typically occurs in areas such as gaps between rings, within rings, and bolt holes, where groundwater can easily infiltrate. These locations exhibit distinct FPFH features and lower intensity characteristics. Therefore, it is crucial to enhance both FPFH and intensity features interactively. Without this enhancement, the network might mistakenly identify non-leakage regions with similar FPFH features on the lining surface as leakage areas. To address this issue, we propose to utilize the GECA block to accurately identify leakage regions with distinct FPFH and intensity features. In the embedding layer of the network, a GECA block is introduced at an early stage of feature extraction to enhance FPFH and intensity features. This helps the model learn the crucial relationships between these features earlier in the process. The GECA block adaptively weights features along the channel dimension to enhance useful features and suppress irrelevant ones, thereby improving the representational capacity and generalization performance of the network. The specific implementation is as follows:

First, the input features $X \in \mathbb{R}^{n \times C}$, where C denotes the number of channels (C is set to 7 in GECA block), include coordinates (x, y, z), intensity (I), and geometric features (FPFH27, FPFH26, FPFH5). The input features are processed through global average pooling (GAP) for each channel, generating a vector with the same number of channels. A

fully connected (FC) layer is then used to reduce the dimensionality of the GAP output to a smaller vector, capturing the global information of each channel. This process is illustrated in Equation (6). Following GAP, a channel attention mechanism is introduced to enhance the correlations between intensity and FPFH feature channels. The aggregated feature can be learned through channel attention ω, as shown in Equation (7).

$$y_i = FC(GAP(X)) \in \mathbb{R}^C \tag{6}$$

$$\omega = \sigma(Wy_i) \tag{7}$$

where σ is a Sigmoid function for nonlinear transformation, producing attention weights for each channel, W is a $C \times C$ parameter matrix, as shown in Equation (8).

Given that the FPFH features already contain spatial features, to maintain efficiency and model size, we mask the coordinate channels (x, y, z) with 0. This approach ensures that the interactive enhancement focuses solely on the FPFH features and intensity. Finally, an element-wise product is performed between the enhanced features and the original input. This combined output is then used as the input to the convolutional layer, dynamically adjusting the responses of the intensity and FPFH channels.

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & w^{I,I} & w^{I,FPFH1} & w^{I,FPFH2} & w^{I,FPFH3} \\ 0 & w^{FPFH1,I} & w^{FPFH1,FPFH1} & w^{FPFH1,FPFH2} & w^{FPFH1,FPFH3} \\ 0 & w^{FPFH2,I} & w^{FPFH2,FPFH1} & w^{FPFH2,FPFH2} & w^{FPFH2,FPFH3} \\ 0 & w^{FPFH3,I} & w^{FPFH3,FPFH1} & w^{FPFH3,FPFH2} & w^{FPFH3,FPFH3} \end{bmatrix} \begin{matrix} \text{coord} \\ \text{intesity} \\ \text{FPFH1} \\ \text{FPFH2} \\ \text{FPFH3} \end{matrix} \tag{8}$$

The structure of the Embedding layer is shown in Fig. 8(a). It initially employs the GECA block to learn and enhance the relationship between intensity and FPFH geometric feature channels. At this stage, the feature information is relatively coarse, and incorporating channel attention mechanisms helps the model in selecting and emphasizing crucial features in higher dimensions, thereby improving the feature extraction in subsequent layers. To ensure model efficiency, sparse convolution is utilized post-GECA block to transform input point cloud data features into higher-dimensional representations. Sparse convolution reduces computational and memory overheads by computing only on existing points, thus lowering computational burdens, which is preferred in 3D large-scale training. Additionally, it focuses solely on actual data points within each convolutional kernel's receptive field, enabling better capture of local features in point clouds and helping the network learn more accurate and meaningful leakage feature representations. By operating directly on non-empty regions of the point cloud, sparse convolution also prevents the distortions and artifacts typically introduced by zero-padding in traditional convolutional methods. This ensures that feature embeddings remain undistorted and spatially accurate, preserving the integrity of positional and contextual information throughout the network. This dual benefit enables the network to effectively represent subtle and complex leakage patterns while maintaining computational efficiency. Finally, layer normalization accelerates the network's convergence process, and stabilizing training, while the introduction of activation layers enables non-linear transformations This empowers the network to learn complex feature representations and enhances its expressive capacity. The output y_i of the embedding layer can be represented by Equation (9):

$$y_i = \sum_{x_i \in X} \{\gamma[Sp_{conv}(\mathbb{G}(\varphi(x_i)))] \oplus \varphi(x_i)\} \tag{9}$$

where γ consists of a layer normalization and an activation layer, Sp_{conv} denotes sparse convolution, $\mathbb{G}$ denotes GECA block, $\varphi(x_i)$ is the serialized inputs, and $\oplus$ denotes the residual connection.

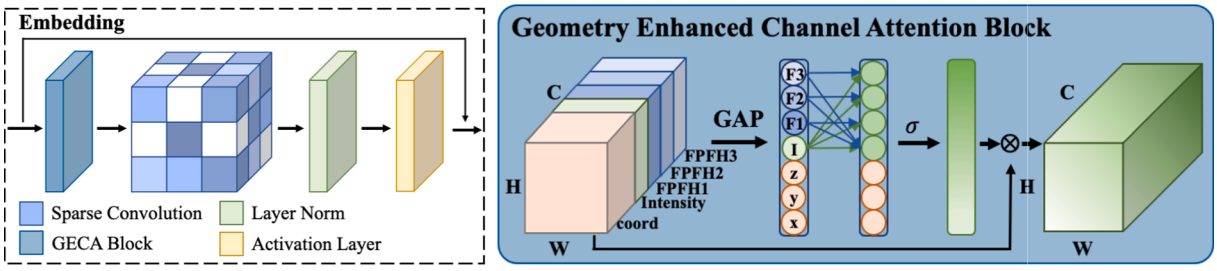
3.3.4. Encode and decode block

The proposed network adopts the U-Net framework, which consists of n encode blocks and n decode blocks. After serialization, intensity and geometric feature channel enhancement, and embedding layers, the resulting high-dimensional feature representations are input into the encoder for feature extraction. Each encode block comprises the following components:

(1) Conditional Position Encoding block

The conditional position encoding block (CPE) is designed to efficiently handle massive data in point cloud transformers, which is realized by employing local attention and relative position encoding (RPE) (Zhao et al., 2021) methods to achieve optimal performance. Compared to the traditional RPE method, CPE enhances efficiency and reduces complexity, making it a preferred choice in the encoding phase of point cloud processing.

The CPE can be described by Equation (10). Sparse convolution operations on the input features play a crucial role in facilitating the extraction of positional relationships within local neighborhoods. This process enables the model to effectively capture the relative spatial positions of the input features, thereby generating robust position-related feature representations. Followed by sparse convolution, a linear layer integrates the local convolutional features to form global feature representations. This step aggregates information across the entire input, which can capture higher-level positional information and relationships that are essential for understanding the spatial context within the point cloud data. Layer normalization is then applied to the output of the linear layer. This normalization process standardizes the feature maps, ensuring that scale differences between different features are minimized. By enhancing the stability and consistency of the positional encoding, layer normalization further improves the ability of the network to learn and generalize spatial relationships effectively. The setup of the CPE block, as depicted in Fig. 9, illustrates how these components work together to encode positional information efficiently within the network. This approach not only enhances the capacity of the



(a) Embedding Layer (b) Geometry Enhanced Channel Attention Block

Fig. 8. Embedding Layer with Geometry Enhanced Channel Attention Block. (a) Embedding Layer first accepts the input point cloud data features, and through GECA block, sparse convolution, layer normalization and activation layer, the feature output obtained is residually connected with the original input features to obtain the final output. (b) GECA block in which GAP denotes global average pooling, σ denotes Sigmoid function and $\otimes$ denotes element-wise product.

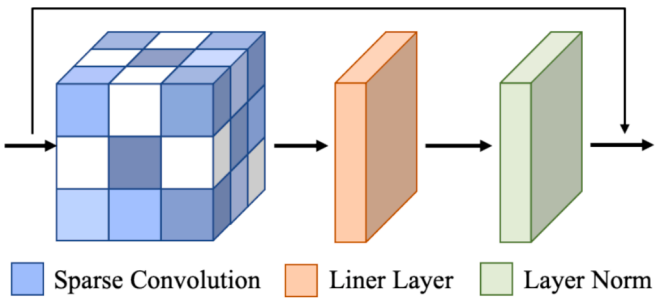


Fig. 9. Conditional position encoding block. CPE block is composed of sparse convolution followed by linear layer and layer normalization.

model to represent spatial relationships accurately but also contributes to overall performance by ensuring robust feature extraction and normalization in point cloud processing task.

$$CPE(y_i) = LN[\ell(Sp_{conv}(y_i))] \oplus y_i \tag{10}$$

where y_i is the output of the previous layer, $\ell(*)$ denotes a linear layer, which can be treated as a fully connection layer, $LN[*]$ indicates the layer normalization and $\oplus$ denotes the residual connection. It is implemented by pre-posing a sparse convolutional layer.

The CPE block plays a pivotal role for accurate tunnel leakage detection by effectively capturing the complex spatial relationships in tunnel point cloud data. Leakages are often associated with specific structural features, such as ring joints, segment joints, and bolt holes, each exhibiting distinct morphological patterns. For instance, leakages along ring and segment joints typically appear as elongated streaks. Accurate detection relies on understanding the spatial distribution and relative positioning of these features. Traditional convolutional approaches, which uses zero-padding to handle irregular or sparse data, can introduce artifacts and distortions, especially as network depth increases. This degradation compromises spatial fidelity and results in the loss of fine-grained information critical for detecting subtle leakages, as illustrated in Fig. 10. In contrast, the CPE block in our framework employs sparse convolution, which inherently avoids the need for zero-padding by operating directly with the non-empty regions of the point cloud data. This method preserves positional information throughout the network, maintaining the spatial accuracy even at deeper depths. Moreover, the use of linear layers and layer normalization further enhances the network's ability to differentiate between various leakage types based on their spatial characteristics.

(2) Serialized Attention Block

In the context of tunnel leakage detection, tunnel structures are complex, with leakages frequently occurring in specific areas such as segment joints, ring joints, and bolt holes. These features present diverse spatial patterns and scales, requiring a model that can capture and analysis these details with precision and efficient.

One of the key features of the Serialized Attention Block is its interaction with serialized point cloud data. By ordering and grouping the serialized patches, the Serialized Attention Block captures a broader range of spatial relationships, as illustrated in Fig. 11. This approach is particularly advantageous for identifying complex and subtle leakage patterns within the tunnel structures, as it allows the attention mechanism to account for spatial dependencies across a wider area, including points that may not be physically adjacent.

The serialized attention block efficiently captures diverse information at different scales during both the encoding and decoding phases, thereby enhancing processing efficiency and model performance. Within this block, multi-head attention is employed to map high-dimensional features into distinct feature spaces. This capability enables a comprehensive analysis of the intricate and varied data patterns found in large-scale tunnel point cloud dataset. In tunnel leakage detection, where leaks can appear in various forms—such as elongated cracks or localized damp spots—this multi-scale attention mechanism ensures that the model can accurately detect and differentiate between these diverse leakage patterns.

