applied sciences

MDPI

Article

Dual-Module Bench-Line Extraction and Surface-Object Segmentation from UAV LiDAR Point Clouds in Open-Pit Mines Using Neighborhood Geometric Analysis and an Enhanced PointNet++ Network

Shanfeng Ge [1,2], Nijia Qian [2,*], Jingxiang Gao [2], Xin Liu [2], Wenyuan Zhang [2], Yong Feng [2] and Dehu Yang [3]

[1] China Coal Pingshuo Group Co., Ltd., Shuozhou 036006, China; tb19160009b2@cumt.edu.cn or geshanfeng@126.com
[2] School of Environment and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China; jxgao@cumt.edu.cn (J.G.); xinliu@cumt.edu.cn (X.L.); zhangwy@cumt.edu.cn (W.Z.); yongfeng@cumt.edu.cn (Y.F.)
[3] Faculty of Geosciences and Engineering, Southwest Jiaotong University, Chengdu 610031, China; tigeryang@my.swjtu.edu.cn
* Correspondence: nijiaqian@cumt.edu.cn

Featured Application

The dual-module workflow provides complementary bench-line and surface-object information from UAV LiDAR point clouds. The geometry-based output can assist visual interpretation of bench morphology, while the semantic labels can support mine-map updating, haul-road mapping, and surface-object inventory. The case study demonstrates the feasibility of obtaining structural and semantic information from a common point-cloud dataset. Further reference-based evaluation and cross-mine testing will extend the applicability of the workflow to broader operational environments.

Abstract

Open-pit mines contain rapidly changing terrain, discontinuous bench structures, and mixed artificial–natural objects, which complicate automated three-dimensional mapping. This study presents a dual-module workflow for UAV LiDAR point clouds. Module A characterizes local geometry using normal and curvature descriptors, constructs local plane support through RANSAC fitting, and detects candidate bench-line points using an angular-gap criterion, followed by regional grouping and Kalman-filter refinement. Qualitative overlay with the orthophoto showed coherent correspondence with principal platform–slope transitions. Module B segments buildings, roads, and vegetation using a PointNet++ network enhanced by local Transformer self-attention and inverted residual feature transformation. Under a fixed spatial hold-out setting, the network achieved an overall accuracy of 97.6% and a mean intersection over union of 96.4%. It obtained the highest overall accuracy, mean intersection over union, and class-wise intersection over union among the selected baselines, whereas Point Transformer achieved a slightly higher mean class accuracy. The two independently operated modules provide complementary structural and semantic information for open-pit mine mapping. Broader applicability requires reference-based bench-line assessment and evaluation across additional mines and survey periods.

Keywords: 3D point-cloud processing; structural discontinuity detection; semantic segmentation; neighborhood geometry; local self-attention; inverted residual learning; digital mine mapping

Check for updates

Received: 11 July 2026
Revised: 6 August 2026
Accepted: 7 August 2026
Published: 8 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Open-pit mining is one of the most intensive forms of surface resource exploitation. During mining operations, stripping, excavation, dumping, road relocation, and slope cutting continuously reshape the terrain, producing a highly dynamic surface environment. The surface of an open-pit mine is usually composed of working benches, bench slopes, bench lines, haul roads, waste dumps, vegetation patches, buildings, and other artificial or natural objects. These elements are not only basic components of mine surveying and mapping but also important spatial indicators for production organization, slope inspection, deformation interpretation, and safety management. In particular, bench lines define the geometric transitions between platforms and slopes and are closely related to the three-dimensional morphology of mine slopes. Therefore, timely extraction of bench-line structures and typical surface objects is important for digital mine updating, intelligent inspection, and safety-oriented surface interpretation in open-pit mines [1,2].

Unmanned aerial vehicle light detection and ranging (UAV LiDAR) has become an effective tool for acquiring high-resolution three-dimensional information in complex mine environments. Compared with conventional ground surveying, UAV LiDAR can rapidly collect dense point clouds over large and difficult-to-access areas, reducing field workload and improving spatial coverage. Compared with image-based photogrammetry, LiDAR directly measures three-dimensional geometry and is less dependent on image texture and illumination, particularly in steep or near-vertical slope areas. The resulting point clouds can provide detailed descriptions of bench morphology, road surfaces, buildings, vegetation, and local terrain changes. These advantages make UAV LiDAR particularly suitable for open-pit mines, where terrain surfaces are irregular, object boundaries are fragmented, and repeated surveys are required for dynamic updating and change detection [1]. The interpretation of three-dimensional point clouds depends not only on data quality but also on the definition of local neighborhoods and the representation of geometric characteristics. Weinmann et al. [3] demonstrated that point-specific neighborhood selection can improve the distinctiveness of local geometric features and subsequent semantic interpretation. Hackel et al. [4] further showed that multi-scale neighborhood features are particularly important for processing unstructured point clouds with strongly varying sampling density. These findings are relevant to UAV LiDAR point clouds of open-pit mines, where terrain relief, scanning geometry, and local occlusion commonly produce substantial variations in point spacing and neighborhood completeness.

Feature-line extraction from three-dimensional point clouds has long been an important topic in photogrammetry, remote sensing, and computer vision. Many geometry-based methods detect feature points or lines by analyzing local surface variation, normal-vector discontinuity, curvature, residuals from plane fitting, or angular gaps within spatial neighborhoods. Ni et al. [5] proposed the analysis of geometric properties of neighborhood method for edge detection and feature-line tracing in three-dimensional point clouds, combining neighborhood geometric analysis, RANSAC-based plane fitting, and an angular-gap criterion. Such explicit geometric descriptors are interpretable and suitable for detecting sharp structural transitions. In open-pit mine applications, Wang et al. proposed an automatic step-line extraction method using neighborhood geometry, curvature weighting, and moving least-squares fitting [6]. Mao et al. developed an automatic method for extracting step-feature lines from UAV point-cloud data, emphasizing their relevance to mine-status mapping and stripping planning [7]. Cui et al. subsequently proposed a structural-characteristic-based method for large-scale open-pit mine point clouds and constructed a dedicated step-feature-line dataset [8]. These studies demonstrate the practical demand for automated bench-line extraction. Nevertheless, geometry-based responses may become fragmented or ambiguous under uneven point density, local occlusion, rough

surfaces, incomplete boundaries, and intersections between benches, roads, and other surface objects.

In parallel with geometry-based analysis, deep learning has substantially advanced point-cloud semantic segmentation. Unlike traditional handcrafted-feature methods, deep neural networks can learn hierarchical representations directly from point coordinates and additional attributes such as color or intensity. Guo et al. provided a comprehensive review of deep-learning methods for three-dimensional point-cloud understanding [9]. PointNet was among the first networks to process unordered point sets directly by using shared multilayer perceptrons and symmetric aggregation [10]. Although simple and effective, PointNet mainly relies on global feature aggregation and has limited ability to represent fine local structures. PointNet++ addressed this limitation by introducing hierarchical sampling, neighborhood grouping, and local feature learning in metric space, making it more suitable for non-uniform point clouds and scenes containing local geometric variation [11].

Subsequent methods have explored different strategies for improving point-cloud representation. DGCNN dynamically constructs neighborhood graphs in feature space and uses EdgeConv to describe relationships between central points and their neighbors [12]. PointCNN learns an X-transformation to weight and rearrange local points before convolution-like feature aggregation [13]. PointConv approximates continuous three-dimensional convolution while accounting for non-uniform sampling density [14], whereas KPConv defines learnable kernel points for flexible and deformable convolution in point clouds [15]. RandLA-Net combines random sampling with local feature aggregation to process large-scale point clouds [16]. Superpoint Graphs organize geometrically homogeneous regions into graph structures [17], while Minkowski convolutional networks employ sparse convolutions for scalable three-dimensional and spatiotemporal perception [18].

Attention-based and surface-aware networks have further strengthened contextual and geometric representation. Point Transformer introduces self-attention into point-cloud processing to model local contextual relationships and long-range dependencies [19]. RepSurf explicitly represents local surfaces using geometric priors [20]. PointNeXt revisits PointNet++ through improved training and scaling strategies and introduces an inverted residual bottleneck and separable multilayer perceptrons into the PointNet++ hierarchy [21]. More recently, Sevinc and Karas developed a Transformer-enhanced PointNet++ architecture for airborne LiDAR semantic segmentation using multi-scale feature fusion [22]. Point Transformer V3 simplifies neighborhood interaction and expands the effective receptive field through serialized point-cloud mapping [23]. Other studies, including SCF-Net [24], CGA-Net [25], and bilateral augmentation with adaptive fusion [26], have emphasized spatial context, category-dependent neighborhood aggregation, and boundary-aware feature fusion. These studies provide the methodological basis for incorporating local self-attention and residual feature transformation into hierarchical point-cloud networks. In this study, these components are configured and evaluated specifically for surface-object segmentation in an open-pit mine environment.

Public benchmarks have also contributed to the development and evaluation of outdoor point-cloud segmentation. DALES provides a large-scale aerial LiDAR dataset for semantic segmentation [27]. The Hessigheim 3D benchmark contains high-density UAV LiDAR point clouds and textured meshes with detailed semantic annotations [28]. Toronto-3D provides mobile LiDAR data for large-scale urban-roadway segmentation [29], while SensatUrban includes nearly three billion annotated points from extensive urban scenes [30]. In addition to sensing and scene variability, stable UAV operation under outdoor disturbances is relevant to the reliability of airborne data acquisition. Recent quadrotor studies have examined backstepping–sliding control and disturbance-observer strategies for maintaining flight stability under external perturbations [31,32]. The benchmark datasets described

above also reveal common difficulties associated with outdoor LiDAR data, including density variation, class imbalance, fragmented boundaries, labeling uncertainty, and domain differences. However, most public datasets represent urban, roadway, or village environments rather than active open-pit mines, where extreme topographic relief and continuous engineering disturbance create distinct geometric and semantic conditions.

Despite the progress in both explicit geometric processing and deep semantic learning, the direct application of existing methods to open-pit mine point clouds remains challenging. First, open-pit mine point clouds contain large elevation differences, irregular bench structures, steep slopes, and mixed artificial–natural surface objects. The same semantic class may exhibit different geometric forms in different production areas, while different classes may share similar local characteristics. For example, haul roads and working platforms can both appear as relatively smooth, low-curvature surfaces, whereas vegetation boundaries may overlap with slope edges or road shoulders. Second, UAV flight altitude, scanning angle, slope orientation, and occlusion can produce considerable variations in point density and local completeness. Third, bench-line extraction and surface-object segmentation correspond to different information requirements. Bench lines are structural discontinuities that can be characterized through local geometric changes, whereas buildings, roads, and vegetation require object-level semantic interpretation. Geometry-based methods do not provide complete semantic labels, while semantic segmentation networks do not necessarily preserve explicit bench-line structures.

Existing studies generally treat bench-line extraction and point-cloud semantic segmentation as separate tasks. The present study does not claim the first use of self-attention or inverted residual learning in a PointNet++-based network. Instead, its methodological contribution lies in a task-oriented configuration for open-pit UAV LiDAR data. First, an explicit geometry branch is constructed to preserve platform–slope discontinuities as line structures. Second, local self-attention and inverted residual feature transformation are embedded at corresponding stages of the PointNet++ hierarchy to improve the representation of mixed road, building, and vegetation boundaries. Third, the two independently operated branches are organized within a common spatially controlled workflow so that structural and semantic information can be obtained from the same preprocessed point cloud without implying end-to-end feature fusion.

The contributions of this study are threefold. First, a dual-module workflow is established to generate complementary bench-line and surface-object information from a common UAV LiDAR dataset. Second, a multi-stage neighborhood geometry-based procedure is applied to bench-line delineation through multi-scale neighborhood analysis, robust local plane fitting, angular-gap detection, spatial grouping, and recursive line refinement. Third, a task-specific PointNet++ configuration is evaluated by introducing local self-attention and inverted residual feature transformation at multiple hierarchical stages, with comparative and ablation experiments used to examine the resulting segmentation behavior.

The remainder of this paper is organized as follows. Section 2 presents the methodology, including the overall dual-module workflow, neighborhood geometry-based bench-line extraction, the enhanced PointNet++ network, and the evaluation protocols. Section 3 describes the case-study area, UAV LiDAR acquisition, auxiliary photogrammetric products, point-cloud preprocessing, semantic annotation, spatial data partition, and training configuration. Section 4 presents the qualitative bench-line demonstration and quantitative semantic-segmentation results. Section 5 discusses geometric rationale, network behavior, complementary application scope, and study limitations.

2. Methodology

2.1. Overview of the Dual-Module Workflow

The proposed workflow is designed to obtain the following two complementary types of information from UAV LiDAR point clouds of open-pit mines: structural bench-line information and point-wise semantic labels of typical surface objects. As illustrated in Figure 1, the workflow consists of point-cloud preprocessing, neighborhood geometry-based bench-line extraction, enhanced PointNet++-based semantic segmentation, and task-specific evaluation.

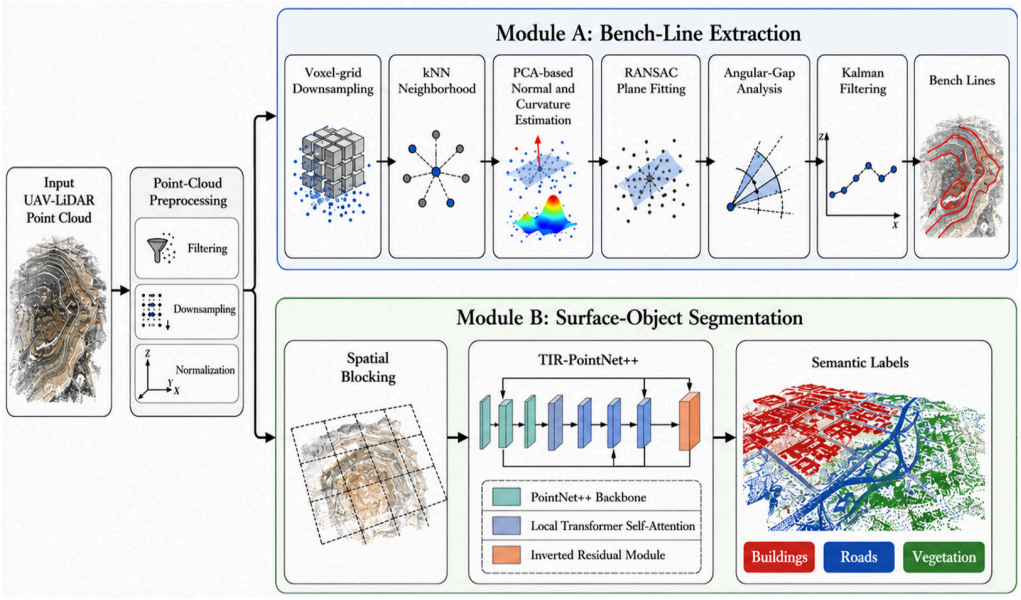


Figure 1. Overall workflow of the dual-module method. The preprocessed UAV LiDAR point cloud is independently processed by the geometry-based bench-line branch and the TIR-PointNet++ semantic-segmentation branch to generate structural line features and point-wise surface-object labels, respectively.