Flash attention, as described by Dao et al. (Dao et al., 2022), is utilized within the serialized attention block to compute attention scores more efficiently. By reducing the computational complexity and memory usage typically associated with large-scale point cloud data, Flash attention significantly enhance the performance of the model. It achieves this by employing a block-sparse representation and block-wise processing, ensuring the model remains efficient and scalable, even

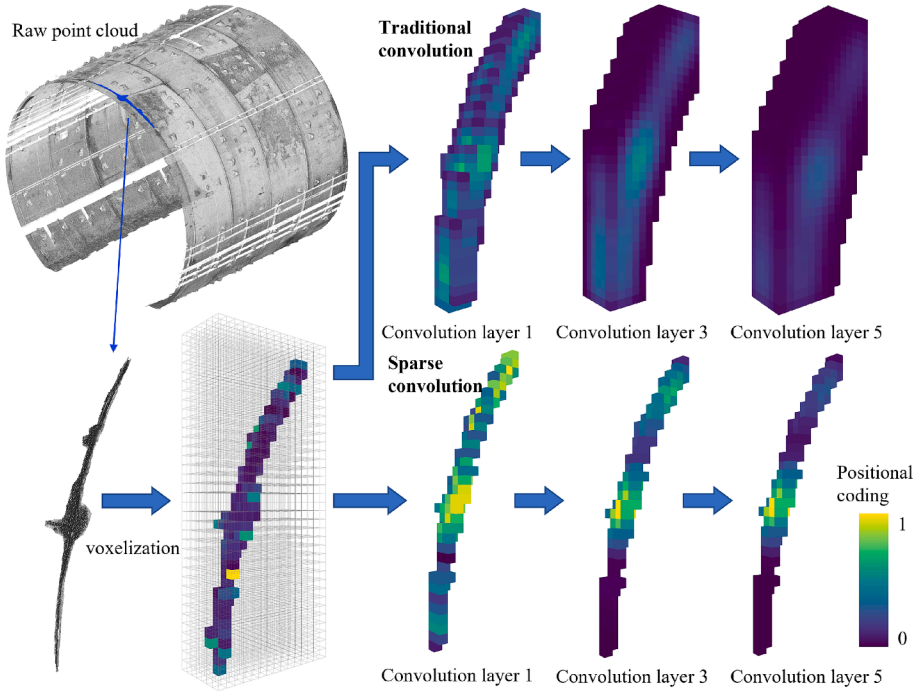


Fig. 10. Visual explanation of CPE block in CPE block. The value of each voxel represents the position coding of the voxel and is normalized.

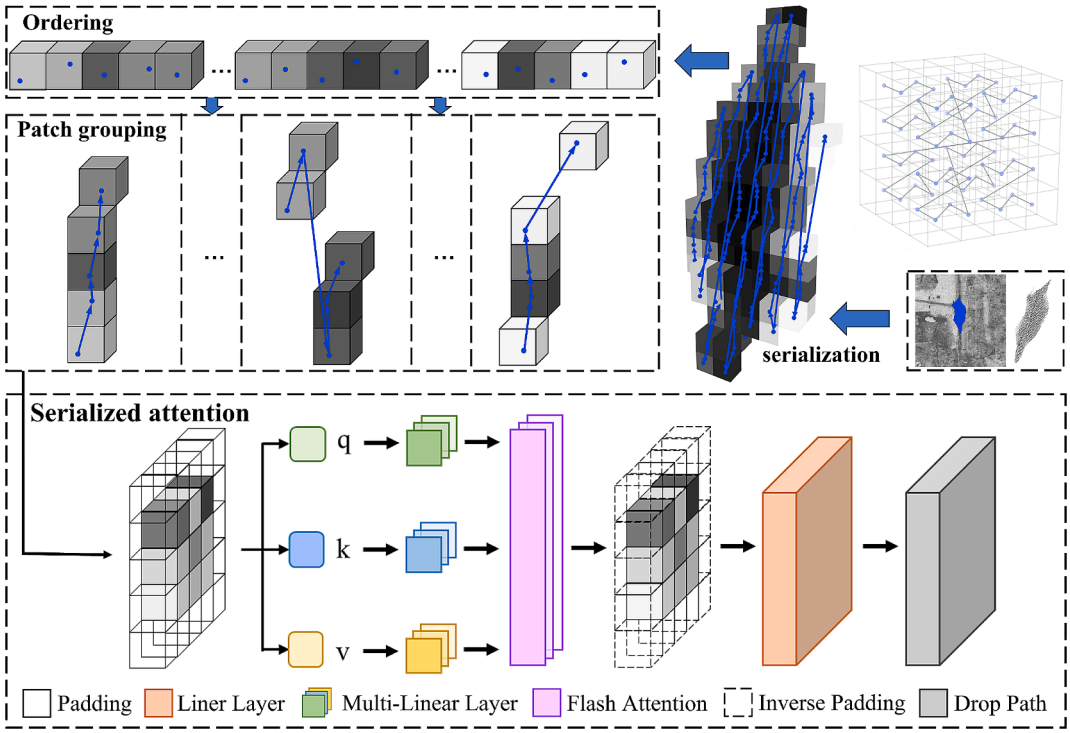


Fig. 11. Serialized Attention Block. After serializing the raw point cloud data, the point cloud patches are ordered and grouped, and then these grouped patches are input into the serialized attention block. The serialized attention block first pads the grouped patches, and then obtains the query (Q), key (K), and value (V) through the Multi linear layer. The attention score is obtained through flash attention, and then the results are de-padded.

when handing extensive sequential data from large tunnel environments. This efficiency is particularly critical in tunnel inspection scenarios, where data volume is substantial and real-time or near-real-time processing is often required. Here is how the Serialized Attention block operates, as illustrated in Fig. 11: (1) Padding: The feature map is padded to ensure that the number of points in each batch is a multiple of the patch size. This simplifies calculations and enhances computational efficiency. (2) queries (Q), keys (K), and values (V) generation: these components are generated from the input features and serve as the basis for computing attention scores. (3) Multi-head Attention Layers: the input features are processed in parallel through multiple attention heads. Each head performs its own transformations using query, key, and value matrices to enhance the ability of the model to capture diverse patterns and relationships within the data. This can be expressed by Equation (11):

$$\begin{aligned} MultiHeadAtten(Q,K,V) &= Concat(head_1, head_2, \cdots, head_n)W_i^O \\ where, head_i &= Atten\left(QW_i^Q, KW_i^K, VW_i^V\right) \end{aligned} \tag{11}$$

where W_i^Q,W_i^K,W_i^V are the transformation matrix for Q, K, V of the i - th head, respectively, and W_i^O indicates the overall transformation matrix. Finally, operations such as MLP and drop path enable the model to learn complex data patterns and improve generalization ability.

(3) Encode Block

In an encoder-decoder architecture network, the encoder plays a pivotal role in transforming input features into meaningful representations that can be effectively utilized by the decoder. It is responsible for capturing and refining essential features from the input data using a series of sophisticated modules designed to enhance performance in tasks such as tunnel leakage detection.

The structure of the encode block is shown in Fig. 5(b). (1) CPE block: Each encode block begins with integrating spatial information by applying a CPE block. (2) Layer Normalization: This is followed by layer normalization to stabilize the training process by standardizing the feature values, thereby improving the efficiency and convergence of the network. (3) Serialized Attention Block: A serialized attention block then focuses on extracting relevant features from different parts of the input at different scales. (4) Drop Path Technique: regularization is introduced through the drop path, which randomly drops during training. This technique helps prevent overfitting by encouraging the network to learn more robust features that generalize well to unseen data. (5): Layer Normalization (again): Another layer normalization step follows to further enhance stability and ensure consistent performance across different batches and layers. (6) Multi-Layer Perceptron (MLP): The MLP further refines the features by applying non-linear transformations. This step helps transform the input into high-level representations that capture more abstract and complex patterns in the data. (7) Final Drop Path Operation: A final drop path operation is applied to reduce overfitting risk and improve the ability of the model to generalize to new data.

(4) Decode Block

The decoder plays a crucial role in reconstructing the spatial structure and detailed information of the original input from the high-level features provided by the encoder. This process involves upsampling the features to their original resolution and integrating skip connections to preserve high-resolution details passed from corresponding encoder layers. These skip connections are crucial for maintaining local detail information and contribute significantly to producing high-quality outputs in the decoder.

As shown in Fig. 5(c), the decode stage differs from the encode block in several aspects: (1) The decoder does not require position encoding, as the input features already contain positional information encoded during the encoding phase. (2) The decoder takes the output from the previous stage and combines it with corresponding outputs from the encoder via skip connections. (3) Serialized attention, MLP, layer normalization, and a residual connection are similar to the encoder, which is applied to obtain the semantic segmentation result.

3.3.5. Loss function

Sample number imbalance of each category in the dataset is a common challenge in point cloud semantic segmentation. To address this issue, employing an appropriate loss function is crucial for achieving accurate and robust model performance. Traditional approaches, such as cross-entropy loss, assign equal weight to each sample. This can lead the model to focus more on the majority classes, thereby biasing the classification towards them. In cases of imbalanced samples, the fewer samples contribute minimally to the loss, causing the model to tend to ignore these samples during optimization, resulting in poor performance on the minority classes. To mitigate the effects of class imbalance and enhance the model performance, Lovasz Loss (Berman et al., 2018) offers a compelling solution, which is inspired by set optimization principles and permutation invariance, making it particularly effective in handling imbalance classes. By directly optimizing the intersection-over-union (IoU) score, Lovasz Loss ensures that each class's contribution to the loss is proportional to its importance in the dataset, thus providing a balanced learning process.

The objective of this study is to achieve binary segmentation of leakages and background. Thus, we employ the Lovasz Hinge Loss function, which combines the Lovasz extension and Hinge loss to measure the distance between predicted values and true labels This IoU-based optimization method focuses on maximizing the overlap between the predicted region and the ground truth, ensuring more accurate segmentation results. This focus on IoU means that even with a small number of minority class samples, inaccurate predictions have a significant impact on the IoU value. As a result, the model is compelled to improve its predictions for these minority classes, enhancing the overall segmentation performance. This approach ensures that each sample, including those from minority classes, contributes to the gradient during optimization. By doing so, the model achieves a better balance in the weights assigned to different classes, leading to improved accuracy and robustness in detecting leakages against the background.

In the binary case, the loss function optimizes the IoU for the foreground class (e.g., leakage class). Initially, the set of mispredicted pixels for class f can be defined by Equation (12), where y^* and $\tilde{y}$ denote the ground truth and the segmentation output, respectively.

$$M_f(y^*,\tilde{y}) = \{y^* = f, \tilde{y} \neq f\} \bigcup \{y^* \neq f, \tilde{y} = f\} \tag{12}$$

With the introduction of the concept of mispredicted pixels, an IoU-based loss function $M_f\Delta_{IoU}$ is constructed, as shown in Equation (13). Subsequently, the Hinge Loss with the prediction of point i is introduced to the Lovasz Hinge Loss framework to enhance the segmentation performance. The Hinge Loss for point i can be defined by Equation (15), where F represents the $i-th$ element of the output scores of the model and $\overline{\Delta_{IoU}}$ is the smooth extension of IoU-based loss, which is the tight convex closure of Δ_{IoU}. The specific extension process for the Lovasz Hinge Loss, as detailed in Equations (8–9) of (Berman et al., 2018), involves constructing a tight convex closure of the IoU-based loss. This smooth approximation ensures that the loss function is differentiable, facilitating effective optimization.

$$M_f\Delta_{IoU} : M_f \in \{0,1\}^p \Rightarrow \frac{|M_f|}{|\{y^* = f\} \bigcup M_f|} \tag{13}$$

$$\mathscr{L}_{Hinge} = \max(1 - F_i(x)y^*, 0) \tag{14}$$

$$\mathscr{L}_{LovaszHinge} = \overline{\Delta_{IoU}}\left(\mathscr{L}_{Hinge}(F)\right) \tag{15}$$

3.4. Evaluation metrics

In assessing the effectiveness of our point cloud deep learning network for tunnel leakage detection, we utilize several tailored evaluation metrics for 3D semantic segmentation tasks. These metrics are crucial in evaluating the performance and reliability of the model in distinguishing leakage areas from non-leakage regions. The Mean Intersection over Union (mIoU) and Mean Accuracy (mAccuracy) serve as overarching measures, providing insights into overall segmentation performance across all classes. However, our primary attention lies in the metrics pertinent to the leakage class. The Intersection over Union (IoU) metric quantifies the alignment between predicted and ground truth masks, offering a nuanced understanding of segmentation accuracy for leakage areas. This metric is particularly useful for assessing how well the model identifies and delineates the leakage region.

The evaluation metrics are defined by Equations 16–19: N denotes the total number of classes. TP_i represents the number of true positives for the class i, indicating the samples correctly classified as positive. FP_i refers to the number of false positives for class i, signifying the samples incorrectly classified as positive. FN_i represents the number of false negatives for the class i, indicating the samples incorrectly classified as negative. Finally, FN_i denotes the number of true negatives for class i, signifying the samples correctly classified as negative.

$$mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i + TN_i} \tag{16}$$

$$mAccuracy = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i} \tag{17}$$

$$IoU\_leakage = \frac{TP}{TP + FP + FN} \tag{18}$$

$$Accuracy\_leakage = \frac{TP + TN}{TP + FP + TN + FN} \tag{19}$$

To evaluate the efficiency of the proposed model, we also consider metrics such as training time, inference time, and model parameter count. These metrics are essential for assessing the efficiency of the model and its utilization of computational resources. Additionally, the model parameter count indicates the number of parameters the model needs to learn, reflecting its complexity and resource requirements.

4. Experimental results

4.1. Experimental settings

4.1.1. Datasets

S3DIS_Leakage. We created a specialized point cloud dataset named S3DIS_leakage dataset. This dataset comprises point cloud data collected from real-world scenarios, specifically designed for shield tunnel leakage detection. The S3DIS leakage dataset was conducted using FARO 330 LiDAR scanning system, which captured raw point cloud data from subway stations between Jiqingmen Street and Xinglong Street on Line 2 of the Nanjing Metro. This dataset includes both 3-D coordinates and intensity information.

First, we preprocessed the raw point cloud data by removing outliers and auxiliary facility point clouds. Furthermore, following the Stanford large-scale 3D Indoor Spaces Dataset (S3DIS) (Armeni et al., 2017), we partitioned the point cloud data into six areas. Each area was further segmented into multiple tunnel blocks at 15-meter intervals along the direction of the tunnel to prevent potential memory overflow issues during the training phase. As shown in Fig. 12, Area 5 is served as the validation set, while the remaining Areas are for training. Within our dataset, we meticulously labeled 4,427 instances of leakages, which collectively span 22.71 million points. These leakages account for approximately 1.5 % of the total data volume, with the background class outnumbering the leakage samples by a factor of 65. The notable class imbalance poses a common challenge in deep learning applications, which is addressed in our research.

After creating the dataset, we performed data cleaning to ensure the accuracy of leakage annotations. Given that the annotations relied on

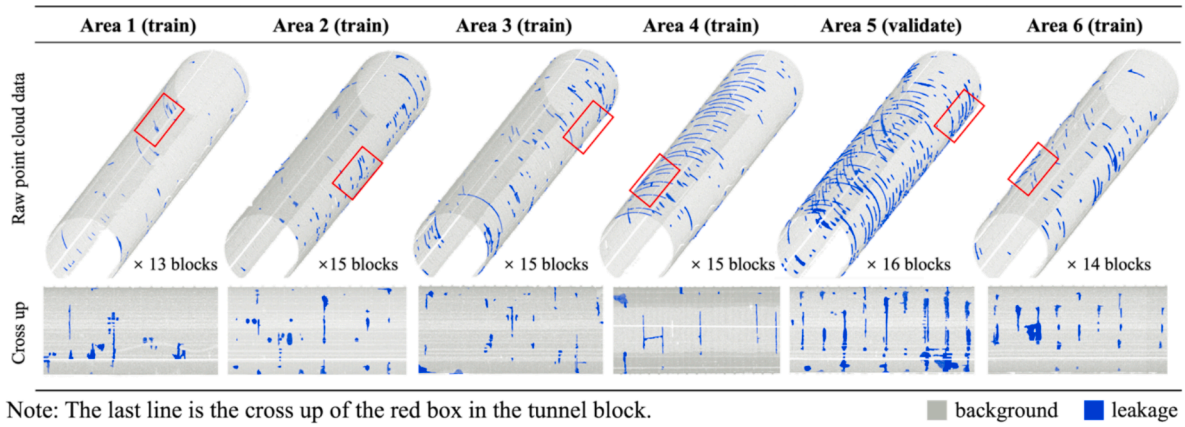


Fig. 12. S3DIS_leakage dataset.

manual labeling, inherent inaccuracies were unavoidable, which include misclassifying non-leakage points as leakage and overlooking genuine leakage points. To this end, we employed a statistical approach based on probability density of the point intensity. A 95 % confidence interval was determined based on standard deviations of the point intensity. For points leakage class, we fitted out those exceeding the upper threshold of the confidence interval, i.e., $+\sigma$. Conversely, points classified as the background were fitted out if their intensity fell below the lower threshold, i.e., $-\sigma$. This meticulous filtering process ensured accurate point categorization, enhancing dataset reliability for model training and evaluation.

Test datasets. The Test dataset was collected from various subway lines in different cities, each located in distinct tunnel environments. These environments are differed in geological conditions, operational durations, and groundwater levels, which impact tunnel humidity and dust accumulation. In fact, there is a noticeable correlation between the dust accumulation of subway tunnels and their operational duration. Generally, longer operational periods tended to result in higher levels of accumulated dust. Additionally, the humidity within subway tunnels is closely related to the groundwater level, with higher groundwater levels contributing to increased humidity within the tunnel environment.

Specially, Dataset 1 was sourced from the Nanjing Metro Line 10, covering 150 m of tunnel length, which began operation in 2022. This line traverses a region characterized by soft soil and silt. In contrast, Dataset 2 was collected from the Nanjing Metro Line 3, spanning approximately 500 m and operational since 2015. The geological conditions along this line are notably diverse, which traverses riverine sand layers, lake sediments, and karst formations. Additionally, Dataset 3 was sourced from the Wuxi Metro Line 2 and has been operational since 2020, covering approximately 500 m of tunnel length. This tunnel is situated in an area with primarily alluvial soils, resulting in distinct groundwater behaviors and related humidity conditions within the tunnel. Notably, the data used in this study from the Nanjing Metro Lines 10, 3, and Wuxi Metro Line 2 are situated in areas with progressively lower groundwater levels. Consequently, it is expected that the humidity within the tunnels decreases accordingly, which in turn affects the occurrence of leakage. Nanjing Metro Line 10 experienced the highest number of leakages, Line 3 had a moderate amount, while Wuxi Metro Line 2 reported the fewest leakages. The above three datasets are from shield-structured tunnels. To validate the generalization of our method, we additionally collected data from a horseshoe-shaped tunnel, which differs structurally from shield tunnels. This dataset, Dataset 4, was sourced from the Nanjing Metro Line 7, which has been operational since 2021.

4.1.2. Experimental environment and settings

The model was implemented on a workstation equipped with an Intel Core i9-11900KF CPU and an NVIDIA RTX 4080 (16 GB VRAM), operating under Ubuntu 20.04. The computations of the model were facilitated using PyTorch version 11.7, executed in the Python programming language. All programming tasks were conducted within a virtual environment established via Anaconda, which incorporated CUDA version 11.8 and cuDNN version 8.04, utilizing Python 3.8.

In the configuration of GEPT-Net, we employed eight workers and set the batch size to four. For optimization, the AdamW optimizer was used, featuring a learning rate of 0.006. The learning rate was modulated using a cosine annealing strategy. To mitigate the risk of overfitting, a dropout probability of 0.3 was implemented in the path dropout regularization technique. Additionally, to enhance model robustness, data augmentation techniques such as Random Scale and Random Flip were applied to the input data. The encoder of our model is structured into four stages, each containing 2, 2, 6, and 2 blocks, respectively. The multi-head self-attention mechanism within these stages utilizes progressively increasing number of heads: 4, 8, 16, and 32, aligning with the complexity of the tasks handled at each stage. Correspondingly, the output channels for these stages are configured to 64, 128, 256, and 512. In contrast, the decoder maintains a uniform configuration across its stages, each comprising 2 blocks. The number of attention heads in the multi-head self-attention mechanism of the decoder stages follows a pattern of 4, 4, 8, and 16.

4.2. Segmentation results

Fig. 13 presents the visualization results of tunnel leakage segmentation by GEPT-Net. These results highlight the capability of the model to accurately detect and segment leakage areas within the tunnel structures. A comparison with the ground truth reveals a high correspondence, illustrating the model's precision and reliability in leakage identification. Additionally, to further evaluate the generalizability and robustness of our network, we conducted additional experiments on Test datasets. As shown in Table 1, the mIoU of three additional Dataset only decreased by 2 % to 3 % compared to the results on the original dataset, indicating minimal performance degradation when applied to new and varied data. These results confirm the adaptability and robustness of our proposed network across diverse tunnel environments and data distributions, demonstrating its strong potential for deployment in real-world scenarios involving tunnels with differing characteristics. To further validate the generalizability of our proposed method, we tested on a segment of a horseshoe-shaped tunnel (Dataset 4). As shown in Table 1,

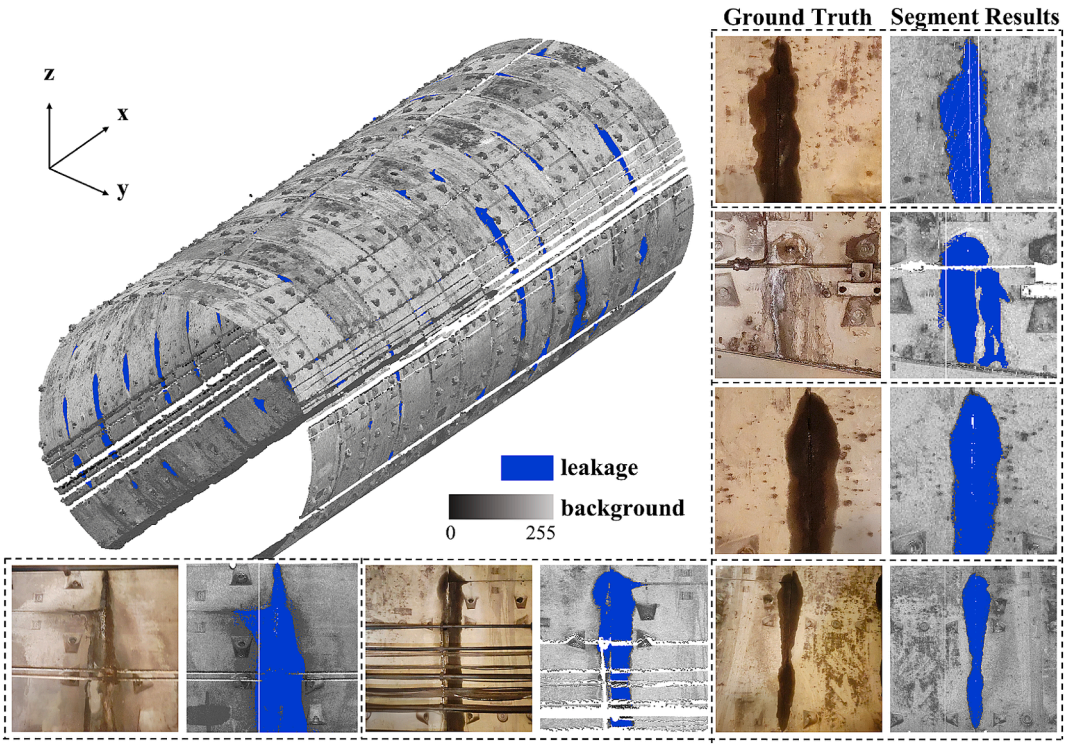


Fig. 13. Segmentation results of GEPT-Net. The left and right column of each dotted box denotes the ground truth and segmentation results, respectively.