The raw UAV LiDAR point cloud is first subjected to outlier removal, statistical filtering, downsampling, coordinate normalization, and spatial organization. The detailed preprocessing operations, coordinate reference system, and case-study settings are described in Section 3. The resulting point cloud is then used as the common input for two independent processing branches.

Module A extracts bench-line structures by analyzing the local geometry of point-cloud neighborhoods. For each target point, a local neighborhood is constructed to estimate the normal direction, curvature-related surface variation, and residual from local plane fitting. Candidate structural points are identified using complementary geometric evidence, including RANSAC-based plane fitting and an angular-gap criterion. Non-maximum suppression, connectivity constraints, and terrain-related conditions are subsequently used to reduce redundant and isolated responses. The retained candidate points are grouped into spatially connected sequences and refined using a Kalman-filter-based line-fitting procedure.

Module B performs point-wise semantic segmentation of buildings, roads, and vegetation. The network retains hierarchical sampling, grouping, feature abstraction, and feature propagation structure of PointNet++ [11]. A local Transformer self-attention component and an inverted residual feature-transformation component are incorporated into the local feature-encoding process. The resulting configuration is referred to as TIR-PointNet++, where "TIR" denotes the use of Transformer self-attention and inverted residual transformation.

The two branches operate independently, and their outputs are evaluated separately according to their respective task characteristics.

2.2. Neighborhood Geometry-Based Bench-Line Extraction

Bench lines in open-pit mines generally occur near the intersections between working platforms and slope faces. In local point-cloud neighborhoods, these transition zones commonly exhibit changes in normal direction, enhanced surface variation, relatively large local plane-fitting residuals, and asymmetric angular distributions of neighboring points. Based on these properties, Module A consists of downsampling, local neighborhood construction, normal and curvature estimation, RANSAC-based local plane fitting, angular-gap analysis, candidate-point filtering, regional grouping, and Kalman-filter-based line refinement. The workflow is shown in Figure 2.

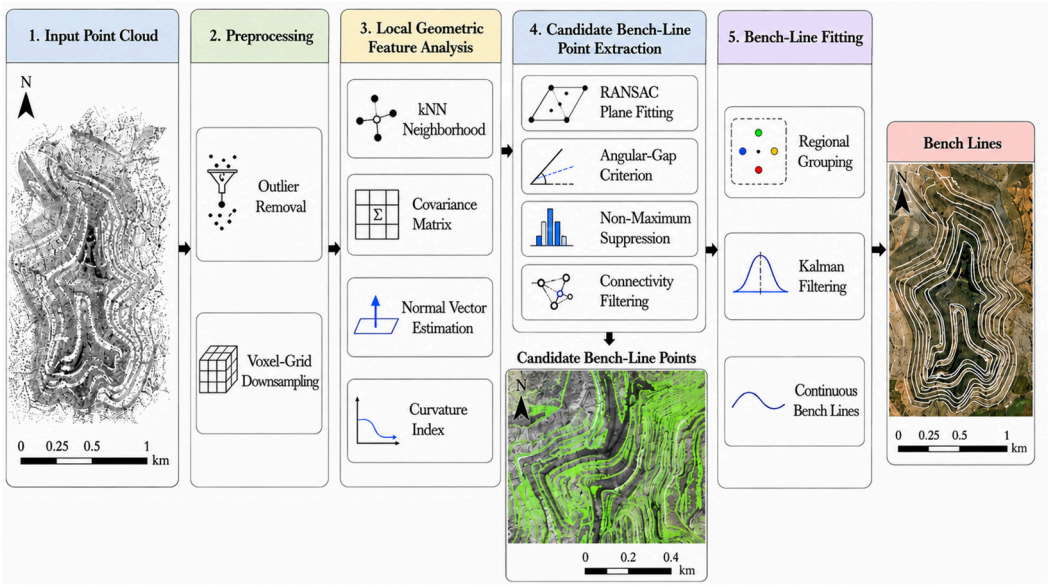


Figure 2. Workflow of neighborhood geometry-based bench-line extraction, including point-cloud downsampling, local neighborhood construction, normal and curvature estimation, RANSAC-based local plane fitting, angular-gap analysis, candidate-point filtering, regional grouping, and Kalman-filter-based line refinement.

2.2.1. Point-Cloud Downsampling and Local Neighborhood Construction

The point cloud is first cleaned and voxel-grid downsampled to reduce isolated noise and redundant sampling while retaining bench-scale surface transitions. Two k-nearest-neighbor supports are then constructed for each target point. A compact neighborhood containing k_g points is used for covariance-based normal and curvature estimation, whereas a larger neighborhood containing k_r points is used for RANSAC plane fitting and angular-gap analysis. The voxel size and the two neighborhood sizes are fixed for the complete case-study area; their values and selection basis are reported in Section 3.3.2.

For each target point $\boldsymbol{p}_i$, a local neighborhood is constructed using its k_g nearest neighbors. Let $\boldsymbol{p}_j$ denote the coordinate vector of the j-th neighboring point and $\overline{p}$ denote the centroid of the neighborhood. The covariance matrix of the neighborhood is calculated as follows:

$$\boldsymbol{C} = \frac{1}{k_g}\sum_{j=1}^{k_g}(\boldsymbol{p}_j - \overline{\boldsymbol{p}})(\boldsymbol{p}_j - \overline{\boldsymbol{p}})^{\mathrm{T}} \tag{1}$$

where $\boldsymbol{C}$ is the 3×3 covariance matrix and k_g is the number of neighboring points used for normal and curvature estimation. Eigenvalue decomposition of $\boldsymbol{C}$ gives three eigenvalues

$\lambda_1 \geq \lambda_2 \geq \lambda_3$ and their corresponding eigenvectors. The eigenvector associated with the smallest eigenvalue λ_3 is used as the estimated local normal vector because it represents the direction of minimum geometric variation in the neighborhood. The curvature index is calculated as follows:

$$V = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (2)$$

where V quantifies the degree of local surface variation. It is retained for morphological characterization but is not used as an independent candidate-point threshold in the implemented decision sequence.

2.2.2. Local Plane Fitting and Angular-Gap Analysis

Local plane fitting is conducted using the random sample consensus (RANSAC) procedure adopted in the neighborhood-geometry framework of Ni et al. [5]. RANSAC estimates a plane model from randomly sampled subsets and evaluates the remaining points according to their fitting residuals. It is therefore suitable for local neighborhoods that may contain noise or points that do not follow a single plane. For a local plane expressed as $ax + by + cz + d = 0$, the perpendicular distance between a neighborhood point and the fitted plane is used as the local fitting residual. Relatively large residuals may indicate a transition between two surface orientations. However, large residuals may also occur in vegetation, rough ground, sparse regions, or noisy observations. Plane-fitting residuals are therefore not used alone to define bench-line points. For each target point, RANSAC is applied to the neighborhood containing k_r points. A plane hypothesis is generated from the minimum three-point sample, and the model with the largest consensus set is retained after the prescribed number of iterations. Neighboring points whose perpendicular distances to the fitted plane do not exceed the inlier-distance threshold are regarded as inliers. No additional fixed minimum-inlier threshold is imposed beyond the three-point model requirement and maximum-consensus selection. The neighborhood size, iteration limit, and inlier-distance threshold are listed in Section 3.3.2.

To further characterize local structural discontinuity, an angular-gap criterion is introduced following the neighborhood geometric-property analysis described in [5]. For a target point $\boldsymbol{p}$, vectors from $\boldsymbol{p}$ to its neighboring points are projected onto the estimated local tangent plane. The projected vectors are sorted according to their angular directions θ_i. The maximum angular gap is defined as follows:

$$g_\theta = \max_{i=1,\ldots,k_r} (\theta_{i+1} - \theta_i), \quad \theta_{k_r+1} = \theta_1 + 2\pi \quad (3)$$

A large g_θ indicates an asymmetric distribution of neighboring points, which commonly occurs near incomplete surface boundaries and platform–slope transitions. The curvature index and post-fit residual are retained as descriptive measures of local surface variation, but neither is subjected to a separate candidate-point threshold. The 0.5 m RANSAC inlier-distance threshold is used only to estimate the local support plane. Candidate points are selected explicitly by the angular-gap condition $g_\theta \geq \tau_\theta$, with $\tau_\theta = 120^\circ$ in this case study; the same threshold is applied throughout the study area.

Candidate responses are thinned using non-maximum suppression within a fixed neighborhood radius. Responses that do not satisfy the minimum local-slope and local height-difference requirements are subsequently removed, thereby suppressing responses on relatively flat roads and working platforms. The retained candidate points are then passed to the regional-grouping procedure described in Section 2.2.3. The corresponding thresholds are reported in Section 3.3.2.

2.2.3. Regional Grouping and Kalman-Filter-Based Line Refinement

After candidate-point filtering, two points are connected when their horizontal distance and the difference between their dominant local directions do not exceed the corresponding thresholds. Connected components containing fewer than the prescribed number of points or having a projected length shorter than the minimum line length are removed. Within each retained component, the points are ordered along the principal line direction and treated as pseudo-sequential spatial observations. The index k therefore represents spatial order along a candidate bench-line sequence rather than physical time. The grouping thresholds are listed in Section 3.3.2.

A Kalman filter is then used as a recursive line-state estimator [33]. The prediction step is written as follows:

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} \tag{4}$$

$$\boldsymbol{P}_k^- = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q} \tag{5}$$

where $\hat{\boldsymbol{x}}_k^-$ and $\boldsymbol{P}_k^-$ are the predicted state and covariance matrix at step k, respectively; $\boldsymbol{A}$ is the state transition matrix; and $\boldsymbol{Q}$ is the process-noise covariance matrix. Kalman refinement is performed in the projected horizontal plane. Elevation is retained for terrain-related filtering and three-dimensional visualization but is not included in the recursive line state. The state vector is defined as follows:

$$\boldsymbol{x}_k = \left[x_k,\ y_k,\ v_{x,k},\ v_{y,k}\right]^{\mathrm{T}} \tag{6}$$

where x_k and y_k are the projected coordinates and $v_{x,k}$ and $v_{y,k}$ represent the local coordinate changes along the ordered point sequence. Because the point index denotes spatial order rather than time, a unit spatial step is adopted. The state-transition matrix is as follows:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The update step is given by the following:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{H}\boldsymbol{P}_k^-\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1} \tag{8}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-\right) \tag{9}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_k^- \tag{10}$$

where $\boldsymbol{K}_k$ is the Kalman gain, $\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ is the observation matrix, $\boldsymbol{z}_k = \begin{bmatrix} x_k^{obs} & y_k^{obs} \end{bmatrix}^{\mathrm{T}}$ is the observation vector, $\boldsymbol{R}$ is the measurement-noise covariance matrix, and $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{P}_k$ are the updated state and covariance matrix, respectively. For each ordered sequence, the initial position is obtained from the first candidate point, and the initial directional components are estimated from the coordinate difference between the first two points. The initial, process-noise, and measurement-noise covariance settings are reported in Section 3.3.2.

The recursive prediction–update process smooths short-range fluctuations in the ordered candidate-point sequences and provides a continuous estimate across limited local interruptions. Kalman filtering is used here only as a line-refinement procedure; its relative performance against spline, moving-least-squares, and other line-fitting methods is not evaluated in the present study.

2.3. Enhanced PointNet++ for Surface-Object Segmentation

Module B performs point-wise semantic segmentation of buildings, roads, and vegetation. PointNet++ is retained as the backbone because its hierarchical set-abstraction and feature-propagation structure is suitable for learning local features from irregular and non-uniform point sets [11].

The original PointNet++ semantic-segmentation framework follows an encoder–decoder organization. In the encoder, set abstraction (SA) modules perform farthest-point sampling, neighborhood grouping, and local feature encoding. Shared multilayer perceptrons map point coordinates and attributes into higher-dimensional feature spaces, and symmetric aggregation produces neighborhood descriptors. Multiple SA modules progressively enlarge the receptive field and encode point-cloud information at different spatial scales.

In the decoder, feature propagation (FP) modules interpolate high-level features back to denser point sets. The interpolated features are concatenated with corresponding lower-level features through skip connections and are subsequently refined using shared feature transformations. The final point-wise classifier produces a semantic probability vector for each point.

To strengthen neighborhood-context and channel-wise feature representation, the evaluated TIR-PointNet++ configuration incorporates local Transformer self-attention and inverted residual feature transformation into the PointNet++ hierarchy. The two components are embedded in the three set-abstraction stages, the bottleneck stage, and the corresponding feature-propagation refinement stages. For an input containing N points, the encoder follows the point-number hierarchy $N \rightarrow N/4 \rightarrow N/16 \rightarrow N/64$, with feature dimensions of 64, 128, 256, and 512, respectively. In the present experiment, N = 4096, corresponding to 4096, 1024, 256, and 64 points. The decoder restores point resolution through three feature-propagation stages and skip connections, as illustrated in Figure 3.