Table 1
Performance comparison of GEPT-Net across different tunnel datasets. Unit: %

	Structure	Data source	mIoU	mAcc	IoU_leak	Acc_leak
S3DIS_leakage	Shield	Najing, Line 2	92.38	94.50	85.31	89.22
Test dataset 1	Shield	Najing, Line 10	90.32	91.43	80.95	82.89
Test dataset 2	Shield	Najing, Line 3	89.56	98.89	79.23	97.88
Test dataset 3	Shield	Wuxi, Line 2	89.88	92.16	79.94	84.37
Test dataset 4	Horseshoe	Najing, Line 7	88.12	98.25	77.30	97.31

the results showed that the IoU for the leakage category reached 77.3 %, demonstrating that our model performs robustly across different tunnel geometries and meets the accuracy requirements for practical engineering applications.

4.3. Comparison experiments

4.3.1. Comparison with 2-D intensity images methods

Although we have thoroughly discussed the limitations of image-based deep learning techniques for tunnel leakage detection in previous sections, we further conducted comparative experiments to substantiate our viewpoint. Specifically, we conducted a comparative analysis of four prominent 2D deep learning networks for tunnel leakage semantic segmentation: YOLOv8 (Jocher et al., 2023), DeepLabv3 (Chen et al., 2018), U-Net (Ronneberger et al., 2015), and PSPNet (Zhao et al., 2017). YOLOv8x was selected due to its optimization for large-scale applications and its superior performance in preliminary tests. To maintain fairness, we opted not to discuss training and inference times due to inconsistent data formats across models, focusing instead on comparative efficiency through model parameters. As detailed in Table 2, GEPT-Net improves accuracy and IoU metrics significantly, surpassing the closest competitor, DeepLabV3+ in IoU_leakage by 11.64 %. For qualitative analysis, we generated intensity maps along tunnel vaults from the 3D output of GEPT-Net, as shown in Fig. 14. Fig. 15 compares leakage semantic segmentation results from different image-based methods. GEPT-Net demonstrates superior performance, accurately identifying almost all leakages. The suboptimal performance of 2D networks can be attributed to their inability to effectively process the unique structural characteristics of shield tunnels. In evaluating model complexity, GEPT-Net strikes a balance between performance and efficiency, achieving superior results to models like YOLOv8 and DeepLabV3+ with fewer parameters, as shown in Table 2.

Table 2
Comparison results of image-based deep learning networks.

Model	mIoU	mAcc	IoU_leak	Acc_leak	Para.
YOLOv8	65.40	90.12	32.82	80.25	71.80
DeepLabV3+	86.83	93.62	73.67	87.25	55.12
U-Net	82.41	92.38	64.83	84.77	**24.89**
PSPNet	77.30	88.95	54.80	78.01	46.71
GEPT-Net (ours)	**92.38**	**94.50**	**85.31**	**89.22**	46.17

Note: The unit for IoU and Acc is %. The unit for parameter (Para.) is million.

4.3.2. Comparison with 3-D point clouds methods

Although 2D methods offer valuable insights, the inherent complexity of 3D data underscores the need for rigorous evaluation against specialized point cloud-based deep learning networks. Therefore, we conducted a comprehensive comparison of GEPT-Net with four cutting-edge 3D deep learning models that have demonstrated top-tier performance in public datasets. These models include Minkowski U-Net (MinkU-Net) (Choy et al., 2019), Sparse U-net (SpU-Net) (Liu et al., 2022b), Point Transformer (PTv1) (Zhao et al., 2021), and Point Transformer v2 (PTv2) (Wu et al., 2022).

As detailed in Table 3, the proposed GEPT-Net exhibits superior

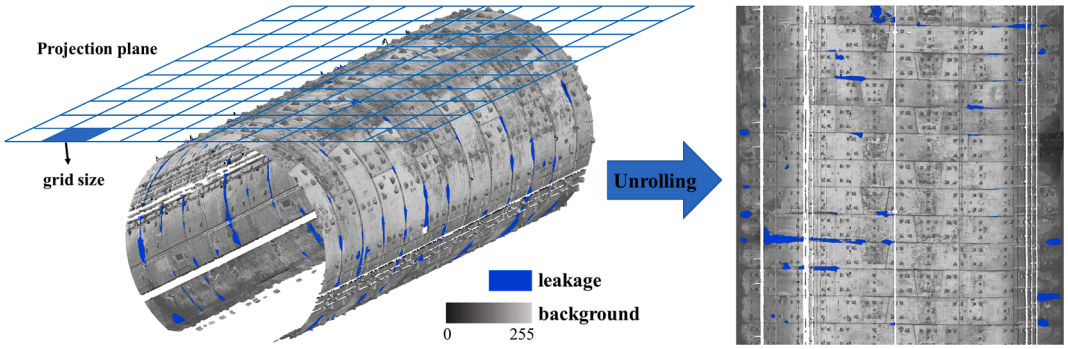


Fig. 14. The process of unrolling and projeting the point clouds into an intensity map.

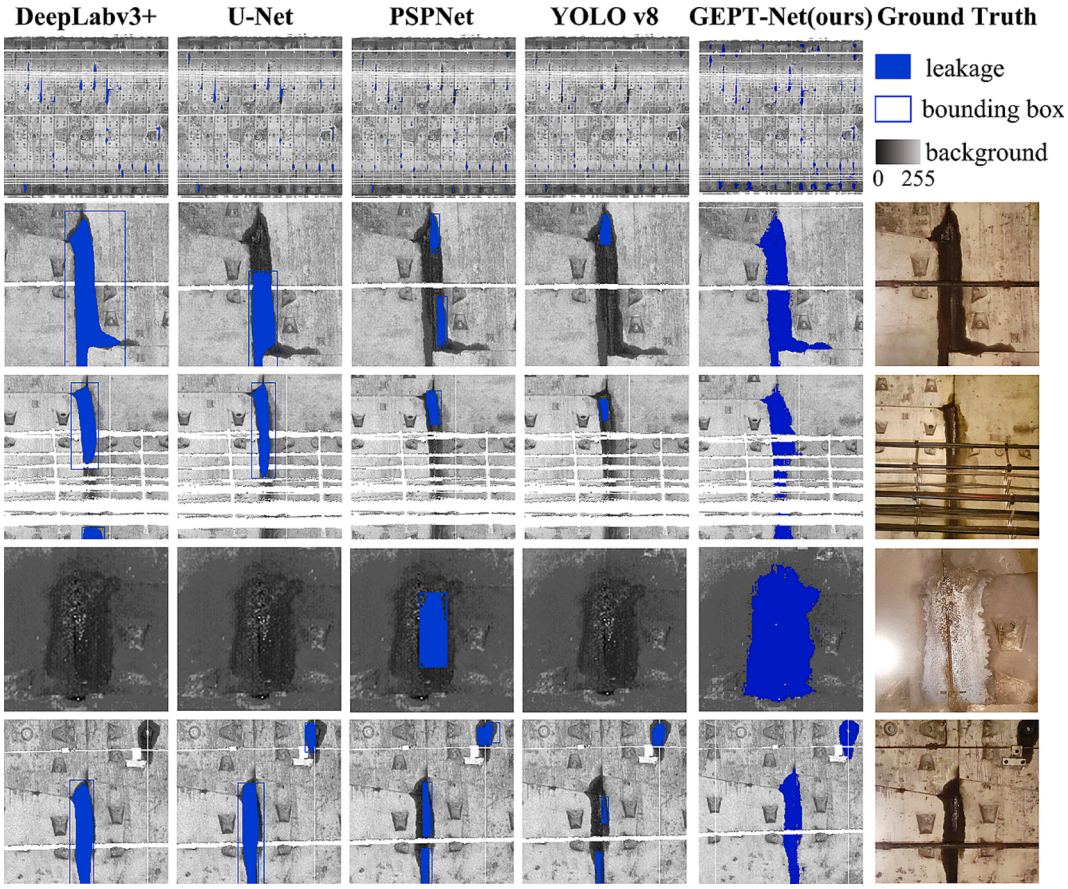


Fig. 15. Qualitative results of leakage semantic segmentation of different image-based methods.

Table 3
Comparison results of point cloud-based deep learning networks.

	mIoU	mAcc	IoU_leak	Acc_leak	Para.	Train. time	Infer. time
MinkU-Net	83.92	89.57	68.95	79.63	**37.86**	2d 5 h 57 m	0.39
SpU-Net	83.84	91.06	68.85	82.75	39.15	2d 6 h 26 m	**0.04**
PT v1	81.22	92.88	63.97	86.88	77.67	3d 2 h 37 m	2.15
PT v2	87.17	90.24	75.17	80.72	39.08	2d 2 h 57 m	1.33
GEPT-Net (ours)	**92.38**	**94.50**	**85.31**	**89.22**	46.17	**2d 1 h 43 m**	0.07

Note: The unit for IoU and Acc is %. The unit for parameter (Para.) is million. The format for training time (Train. time) is *x* days (d) *y* hours (h) *z* minutes (m), and the unit for inference time (Infer. time) is second.

performance, notably excelling in both mIoU and mAcc compared to other evaluated methodologies. Specifically, GEPT-Net shows significant surpass PT v2 by 10.14 % in IoU_leakage and 8.50 % in Acc_leakage. Fig. 16 illustrates the performance of GEPT-Net in complex scenarios, such as leakages obscured by attachments or those located at bolt holes, where the model demonstrates adept handling. In comparison, other methods exhibit varying degrees of missed predictions and false positives. This is particularly evident in their incorrect identification of bolt holes and grout holes, which exhibit point cloud intensities similar to those of leakages. In terms of efficiency, MinkU-Net and SpU-

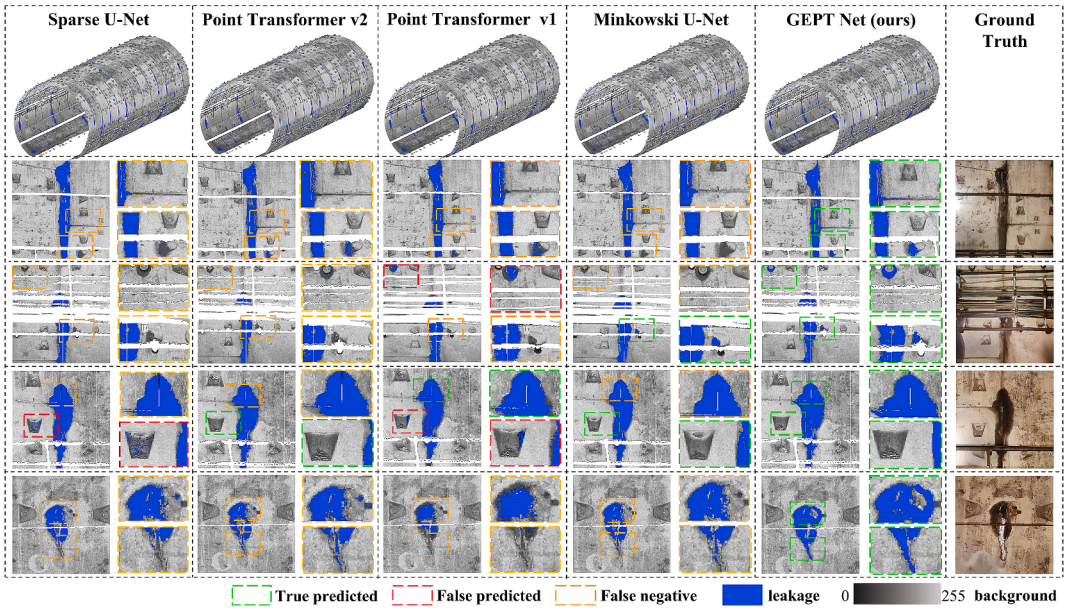


Fig. 16. Qualitative results of leakage semantic segmentation of different point cloud-based methods. In each dashed box, the image on the left is an instance of the leakage detection result corresponding to the ground truth, and the two small dashed boxes on the right are the cross up of the instance detection result.

Net, utilizing sparse convolution operations for point cloud processing, which are computationally efficient and minimize memory usage. However, they often fail to adequately capture the complex geometric features essential for precise leakage identification. PT v1, despite its lower parameter count, incurs prolonged inference times without a corresponding enhancement in leakage detection efficacy. Conversely, PT v2, although featuring a higher parameter count, does not achieve the precision necessary for effective leakage detection. In contrast, GEPT-Net achieves superior mIoU and mAcc while balancing training time and inference efficiency. It trains on a billion-point dataset in 2 days, with an inference time of 0.07 s per batch and 46.17 million parameters, effectively capturing geometric features for tunnel leakage detection.

4.3.3. Comparison on the S3DIS dataset

To further evaluate the generalization of GEPT-Net, we conducted comparative experiments on the S3DIS dataset, a widely recognized benchmark for semantic segmentation tasks. GEPT-Net was evaluated alongside state-of-the-art 3D deep learning models under identical configuration settings to ensure a fair comparison. As summarized in Table 4, GEPT-Net achieved a mIoU of 61.7 %, which represents the second-best performance among all methods. While Point Transformer V2 achieved the best performance, it is specifically optimized for general-purpose semantic segmentation. Such networks may face challenges when applied to specific downstream tasks like tunnel leakage detection. GEPT-Net's strong competitive performance highlights its robustness and generalization capabilities beyond its specific design focus on tunnel leakage detection. These results emphasize GEPT-Net's adaptability and its ability to balance domain-specific specialization with broader applicability.

5. Discussion

5.1. Selection of FPFH features

Geometric features represent a significant advancement in the detection of tunnel leakage, and the selection of these features is a critical step in the process. As discussed in the previous sections, we have provided detailed justifications for selecting FPFH5, FPFH16, and FPFH27 as the primary geometric features. To further assess the robustness and generalizability, we selected a section of tunnel data from each dataset in the Test datasets, which contains various tunnel environmental conditions, and computed these FPFH features, followed by visualization of these features. Our experiments demonstrate that FPFH5, FPFH16, and FPFH27 effectively differentiate between bolt holes, joints, and tunnel lining surfaces. As shown in Fig. 17, the values of these FPFH features in joint and bolt hole regions are clearly distinct from those on tunnel linings. Specifically, FPFH27 and FPFH5 show

Table 4
Category-wise segmentation results evaluated on S3DIS validation set. The evaluation metric for each category is IoU (%). **Bold fonts** indicate the best performance in the comparison test, underlined font indicate suboptimal performance in comparison tests.

Method	**mIoU**	**mAcc**	ceiling	floor	wall	beam	column	window
Minkow-UNet	60.1	66.5	94.2	97.3	81.4	0.0	19.2	41.2
PTv1	56.3	64.7	92.5	**97.9**	78.5	0.0	22.6	42.4
PTv2	**65.1**	**71.9**	**94.5**	97.7	81.5	0.0	**33.4**	49.5
Sp-UNet	60.3	66.4	92.1	95.9	79.2	0.0	26.2	37.9
GEPT-Net (ours)	61.7	69.4	94.0	97.2	**82.2**	0.0	33.0	**50.8**

	door	table	chair	sofa	bookcase	board	clutter
Minkow-UNet	**57.7**	81.4	87.6	35.7	66.5	**63.8**	54.2
PTv1	37.7	73.5	81.5	36.5	66.5	53.2	49.7
PTv2	50.0	**83.2**	**92.2**	**72.5**	**73.0**	60.0	**58.9**
Sp-UNet	56.1	78.6	89.6	51.4	66.6	58.0	52.8
GEPT-Net (ours)	46.8	78.5	87.9	53.1	66.6	60.6	50.1

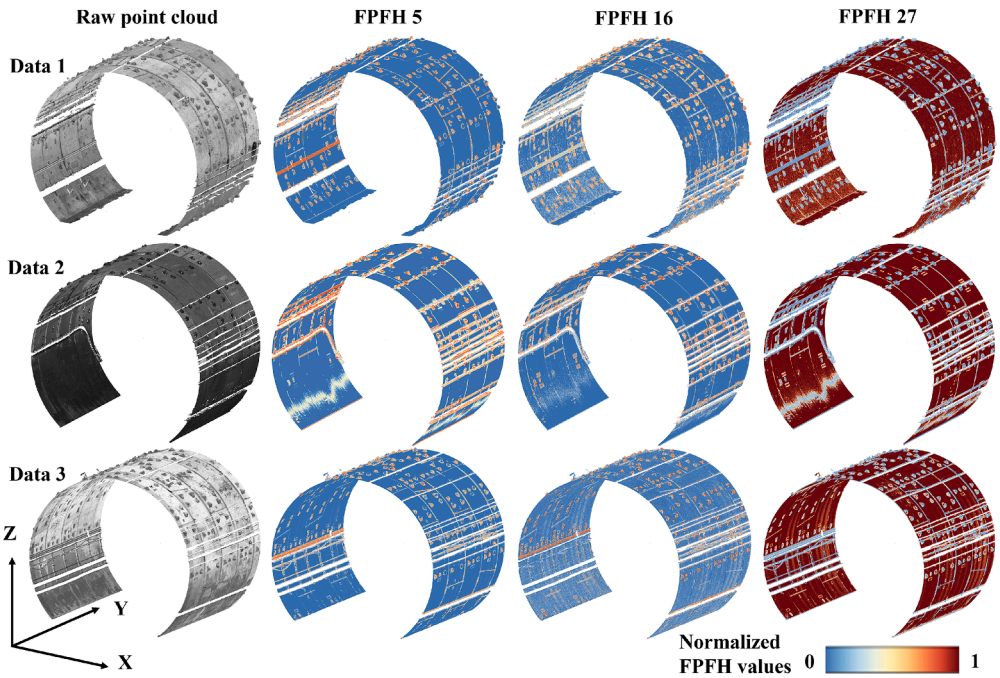


Fig. 17. Visualizationof the three FPFH features used in the proposed GEPT-Net on point cloud data from different tunnel environments. The first column displays the raw point cloud data obtained from various tunnels, while the three columns on the right illustrate the visualization of FPFH5, FPFH16, and FPFH27 feature calculations in different tunnel scenarios.

strong discriminatory power for differentiating joints and bolt holes, while FPFH16 is particularly effective in distinguishing bolt holes. The consistency of these FPFH features across different tunnels is attributed to the uniform construction methods used by tunnel boring machines (TBMs) and the standardized shield tunnel assembly processes. Therefore, the introduction of these FPFH features enhances the generalizability of the network across diverse tunnel environments.

Moreover, we conducted additional ablation studies to further analyze the importance of the FPFH features. As shown in Table 5, the baseline model, without FPFH features, registers a mIoU of 76.98 % and an IoU of 55.50 % for the leakage class, reflecting the challenges faced by models in accurately classifying leakage areas without the incorporation of geometric features. The incorporation of individual FPFH features significantly enhances model performance. Specifically, the inclusion of any FPFH notably improves the accuracy and IoU of leakage class, highlighting the crucial role of geometric features in tunnel leakage detection. Interestingly, combining any two geometric features from the FPFH set does not exceed the performance achieved with a single FPFH feature and tends to slightly underperform. This phenomenon could be attributed to the significant differences in the geometric features between the leakage and background classes in FPFH2, while the features in FPFH1 and FPFH3 exhibit slight similarities across these classes. The most significant performance enhancement is observed when all three FPFH features are combined. These results illustrate the addition of geometric features enables the network to achieve exceptional localization precision and significantly improve prediction accuracy for the leakage class. This integration of comprehensive geometric characteristics across different orientations markedly enhances the ability of the network to accurately segment leakage points within complex tunnel environments.

Notably, the enhancements in detection accuracy are achieved with only a minimal increase in the parameter count. Despite expanding the input dimensions of our network to include three additional FPFH geometric features, the total number of model parameters increased only marginally—only 0.12 million increased. This modest increase demonstrates our model's capacity to integrate additional features without significantly complicating its structure. Additionally, the inclusion of more FPFH features does lead to longer training times, however, GEPT-Net maintains a substantial efficiency advantage over other deep learning networks, highlighting its practical utility in scenarios where both accuracy and computational efficiency are paramount. In summary, the results emphasize the pivotal role of geometric features in precisely identifying leakage points, validating the effectiveness of our approach. The demonstrated improvements highlight the potential of our network for practical deployment in tunnel maintenance and monitoring systems.

Table 5
The results of different combinations of FPFH features.

	mIoU	mAcc	IoU_leak	Acc_leak	Para.	Train. time
wo FPFH	76.98	82.23	55.50	64.97	**46.158**	**22 h 38 m**
FPFH 1	87.82	89.68	76.40	79.49	46.162	1d 1 h 43 m
FPFH 2	89.26	91.31	79.19	82.76		1d 1 h 1 m
FPFH 3	87.24	89.15	75.27	83.43		1d 2 h 8 m
FPFH 1 + 2	89.21	91.63	79.11	78.44		1d 8 h 46 m
FPFH 1 + 3	88.04	91.41	76.85	83.08	46.166	1d 5 h 13 m
FPFH 2 + 3	88.24	89.88	77.22	79.87		1d 10 h 33 m
FPFH 1 + 2 + 3	**90.52**	**93.18**	**81.64**	**86.54**	46.170	2d 4 h 20 m

Note: The unit for IoU and Acc is %. The unit for the parameter is million. wo denote without.