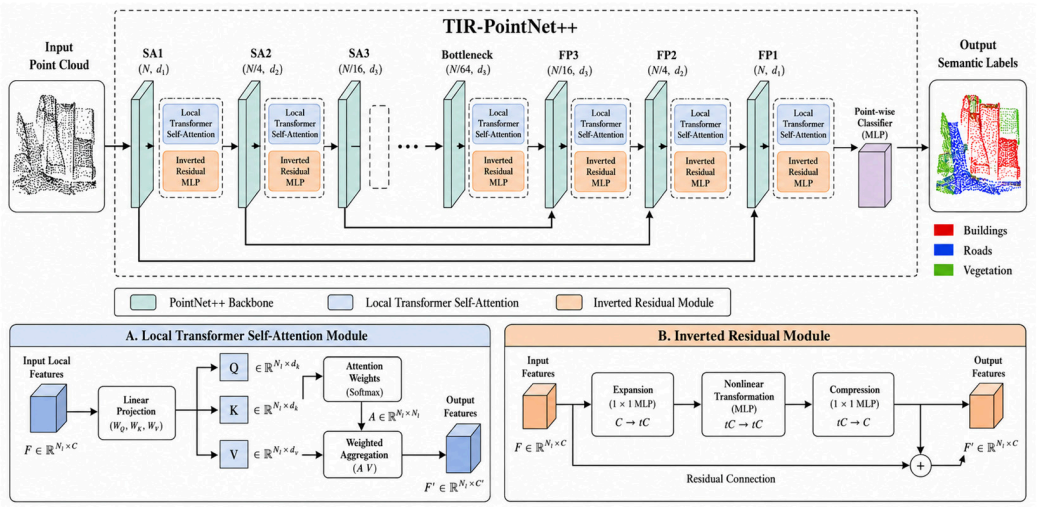


Figure 3. Architecture of TIR-PointNet++, in which local Transformer self-attention and inverted residual blocks are embedded within the hierarchical encoder–decoder structure of PointNet++. The corresponding point numbers, feature dimensions, neighborhood size, and expansion factor are summarized in Section 3.4.2.

2.3.1. Local Transformer Self-Attention

The self-attention component is embedded into the local neighborhood feature-encoding stage before neighborhood aggregation. It is not implemented as a fully global Transformer operating on the complete mine point cloud. Instead, it models contextual relationships among points within each local neighborhood.

Given a local feature matrix $\boldsymbol{F}$, query, key, and value matrices are obtained through learnable linear transformations:

$$\boldsymbol{Q} = \boldsymbol{F}\boldsymbol{W}_Q, \boldsymbol{K} = \boldsymbol{F}\boldsymbol{W}_K, \boldsymbol{V} = \boldsymbol{F}\boldsymbol{W}_V \tag{11}$$

where $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$ are learnable weight matrices.

The self-attention output is calculated as follows:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}\right)\boldsymbol{V} \tag{12}$$

where d is the feature dimension of the corresponding hierarchy stage. Local attention is calculated within a 32-point neighborhood using a single attention head. The query, key, and value dimensions match the feature width of the corresponding stage, namely 64, 128, 256, or 512.

The attention weights allow neighboring points to contribute differently to the encoded local feature. This configuration is intended to improve contextual interpretation in ambiguous neighborhoods, including road–vegetation and building–vegetation transition zones. The use of self-attention follows the general Transformer principle [34] and established point-cloud attention models such as Point Transformer [19]. The local self-attention block is evaluated as a task-specific component of the mine-oriented PointNet++ configuration.

2.3.2. Inverted Residual Feature Transformation

An inverted residual feature-transformation component is incorporated into the local encoding path without changing the sampling and neighborhood-grouping procedures of PointNet++. The component follows an expansion–transformation–compression structure inspired by inverted residual bottlenecks [35].

First, input features are projected from the original channel dimension into a higher-dimensional feature space. Nonlinear transformations are then applied within the expanded representation. Finally, the feature channels are projected back to the required output dimension. A residual connection is used when the input and output dimensions are compatible. For an input feature dimension C, the block expands the feature channels to $4C$, performs nonlinear transformation in the expanded space, and then compresses the channels back to C. The expansion factor is therefore set to $t = 4$. A residual connection is applied when the input and output dimensions are identical.

The inverted residual block follows the expansion–transformation–compression design of MobileNetV2 [35] and is consistent with its subsequent use in PointNeXt [21]. In the present network, it is incorporated into the PointNet++ hierarchy as an additional channel-transformation path. Its contribution is evaluated together with local self-attention through the ablation experiments reported in Section 4.4.

2.3.3. Point-Wise Classification Objective

After hierarchical encoding and feature propagation, the final point-wise features are passed through shared classification layers and a softmax function. The cross-entropy loss is used as the classification objective:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log(\hat{y}_{ic}) \tag{13}$$

where N is the number of valid labeled points in a batch, $C = 3$ represents buildings, roads, and vegetation, y_{ic} is the one-hot ground-truth label, and $\hat{y}_{ic}$ is the predicted probability. Points belonging to other surface types are assigned an ignore label and are excluded from

both loss calculation and metric evaluation. Unweighted cross-entropy loss is used, and no class-frequency weighting is applied. A dropout rate of 0.5 is applied before the final point-wise classification layer.

2.4. Evaluation Protocols and Metrics

The two modules are evaluated using task-specific criteria corresponding to their different output forms.

2.4.1. Qualitative Evaluation of Bench-Line Extraction

Bench-line extraction is evaluated through spatial overlay with the UAV orthophoto and visible platform–slope morphology. The assessment examines whether the extracted lines are concentrated near principal platform–slope transitions, follow the dominant orientations of the mine benches, avoid extensive responses on clearly flat surfaces, and remain spatially coherent in locally incomplete regions. Accordingly, the evaluation focuses on morphological correspondence and spatial consistency rather than line-level positional accuracy. The orthophoto is used only as an auxiliary visual reference and is not treated as an independently interpreted bench-line ground truth. Therefore, completeness, correctness, F1-score, average perpendicular distance, and Hausdorff distance are not calculated in the present evaluation.

2.4.2. Quantitative Evaluation of Semantic Segmentation

Surface-object segmentation is evaluated using overall accuracy (OA), mean class accuracy (mAcc), class-wise intersection over union (IoU), and mean intersection over union (mIoU).

Overall accuracy represents the proportion of correctly classified points among all test points:

$$OA = \frac{\sum_{c=1}^{C} TP_c}{N} \tag{14}$$

where TP_c is the number of correctly classified points belonging to class c, C is the number of classes, and N is the total number of test points.

The accuracy of class c is calculated as follows:

$$Acc_c = \frac{TP_c}{TP_c + FN_c} \tag{15}$$

and the mean class accuracy is as follows:

$$mAcc = \frac{1}{C}\sum_{c=1}^{C} Acc_c \tag{16}$$

where FN_c is the number of ground-truth points of class c that are incorrectly assigned to other classes.

The intersection over union for class c is defined as follows:

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c} \tag{17}$$

and the mean intersection over union is as follows:

$$mIoU = \frac{1}{C}\sum_{c=1}^{C} IoU_c \tag{18}$$

where FP_c is the number of points from other classes that are incorrectly predicted as class c. All metrics are calculated using the fixed spatial hold-out test set described in Section 3.4.