5.2. *Influence of GECA block*

We assessed the performance of GEPT-Net both with and without the inclusion of the GECA block. These results, detailed in Fig. 18 and Table 6, reveal a noticeable performance degradation of 3.67 % in IoU_leakage when the GECA block is omitted. GEPT-Net, when augmented with geometric features, effectively localizes leakages within segment joints. However, groundwater infiltration through these joints also spreads onto the adjacent lining surface, causing additional leakage

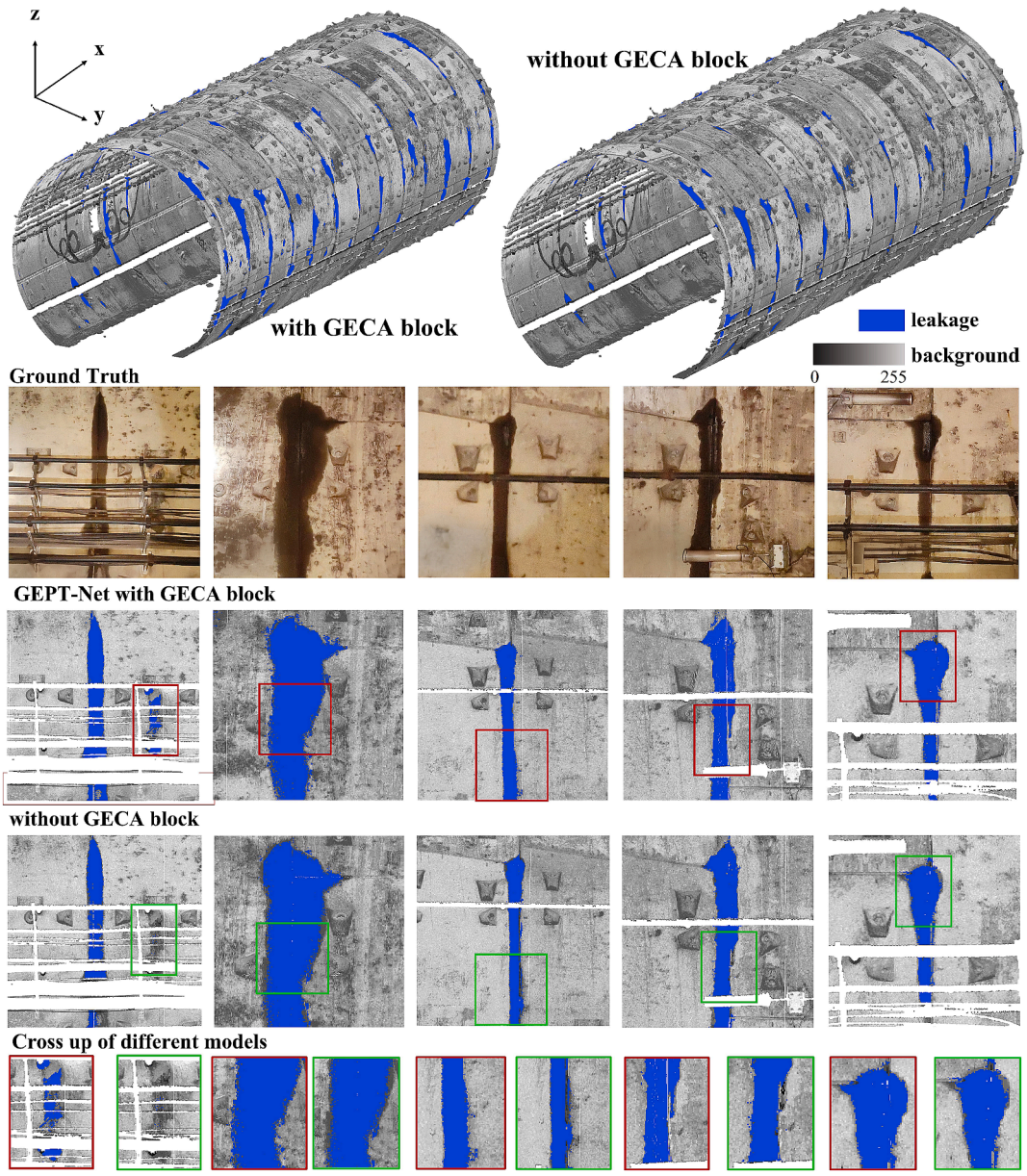


Fig. 18. Segmentation results of GEPT-Net with or without GECA block. The red and green wireframes represent the cross up of instance detection results for GEPT Net with/without GECA block, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
The results of GEPT-Net with or without GECA block.

	mIoU %	mAcc %	IoU_leak %	Acc_leak%
wo GECA block	90.52	93.18	81.64	86.54
with GECA block	**92.38**	**94.50**	**85.31**	**89.22**

Note: wo denotes without.

points. The FPFH features of the lining surface closely resemble those of non-leaking areas, which may result in missed detections of surface leakages, as indicated by the green boxes in Fig. 18. The GECA block significantly enhances the interaction between geometric feature channels and intensity channels through a channel attention mechanism. This adaptation enables the model to utilize intensity channels for accurately identifying leakages on the lining surface when geometric feature channels are insufficient. As demonstrated by the red boxes in Fig. 18, the inclusion of the GECA block allows the model to precisely localize leakage areas on both the lining surface and the segment joints, thereby substantially improving detection accuracy.

5.3. *Significance of serialization pattern*

To illustrate the enhancement achieved through point cloud serialization, we conducted a detailed comparative analysis of our network. The results, as presented in Table 7, showcase the performance metrics associated with different serialization patterns. The application of the Hilbert curve method demonstrated effective performance within our

Table 7
The results of different patterns of point cloud serialization.

	mIoU %	mAcc %	IoU_leak %	Acc_leak%
Z-order	87.62	90.32	76.03	80.83
Hilbert curve	84.68	88.81	70.38	77.97
Z-order + Hilbert	**90.52**	**93.18**	**81.64**	**86.54**

network. However, the Z-order method provided a noticeable enhancement in the evaluated metrics. Most notably, the combination of both Z-order and Hilbert methods, referred to as Z-order + Hilbert, yielded the most significant improvements. Special attention was given to the leakage detection metrics (IoU_leakage and Acc_leakage), where the Hilbert curve method exhibited lower performance. In contrast, the Z-order method showed a marked improvement in the leakage-specific metrics, highlighting its effectiveness in accurately identifying areas prone to leakage. Furthermore, the combined Z-order + Hilbert method demonstrated substantial gains, increasing IoU_leakage by approximately 15.9 % and Acc_leakage by about 11 %. As discussed in the methods section, the Z-order curve's ability to traverse different spatial locations allows adjacent points to span considerable distances. This serialization effectively encodes local details, enabling the receptive field of each input patch to cover extensive distances within the point cloud, enhancing leakage detection accuracy. These results demonstrate that integrating Z-order and Hilbert curves significantly enhances our network's performance by expanding the receptive field, enabling the network to capture both local details and global context, thereby improving detection accuracy.

5.4. Influence of loss function

To further assess the effectiveness of the Lovasz Hinge Loss in handing extreme class imbalance, we conducted an additional experiment on Dataset 3 of Test datasets, where background-to-leakage class ratio reached 334:1. This severe imbalance poses a significant challenge to model performance, particularly in detecting minority classes. Lovasz Hinge Loss, which directly optimizes the IoU for each class, is particularly well-suited for such scenarios, effective optimization of minority class samples during training. To evaluate its robustness, we trained our GEPT-Net on both the S3DIS_leakage dataset and the Test dataset 3, using consistent network parameters. As shown in Table 8, GEPT-Net accurately identified tunnel leakages even under extreme class imbalance. These results confirm that the Lovasz Hinge Loss is capable of handling highly imbalanced datasets, making it a suitable function selection for various imbalance conditions, including those encountered in tunnel leakage detection.

Additionally, we conducted a detailed comparative analysis for several loss functions. These include Cross Entropy Loss (De Boer et al., 2005), Focal Loss (Lin et al., 2017), and Dice Loss (Milletari et al., 2016), which are mathematically delineated in Equations (20) to (22), respectively:

$$\mathscr{L}_{CrossEntropy} = H(p,q) = -\sum_{i=1}^{n} p(x_i)\log(q(x_i)) \tag{20}$$

where $p(x_i)$ denotes the true probability distribution, and $q(x_i)$ represents the predicted probability distribution.

$$\mathscr{L}_{Focal} = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{21}$$

where p_t denotes the probability of the ground truth, α_t and γ are modulation factors designed to address the imbalance between the number of positive and negative samples and to control the disparity between easy and hard samples. We set the α_t and γ to 0.25 and 2, respectively, as suggested by (Lin et al., 2017).

$$\mathscr{L}_{Dice} = 1 - \frac{2\sum_i p_i g_i}{\sum_i p_i^2 g_i} \tag{22}$$

where p_i, q_i represent the predicted result and the true label at the i-th point, respectively.

As demonstrated in Table 9, our S3DIS_leakage dataset exhibits significant class imbalance, with the background class being substantially more prevalent than the leakage class. This imbalance poses a challenge for typical segmentation tasks. Cross Entropy Loss, widely adopted in segmentation applications, unfortunately does not inherently address the class imbalance. As a result, its performance on our dataset was suboptimal, with a lower mIoU and mAcc. In contrast, Focal Loss and Dice Loss, which are designed to address class imbalance by down-weighting well-classified examples, yielded only marginal improvement. This highlights its limited effectiveness in case of extreme class disparity. This limitation is due to its focus on overlap, which inadequately penalizes misclassifications in the minority class. In contrast, the Lovasz Hinge Loss, specifically designed to optimize the IoU, excels in handling imbalanced datasets. Its superior performance, as shown in Table 9, can be attributed to its direct focus on IoU optimization. This approach ensures a more balanced treatment of the minority leakage class relative to the majority background class. Furthermore, metrics specific to the leakage class showing significant improvement with the application of Lovasz Hinge Loss, highlighting its efficacy in addressing class imbalance challenges.

To further validate the effectiveness of Lovasz Hinge Loss in handling imbalanced datasets, we analyzed the loss and IoU polylines during training for different loss functions. As shown in Fig. 19, cross-entropy loss exhibits fast convergence and maintains a relatively low loss value throughout training. However, this is not necessarily beneficial in scenarios with class imbalance. Cross-entropy loss, designed to optimize global accuracy, tends to be biased towards the majority class (i.e., background), neglecting the minority class (i.e., leakage). This bias results in stable loss values that do not improve minority class predictions. Dice Loss, while emphasizing the minority class by assigning higher weights to difficult samples, introduces optimization instability, leading to oscillations in the loss polyline and a complex training trajectory. Focal Loss achieves smoother loss reduction and moderate convergence. However, it struggles to balance optimization between majority and minority classes, as evidenced by persistent IoU polyline oscillations during training. Similar challenges are observed with cross-entropy loss and Dice Loss, which fail to consistently improve IoU for the minority class. In contrast, Lovasz Hinge Loss, designed to directly optimize IoU, shows oscillations during the early stages of training in its loss polyline due to sharp gradient changes addressing IoU discrepancies in imbalanced regions. This reflects active optimization towards minority class predictions, progressively improving IoU. The IoU polyline for Lovasz Hinge Loss stabilizes in later stages shows a sustained upward trend,

Table 9
The results of the different loss functions.

	mIoU %	mAcc %	IoU_leak %	Acc_leak%
Cross Entropy Loss	82.79	85.81	66.70	71.86
Focal Loss	82.72	84.86	66.52	69.88
Dice Loss	81.44	83.95	64.06	68.09
Lovasz Hinge Loss (ours)	**92.38**	**94.50**	**85.31**	**89.22**

Table 8
Performance of Lovaz Hinge Loss on datasets with varying degrees of class imbalance.

	Num_leakage	Proportion	mIoU %	mAcc %	IoU_leak %	Acc_leak%
S3DIS_leakage	4,427	65:1	92.38	94.50	85.31	89.22
Test dataset 3	576	334:1	90.32	91.43	80.95	82.89

Note: Proportion represents the ratio between the number of points in the background and the leakage class.