3. Case Study Data and Experimental Setup

3.1. Study Area and Data Overview

The study area is located in the Pingshuo Mining Area, Shanxi Province, China, with the East Open-Pit Mine selected as the principal case-study site. The Pingshuo Mining Area extends approximately from 112°17′11″ E to 112°27′54″ E and from 39°25′12″ N to 39°36′11″ N, with an average elevation of approximately 1300 m. It is situated in the northern Shanxi Loess Plateau, where the terrain is dominated by loess hills, gullies, and mining-induced landforms. The surface is characterized by sparse vegetation, strong terrain fragmentation, and substantial anthropogenic modification caused by long-term open-pit mining. The location of the study area and the principal UAV survey products are shown in Figure 4.

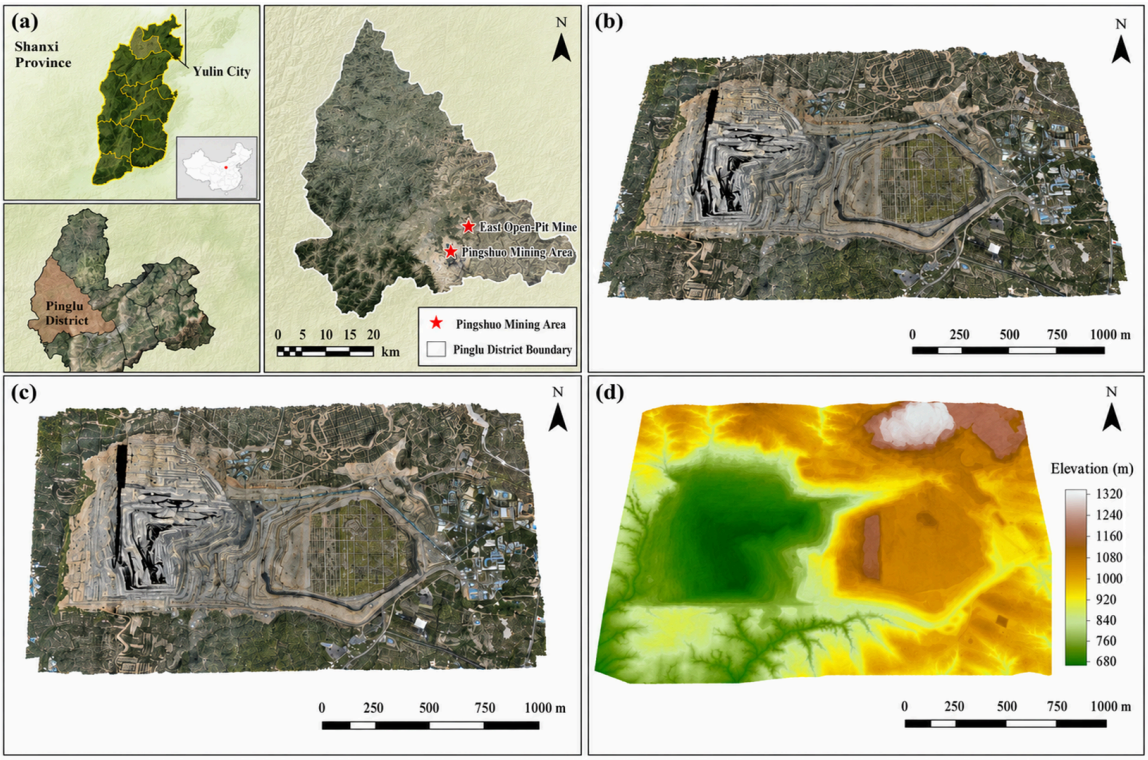


Figure 4. Location and UAV survey data of the study area. (**a**) Location of the Pingshuo Mining Area and the East Open-Pit Mine; (**b**) UAV LiDAR point cloud; (**c**) UAV orthophoto; and (**d**) UAV-derived digital elevation model.

The East Open-Pit Mine provides a representative engineering environment for evaluating UAV LiDAR-based surface-feature processing. Its surface contains working benches, bench slopes, haul roads, buildings, vegetation patches, dumping areas, and locally fragmented terrain boundaries. These elements differ substantially in geometric form, surface roughness, elevation variation, and spatial continuity. Bench lines are generally distributed along the transition zones between working platforms and inclined slope faces, where local surface orientation and curvature may change abruptly. Roads and working platforms tend to exhibit relatively smooth and continuous surfaces, whereas vegetation and buildings commonly present irregular or clustered point-cloud structures.

The principal UAV LiDAR dataset used for the bench-line and semantic-segmentation experiments was acquired on 22 July 2024. The dataset included the active mine and repre-

sentative surrounding areas containing buildings, roads, and vegetation. The LiDAR point cloud constituted the primary three-dimensional dataset used by both processing modules. Orthophotos, digital terrain products, CAD data, and field-surveyed coordinates were used as auxiliary information for interpretation, annotation, visualization, and geometric quality assessment.

3.2. UAV LiDAR Acquisition and Auxiliary Photogrammetric Products

3.2.1. UAV Platform and LiDAR Mapping System

The UAV LiDAR data were acquired using a DJI Matrice 350 RTK UAV platform (SZ DJI Technology Co., Ltd., Shenzhen, China) equipped with an AA9 laser aerial survey system (Shanghai Huace Navigation Technology Ltd., Shanghai, China). The UAV platform has a maximum payload of 2.7 kg, a maximum take-off weight of 9 kg, and a maximum flight time of 55 min under standard conditions. Its nominal RTK positioning accuracy is 1 cm + 1 ppm in the horizontal direction and 1.5 cm + 1 ppm in the vertical direction.

The AA9 laser aerial survey system integrates a surveying-grade LiDAR scanner, a positioning and orientation system, and a mapping camera. The nominal system accuracy is 5 cm in planimetry and 5 cm in elevation. The LiDAR scanner has a ranging distance of 10–600 m, a maximum pulse emission rate of 500,000 points/s, a scan-line rate of 50–250 lines/s, a field of view of 75°, and support for up to six returns. The ranging accuracy and repeatability are 15 mm and 5 mm, respectively, at a distance of 150 m. The integrated camera has a resolution of 26 megapixels, a focal length of 16 mm, and a minimum image interval of 1 s. The principal specifications are summarized in Table 1.

Table 1. Main specifications of the UAV platform and LiDAR mapping system.

Component	Parameter	Specification
UAV platform	Platform type	DJI Matrice 350 RTK
	Maximum payload	2.7 kg
	Maximum take-off weight	9 kg
	Nominal RTK positioning accuracy	1 cm + 1 ppm horizontal; 1.5 cm + 1 ppm vertical
	Maximum flight time	55 min
LiDAR system	System type	AA9 laser aerial survey system
	Nominal system accuracy	5 cm horizontal; 5 cm vertical
	Ranging distance	10–600 m
	Maximum pulse emission rate	500,000 points/s
	Scan-line rate	50–250 lines/s
	Ranging accuracy/repeatability	15 mm/5 mm at 150 m
	Field of view	75°
	Number of returns	Up to 6
Camera	Image resolution	26 megapixels
	Focal length	16 mm
	Minimum image interval	1 s

The values in Table 1 are manufacturer-specified system characteristics and describe the nominal acquisition platform. They are not used as substitutes for the independently assessed geometric quality of the point-cloud product.

3.2.2. Relationship Between LiDAR and Photogrammetric Processing

The LiDAR point cloud was the primary data source for bench-line extraction and point-wise semantic segmentation. The camera integrated into the aerial survey system simultaneously acquired optical images. These images were processed photogrammetrically to generate auxiliary orthophoto and terrain products.