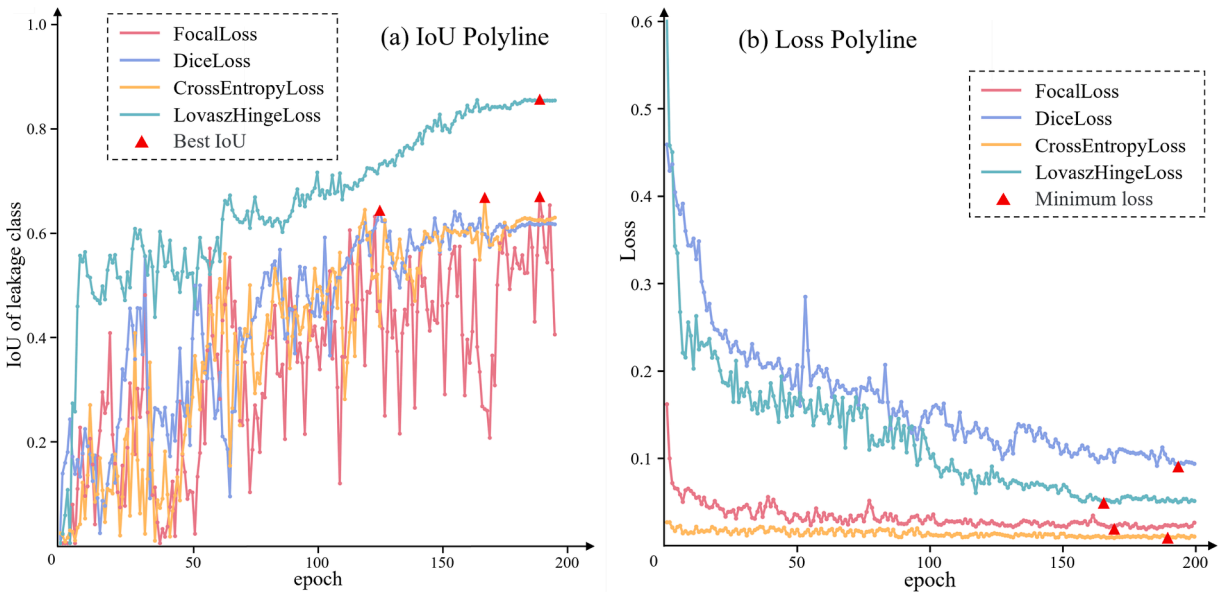


Fig. 19. Polylines of loss and IoU during training process.

eventually converging to a higher IoU value than other loss functions. This indicates that Lovasz Hinge Loss effectively resolves class imbalance, achieving superior segmentation performance.

5.5. Ablation study

As demonstrated in Table 10, the exclusion of the FPFH feature leads to a marked decline in the performance of GEPT-Net, evidenced by a decrease in mIoU by approximately 15.4 % and a reduction in mAcc by 12.3 %. This significant drop underscores the critical role of geometric feature aggregation in enhancing the accuracy of leakage detection. The integration of distinctive geometric features within typical tunnel leakage regions, when combined with point cloud intensity features, facilitates the precise localization of leakages. Additionally, the removal of the GECA block results in mIoU and mAcc reductions of 1.86 % and 1.32 %, respectively. Although this decrease is relatively modest, the enhanced interaction between geometric feature channels and intensity channels proves vital for the accurate identification of leakage areas characterized by both low-intensity and prominent FPFH features. Furthermore, the elimination of point cloud serialization resulted in mIoU and mAcc reductions of 13.1 % and 13.3 %, respectively, highlighting the necessity of serialization for the network's segmentation performance. The benefits of serialization can be attributed to its ability to incorporate a larger number of points into each patch processed by the network, without incurring additional computational costs, thereby significantly expanding the receptive field. This expanded receptive field is crucial for improving the representation capability of the network, directly contributing to performance enhancements.

Table 10
The results of the ablation study.

	FPFH feature	GECA block	Serialization	mIoU %	mAcc %
Model 1	×	✓	✓	76.98	82.23
Model 2	✓	×	✓	90.52	93.18
Model 3	✓	✓	×	79.24	84.17
Model 4 (full)	✓	✓	✓	**92.38**	**94.50**

Note: the "✓" indicates that the module is used in the network and the "×". indicates that the module is removed from the network.

5.6. Effective receptive field visualization

To intuitively reflect the receptive field of our method, we utilized gradients to visualize the Effective Receptive Field (ERF) (Lai et al., 2022; Luo et al., 2016) and compared it with the ERF of RandLA-Net, a well-established semantic segmentation method for large-scale scenes. As shown in Fig. 20, our network exhibits more extensive high-saliency regions, indicating a larger ERF. Unlike RandLA-Net, which relies on random point sampling and local feature aggregation, our method serializes the 3D point cloud data into a one-dimensional sequence. This approach allows us to process more data points per batch under the same computational constraints. Additionally, our serialization technique inherently aligns with the spatial distribution of tunnel point clouds in three-dimensional space, particularly the Z-order, making it more suitable for tunnel leakage detection. Benefits from a broader receptive field, our method achieves a mIoU that is 8.25 points higher than RandLA-Net, which highlights its superior accuracy and segmentation performance in the context of tunnel leakage detection.

5.7. Balance between model efficiency and accuracy

Balancing efficiency and accuracy in deep learning models is inherently challenging. Increasing accuracy often leads to higher computational demands, while enhancing efficiency can sometimes compromise precision. For our network, GEPT-Net, we have sought to achieve an optimal balance between these two factors, specifically for tunnel leakage detection. We assessed GEPT-Net's efficiency by comparing it with other state-of-the-art models, focusing on both latency and memory usage during training and inference. Latency refers to the time required for the network to process input and generate output, while memory usage indicates the computational resources needed to store and manage the model and data throughout these stages.

As shown in Table 11, models such as SpUnet and MinkUnet exhibit lower memory consumption during both training and inference. However, their accuracy does not meet the requirements for effective leakage detection. In contrast, GEPT-Net achieved the best performance in terms of latency, indicating its capacity to process data rapidly and efficiently, a benefit attributed to its point cloud serialization design. Although GEPT-Net may not have the smallest parameter set, it maintains relatively modest memory usage during both training and inference phases. This indicates that it can be deployed without extensive computational infrastructure, making it suitable for resource-constrained environments

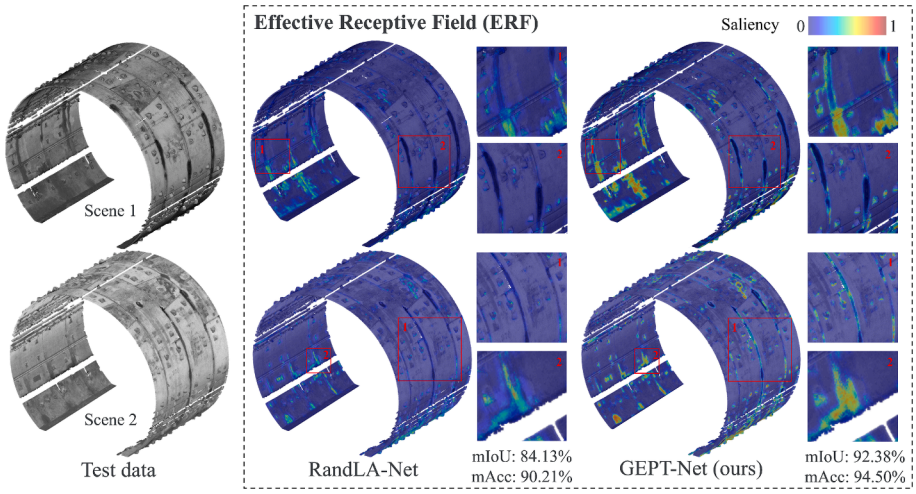


Fig. 20. Effective receptive field (ERF) visualization results. Saliency is obtained by performing a forward pass to get the target class probability, followed by a backward pass to compute gradients with respect to the input, highlighting the effective receptive field.

Table 11
Balance between model efficiency and accuracy.

Methods	Accuracy	Efficiency	Training		Inference	
	IoU_leak%	Params.	Latency	Memory	Latency	Memory
MinkU-Net	68.95	**37.86 M**	571 ms	4.95G	190 ms	4.53G
SpU-Net	68.85	39.15 M	459 ms	**1.93G**	218 ms	**2.42G**
PTv1	63.97	77.67 M	1402 ms	9.87G	1075 ms	9.85G
PTv2	75.17	39.08 M	469 ms	9.02G	233 ms	9.00G
GEPT-Net (ours)	**85.31**	46.17 M	**176 ms**	6.68G	**72 ms**	5.51G

in practical engineering applications. Additionally, as demonstrated in Table 11, GEPT-Net's inference time is highly competitive, approaching real-time processing speeds. This makes it a viable solution for real-world applications where rapid and accurate detection is crucial. In conclusion, GEPT-Net successfully balances accuracy and efficiency, offering strong real-time inference capabilities and low latency, making it well-suited for deployment in resource-constrained engineering environments.

6. Conclusion

In this research, an innovative semantic segmentation network GEPT-Net, meticulously crafted for the detection of tunnel leakages is proposed. Our research highlights that tunnel leakages predominantly manifest at specific geometric features such as segment joints, within-segment joints, and bolt holes, which are also characterized by lower intensity. GEPT-Net capitalizes on this observation by integrating these geometric characteristics into its architectural design. The network is trained and tested on our custom S3DIS_leakage dataset and achieves state-of-the-art performance. A pivotal advancement in our network is the incorporation of FPFH as geometric features, significantly enhancing the precision of leakage segmentation. Additionally, we propose the GECA block to synergize the interactions between FPFH geometric channels and intensity channels. This allows the network to accurately localize regions exhibiting lower intensity and prominent FPFH features, which are region of potential leakages. Furthermore, GEPT-Net also introduces a novel point cloud serialization method that substantially expands the receptive field using space-filling curves, thereby capturing richer feature information without the need for complex network modules. This approach not only improves accuracy but also maintains efficiency, achieving an optimal balance between precision and computational requirements. To address the severe class imbalance in our S3DIS_leakage dataset, we implement the Lovasz Hinge Loss function, which directly optimizes the IoU for each class. This approach ensures that even minority classes contribute to the gradient during optimization. Extensive experiments demonstrate the superior performance of GEPT-Net in the semantic segmentation of shield tunnels, surpassing existing cutting-edge methods by at least 11 % in IoU for leakage classes. Additionally, comparisons with 2D deep learning networks show a 12 % improvement in leakage detection accuracy, further validating the efficacy of our network design.

Besides, our comparative analysis of model parameters, training time, and inference time indicates that GEPT-Net effectively balances performance and efficiency. Even with an increased input dimensionality to 7, the efficiency of the GEPT-Net remains comparable to other methods, rendering it practical for real-world applications. Future work will explore extending GEPT-Net to full-feature segmentation of shield tunnels, as other tunnel components such as cables and grouting holes also exhibit distinctive geometric characteristics. Additionally, we aim to develop an end-to-end leakage semantic segmentation network that autonomously learns useful geometric features, thereby enhancing its applicability in engineering contexts. In terms of leakage analysis, while we have qualitatively validated the performance of our model using data from different tunnel environments, the quantitative relationship between specific conditions within these environment (such as dust levels and tunnel humidity, etc.) and tunnel leakage have not yet been thoroughly explored. In future work, we will aim to identify and analyze these potential quantitative relationships.

CRediT authorship contribution statement

Jundi Jiang: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Yueqian Shen:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis. **Jinhu Wang:** Writing –

review & editing, Supervision, Software, Formal analysis, Conceptualization. **Jinguo Wang:** Writing – review & editing, Supervision, Software, Formal analysis, Conceptualization. **Chenyang Zhang:** Data curation. **Jingyi Wang:** Data curation. **Vagner Ferreira:** Supervision, Resources, Project administration.