During the photogrammetric workflow, 45 horizontal and vertical ground control points and 13 photogrammetric check points were manually selected from the 1:500 topographic map of the East Open-Pit Mine. A total of 15 flight strips and 531 images were included in the aerial triangulation. Camera calibration information, image exposure positions, automatically matched tie points, and the 45 ground-control points were used in the block adjustment, whereas the 13 photogrammetric check points were withheld from the adjustment. The adjusted mapping-camera imagery was also used to assign RGB values to the LiDAR points during point-cloud colorization, after which the orthophoto and terrain products were generated. Accordingly, Module A used LiDAR geometry only, whereas Module B used XYZ coordinates together with image-derived RGB attributes. The orthophoto and terrain products served as auxiliary sources for semantic annotation, visual inspection, result presentation, and qualitative bench-line comparison.

3.2.3. Independent Geometric Quality Assessment

Two checkpoint sets were used for different quality-control purposes. The 13 photogrammetric check points were used to evaluate the aerial-triangulation results, whereas 10 independently RTK-surveyed validation points were used to assess the geometric quality of the final UAV LiDAR point-cloud product.

After completion of the UAV survey, 10 validation points with identifiable surface characteristics and a relatively uniform spatial distribution were selected across the mine area. The coordinates of these points were identified in the UAV point cloud and independently measured using real-time kinematic positioning. The point-cloud coordinates were then compared with the RTK measurements.

The resulting root-mean-square errors were 8.9 cm in planimetry and 9.4 cm in elevation. These values characterize the overall geometric quality of the source point-cloud product and provide a positional-quality reference for the subsequent processing.

3.3. Point-Cloud Preprocessing, Coordinate Reference, and Annotation

3.3.1. Coordinate Reference System

All spatial products were referenced to CGCS2000 using a 3-degree Gauss–Krüger projection with a central meridian of 111° E, and elevations were referenced to the National Height Datum 1985. Coordinate normalization was applied before geometric feature calculation and network training, while the original georeferenced coordinates were retained for result visualization and engineering interpretation.

3.3.2. Point-Cloud Preprocessing

The complete preprocessing and bench-line extraction settings are summarized in Table 2. The original UAV LiDAR dataset contained more than 40 million points, with an average density of approximately 25 points/m^2. Obvious outliers and isolated observations were removed using statistical filtering with 20 neighboring points and a standard-deviation multiplier of 2.0. Voxel-grid downsampling with a voxel size of 0.5 m reduced the geometry-processing point cloud to approximately 4 million points while retaining the principal bench-scale surface transitions.

Table 2. Preprocessing and bench-line extraction parameters used in the case study.

Stage	Parameter	Setting
Statistical filtering	Number of neighbors	20
Statistical filtering	Standard-deviation multiplier	2.0
Voxel downsampling	Voxel size	0.5 m
Normal/curvature estimation	k_g	30 nearest neighbors
RANSAC and angular-gap support	k_r	200 nearest neighbors
RANSAC	Minimum sample size	3 points
RANSAC	Maximum iterations	1000
RANSAC	Inlier-distance threshold	0.5 m
RANSAC	Additional minimum-inlier threshold	Not separately imposed
Angular-gap analysis	τ_θ	120°
Non-maximum suppression	Neighborhood radius	1.0 m
Terrain filtering	Minimum local slope	15°
Terrain filtering	Minimum local height difference	1.0 m
Regional grouping	Maximum connection distance	2.0 m
Regional grouping	Maximum direction difference	20°
Regional grouping	Minimum connected points	15
Line filtering	Minimum projected line length	5 m
Kalman filter	State vector	$[x, y, v_x, v_y]^T$
Kalman filter	Initial covariance	$\boldsymbol{P}_0 = \boldsymbol{I}_4$
Kalman filter	Process covariance	$\boldsymbol{Q} = 0.01\boldsymbol{I}_4$
Kalman filter	Measurement covariance	$\boldsymbol{R} = 0.1\boldsymbol{I}_2$

For semantic segmentation, each point was represented by six attributes, namely X, Y, Z, R, G, and B. The spatial coordinates were transformed into the local block reference and normalized before network input, whereas RGB values were divided by 255 and scaled to [0, 1]. The original georeferenced coordinates were retained for spatial overlay, result visualization, and engineering interpretation.

Point-cloud filtering and geometric processing were implemented using Python 3.7 and Open3D 0.15.2 [36]. ArcGIS Pro 3.3.1 was used for spatial overlay and map-based inspection, whereas CloudCompare 2.13.2 was used for point-cloud visualization, inspection, and annotation support.

At an average density of approximately 25 points/m^2, the nominal horizontal point spacing was approximately 0.2 m. The 0.5 m voxel size was selected to reduce redundant sampling without removing the principal platform–slope transitions. The 30-point neighborhood retained local sensitivity for covariance estimation, whereas the 200-point neighborhood provided more stable support for robust plane fitting. The remaining thresholds were determined with reference to the voxel scale and the characteristic morphology of the mine benches. All settings were fixed for the complete study area and were not adjusted according to individual result regions.

3.3.3. Semantic Categories and Manual Annotation

The semantic-segmentation task included the following three classes: buildings, roads, and vegetation. Buildings mainly comprised industrial and auxiliary structures around the mine. The road class included haul roads and other traffic surfaces associated with mine operations. Vegetation included trees, shrubs, and other plant-covered areas.

These classes were selected because they represent common surface objects relevant to mine mapping, transportation interpretation, ecological-restoration assessment, and surface-condition analysis. These three classes do not constitute a complete semantic inventory of an open-pit mine. Points belonging to working platforms, slope faces, exposed rock, waste dumps, coal piles, water bodies, and mining equipment were assigned an

ignore label and were not merged into any of the three target classes. Ignored points were excluded from loss calculation and metric evaluation. The semantic dataset contained approximately 4.0 million valid labeled points. Buildings accounted for approximately 1.2 million points (30%), roads for approximately 0.8 million points (20%), and vegetation for approximately 2.0 million points (50%). These values refer to unique labeled points before overlapping block generation.

Manual annotation was conducted after the point cloud had been organized into spatial areas. CAD vector data and UAV orthophotos were used as auxiliary interpretation references. ArcGIS Pro 3.3.1 and CloudCompare 2.13.2 were used jointly for spatial inspection and point-level labeling.

Boundary-decision and hierarchical-priority rules were established to improve annotation consistency. In overlapping or ambiguous regions, the dominant surface object was assigned as the semantic category. Sparse or partially occluded areas were interpreted using neighboring point geometry and orthophoto information. Regions for which the category remained uncertain were reviewed before the labels were finalized. Representative annotated samples are shown in Figure 5. The principal characteristics of the dataset and annotation process are summarized in Table 3.