Declaration of competing interest

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

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant Nos. 41801379, 42201487, W2432026), the Open Found of Tunnel and Underground Engineering Research Center of Jiangsu Province (Grant No. 2023-SDJJ-01).

References

Armeni, I., Sax, S., Zamir, A.R., Savarese, S., 2017. Joint 2d-3d-semantic data for indoor scene understanding, arXiv preprint arXiv:1702.01105, https://doi.org/10.48550/arXiv.1702.01105.

Bao, Y., Wen, Y., Tang, C., Sun, Z., Meng, X., Zhang, D., Wang, L., 2024. Three-dimensional point cloud denoising for tunnel data by combining intensity and geometry information. Sustainability 16, 2077–2098. https://doi.org/10.3390/su16052077.

Berman, M., Triki, A.R., Blaschko, M.B., 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2018.00464.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: 2018 European Conference on Computer Vision (ECCV). https://doi.org/10.1007/978-3-030-01234-2_49.

Cheng, X., Hu, X., Tan, K., Wang, L., Yang, L., 2021. Automatic detection of shield tunnel leakages based on terrestrial mobile LiDAR intensity images using deep learning. IEEE Access 9, 55300–55310. https://doi.org/10.1109/access.2021.3070813.

Choy, C., Gwak, J., Savarese, S., 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/CVPR.2019.00319.

Cui, H., Ren, X., Mao, Q., Hu, Q., Wang, W., 2019. Shield subway tunnel deformation detection based on mobile laser scanning. Autom. Constr. 106, 102889–102901. https://doi.org/10.1016/j.autcon.2019.102889.

Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C., 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. Adv. Neural Inf. Proces. Syst. 35, 16344–16359. https://doi.org/10.48550/arXiv.2205.14135.

De Boer, P.-T., Kroese, D.P., Mannor, S., Rubinstein, R.Y., 2005. A tutorial on the cross-entropy method. Ann. Oper. Res. 134, 19–67. https://doi.org/10.1007/s10479-005-5724-z.

Du, L., Zhong, R., Sun, H., Pang, Y., Mo, Y., 2022. Dislocation detection of shield tunnel based on dense cross-sectional point clouds. IEEE Trans. Intell. Transp. Syst. 23, 22227–22243. https://doi.org/10.1109/tits.2022.3156385.

Feng, S.J., Feng, Y., Zhang, X.L., Chen, Y.H., 2023b. Deep learning with visual explanations for leakage defect segmentation of subway shield tunnel. Tunn. Undergr. Space Technol. 136, 105107–105125. https://doi.org/10.1016/j.tust.2023.105107.

Feng, H., Ma, L., Yu, Y., Chen, Y., Li, J., 2023a. SCL-GCN: stratified contrastive learning graph convolution network for pavement crack detection from mobile LiDAR point clouds. Int. J. Appl. Earth Obs. Geoinf. 118, 103248. https://doi.org/10.1016/j.jag.2023.103248.

Geng, P., Tan, Z., Luo, J., Wang, T., Li, F., Bei, J., 2023. ACPA-net: atrous channel pyramid attention network for segmentation of leakage in rail tunnel linings. Electronics 12, 255–277. https://doi.org/10.3390/electronics12020255.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R.R., Hu, S.-M., 2021. PCT: Point cloud transformer. Comput. Vis. Media 7, 187–199. https://doi.org/10.1007/s41095-021-0229-5.

Guo, Z., Wei, J., Sun, H., Zhong, R., Ji, C., 2024. Enhanced water leakage detection in shield tunnels based on laser scanning intensity images using RDES-net. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 17, 5680–5690. https://doi.org/10.1109/jstars.2024.3365535.

Hilbert, D., Hilbert, D., 1935. Über die stetige Abbildung einer Linie auf ein Flächenstück, Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes: Nebst Einer Lebensgeschichte, 1-2, https://link.springer.com/chapter/10.1007/978-3-662-38452-7_1.

Huang, H., Cheng, W., Zhou, M., Chen, J., Zhao, S., 2020. Towards automated 3D inspection of water leakages in shield tunnel linings using mobile laser scanning data. Sensors 20, 6669. https://doi.org/10.3390/s20226669.

Ji, A., Chew, A.W.Z., Xue, X., Zhang, L., 2022. An encoder-decoder deep learning method for multi-class object segmentation from 3D tunnel point clouds. Autom. Constr. 137, 104187. https://doi.org/10.1016/j.autcon.2022.104187.

Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics YOLO (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified transformer for 3d point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Li, J., Zhang, Z., Sun, H., Xie, S., Zou, J., Ji, C., Lu, Y., Ren, X., Wang, L., 2023. GL-Net: Semantic segmentation for point clouds of shield tunnel via global feature learning and local feature discriminative aggregation. ISPRS J. Photogramm. Remote Sens. 199, 335–349. https://doi.org/10.1016/j.isprsjprs.2023.04.011.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision. https://doi.org/10.1109/iccv.2017.324.

Lin, W., Li, P., Xie, X., 2022. A novel detection and assessment method for operational defects of pipe jacking tunnel based on 3D longitudinal deformation curve: a case study. Sensors (Basel) 22, 7648. https://doi.org/10.3390/s22197648.

Liu, X., Liu, X., Liu, Y.S., Han, Z., 2022b. SPU-Net: self-supervised point cloud upsampling by coarse-to-fine reconstruction with self-projection optimization. IEEE Trans Image Process 31, 4213–4226. https://doi.org/10.1109/TIP.2022.3182266.

Liu, S., Sun, H., Zhang, Z., Li, Y., Zhong, R., Li, J., Chen, S., 2022a. A multiscale deep feature for the instance segmentation of water leakages in tunnel using MLS point cloud intensity images. IEEE Trans. Geosci. Remote Sens. 60, 1–16. https://doi.org/10.1109/tgrs.2022.3158660.

Luo, W., Li, Y., Urtasun, R., Zemel, R., 2016. Understanding the effective receptive field in deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 29.

Milletari, F., Navab, N., Ahmadi, S.-A., 2016. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). https://doi.org/10.1109/3dv.2016.79.

Morton, G.M. 1966. A computer oriented geodetic data base and a new technique in file sequencing, https://www.scirp.org/reference/referencespapers?referenceid=1109616.

Protopapadakis, E., Doulamis, N., 2015. Image based approaches for tunnels' defects recognition via robotic inspectors. In: Advances in Visual Computing. https://doi.org/10.1007/978-3-319-27857-5_63.

Ronneberger, O., Fischer, P., Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In: 2015 Medical image computing and computer-assisted intervention–MICCAI. https://doi.org/10.1007/978-3-319-24574-4_28.

Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M., 2008a. Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. https://doi.org/10.1109/IROS.2008.4650967.

Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M., 2008b. Towards 3D Point cloud based object maps for household environments. Rob. Auton. Syst. 56, 927–941. https://doi.org/10.1016/j.robot.2008.08.005.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. https://doi.org/10.1109/ROBOT.2009.5152473.

Stałowska, P., Suchocki, C., Rutkowska, M., 2022. Crack detection in building walls based on geometric and radiometric point cloud information. Autom. Constr. 134, 104065. https://doi.org/10.1016/j.autcon.2021.104065.

Tan, K., Cheng, X., Ju, Q., Wu, S., 2016. Correction of mobile TLS intensity data for water leakage spots detection in subway tunnels. IEEE Geosci. Remote Sens. Lett. 13, 1711–1715. https://doi.org/10.1109/LGRS.2016.2605158.

Tombari, F., Salti, S., Di Stefano, L., 2010. Unique signatures of histograms for local surface description. In: 2010 European Conference on Computer Vision (ECCV). https://doi.org/10.1007/978-3-642-15558-1_26.

Wang, K., Zhang, Z., Wu, X., Zhang, L., 2022a. Multi-class object detection in tunnels from 3D point clouds: an auto-optimized lazy learning approach. Adv. Eng. Inf. 52, 101543. https://doi.org/10.1016/j.aei.2022.101543.

Wang, K., Wu, X., Li, H., Wang, F., Zhang, L., Chen, H., 2022b. Adaptively unsupervised seepage detection in tunnels from 3D point clouds. Struct. Infrastruct. Eng. 1, 1–19. https://doi.org/10.1080/15732479.2022.2136718.

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: 2024 IEEE/CVF Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.48550/arXiv.2312.10035.

Wu, C., Huang, H., Zhang, L., Chen, J., Tong, Y., Zhou, M., 2023. Towards automated 3D evaluation of water leakage on a tunnel face via improved GAN and self-attention DL model. Tunn. Undergr. Space Technol. 142, 105432. https://doi.org/10.1016/j.tust.2023.105432.

Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H., 2022. Point transformer v2: Grouped vector attention and partition-based pooling. Adv. Neural Inf. Proces. Syst. 35, 33330–33342. https://doi.org/10.48550/arXiv.2210.05666.

Xie, X., Tian, H., Zhou, B., Li, K., 2021. The life-cycle development and cause analysis of large diameter shield tunnel convergence in soft soil area. Tunn. Undergr. Space Technol. 107, 103680. https://doi.org/10.1016/j.tust.2020.103680.

Xu, T., Xu, L., Li, X., Yao, J., 2018. Detection of water leakage in underground tunnels using corrected intensity data and 3D point cloud of terrestrial laser scanning. IEEE Access 6, 32471–32480. https://doi.org/10.1109/access.2018.2842797.

Xue, Y., Li, Y., 2018. A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. Comput. Aided Civ. Inf. Eng. 33, 638–654. https://doi.org/10.1111/mice.12367.

Xue, Y., Jia, F., Cai, X., Shadabfar, M., Huang, H., 2021. An optimization strategy to improve the deep learning-based recognition model of leakage in shield tunnels. Comput. Aided Civ. Inf. Eng. 37, 386–402. https://doi.org/10.1111/mice.12731.

Xue, Y., Shi, P., Jia, F., Huang, H., 2022. 3D reconstruction and automatic leakage defect quantification of subway tunnel based on SfM-Deep learning method. Underground Space 7, 311–323. https://doi.org/10.1016/j.undsp.2021.08.004.

Yang, H., Xu, X., 2021. Structure monitoring and deformation analysis of tunnel structure. Compos. Struct. 276, 114565. https://doi.org/10.1016/j.compstruct.2021.114565.

Yuan, Y., Jiang, X., Liu, X., 2013. Predictive maintenance of shield tunnels. Tunn. Undergr. Space Technol. 38, 69–86. https://doi.org/10.1016/j.tust.2013.05.004.

Zhang, Z., Ji, A., Wang, K., Zhang, L., 2022. UnrollingNet: an attention-based deep learning approach for the segmentation of large-scale point clouds of tunnels. Autom. Constr. 142, 104456. https://doi.org/10.1016/j.autcon.2022.104456.

Zhang, Z., Ji, A., Zhang, L., Xu, Y., Zhou, Q., 2023. Deep learning for large-scale point cloud segmentation in tunnels considering causal inference. Autom. Constr. 152, 104915. https://doi.org/10.1016/j.autcon.2023.104915.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. In: 2017 IEEE Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.48550/arXiv.1612.01105.

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V., 2021. Point transformer. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). https://doi.org/10.48550/arXiv.2012.09164.

Zhao, S., Zhang, D.M., Huang, H.W., 2020. Deep learning–based image instance segmentation for moisture marks of shield tunnel lining. Tunn. Undergr. Space Technol. 95, 103156. https://doi.org/10.1016/j.tust.2019.103156.