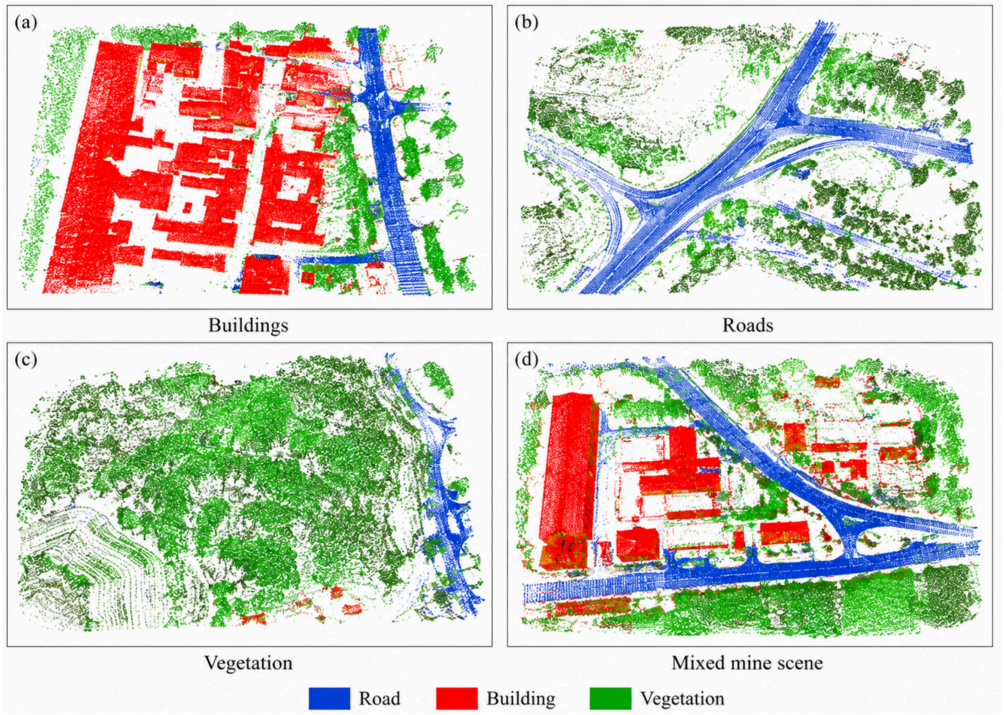


Figure 5. Examples of annotated UAV LiDAR point clouds used for surface-object segmentation. (**a**) Buildings; (**b**) roads; (**c**) vegetation; and (**d**) a mixed mine scene containing buildings, roads, and vegetation.

3.4. Spatial Data Partition and Training Configuration

3.4.1. Spatial Data Partition and Augmentation

The five representative spatial areas served as data-management units for annotation and inspection. For experimental partitioning, the combined labeled footprint was repartitioned into six spatially disjoint portions. One predetermined portion was used as the independent test set, whereas the remaining five portions formed the training and development data. The development data were further divided spatially into training and validation subsets. Thus, the five annotation areas and the six experimental portions represent different organizational levels.

Table 3. UAV LiDAR dataset and semantic-annotation configuration.

Item	Description
Case-study site	East Open-Pit Mine and representative surrounding areas
Principal acquisition date	22 July 2024
Primary data source	UAV LiDAR point clouds
Original point count	More than 40 million points
Average point density	Approximately 25 points/m^2
Geometry-processing points	Approximately 4 million points after voxel downsampling
Valid semantic points	Approximately 4.0 million points
Building points	Approximately 1.2 million points (30%)
Road points	Approximately 0.8 million points (20%)
Vegetation points	Approximately 2.0 million points (50%)
Other surface types	Assigned an ignore label and excluded from loss and metrics
Point-block size	20 m × 20 m
Sliding-window stride	10 m
Points per block	4096
Total valid blocks	3600
Training blocks	2500
Validation blocks	500
Test blocks	600
Auxiliary products	UAV orthophoto, terrain products, CAD vector data, and RTK validation coordinates
Point attributes	X, Y, Z coordinates and RGB information
Horizontal reference	CGCS2000
Map projection	3-degree Gauss–Krüger projection, central meridian 111° E
Vertical reference	National Height Datum 1985
Main preprocessing	Outlier removal, statistical filtering, voxel-grid downsampling, and attribute normalization
Data-management units	Five representative spatial areas
Experimental partitions	Six spatially disjoint portions
Semantic categories	Buildings, roads, vegetation
Annotation references	CAD vector data and UAV orthophotos
Annotation software	ArcGIS Pro 3.3.1 and CloudCompare 2.13.2

Sliding-window blocking was performed separately within the training, validation, and test portions using a block size of 20 m × 20 m and a stride of 10 m, corresponding to 50% overlap. No block was allowed to cross a subset boundary. Each block contained 4096 points. Blocks containing more than 4096 valid points were randomly subsampled, whereas blocks with fewer than 4096 valid points were up-sampled to 4096 points by uniform random sampling with replacement from the valid points in the same block. Sampling was point-wise rather than class-conditioned; therefore, it preserved the original block-level class proportions in expectation, although stochastic duplication could slightly alter the realized proportions within an individual sparse block, particularly near class boundaries. This procedure produced 3600 valid blocks, including 2500 training blocks, 500 validation blocks, and 600 test blocks.

Data augmentation was applied only to the training blocks. Each block was randomly rotated around the vertical axis within 0–360°, isotropically scaled by a factor sampled from 0.9 to 1.1 and translated by a value sampled from −0.1 to 0.1 along each normalized coordinate axis. Clipped Gaussian coordinate jitter with σ = 0.01 and a maximum absolute value of 0.05 was then applied. The normalized RGB attributes were perturbed within ±0.05 and clipped to [0, 1]. No augmentation was applied to the validation or test blocks.

During test inference, three voting passes were performed. For points occurring in multiple overlapping blocks or voting passes, the predicted class probabilities were averaged, and the class with the highest mean probability was assigned as the final label.

3.4.2. Training Configuration

The complete point-block, network, software, and training settings are summarized in Table 4. The semantic-segmentation models were implemented using Python 3.7, PyTorch 1.10.1 [37], and CUDA 11.3 under Ubuntu 20.04 LTS. Training was conducted on an NVIDIA GeForce RTX 3090 GPU with 24 GB of memory (NVIDIA Corporation, Santa Clara, CA, USA). AdamW was used with an initial learning rate of 1×10^{-3} and a weight decay of 1×10^{-4} [38]. Each model was trained for 100 epochs with a batch size of eight. The learning rate was adjusted using cosine annealing with $T_{\max} = 100$ and a minimum learning rate of 1×10^{-5} [39]. The checkpoint with the highest validation-set mIoU was used for evaluation on the fixed test set.

Table 4. Point-block, network, software, and training settings used for semantic segmentation.

Group	Item	Setting
Input	Block size	20 m × 20 m
Input	Sliding-window stride	10 m
Input	Points per block	4096
Input	Point attributes	XYZ + RGB
Input	Ignored points	Excluded from loss and metrics
Network	Point hierarchy	4096 → 1024 → 256 → 64
Network	Feature dimensions	64 → 128 → 256 → 512
Network	Local neighborhood	32 points
Network	Module placement	Three SA stages, bottleneck, and corresponding FP stages
Transformer	Attention heads	1
Inverted residual	Expansion factor	4
Classifier	Dropout rate	0.5
Loss	Objective	Unweighted cross-entropy
Optimizer	AdamW learning rate	1×10^{-3}
Optimizer	Weight decay	1×10^{-4}
Schedule	Cosine annealing	$T_{\max} = 100$, $\eta_{\min} = 1 \times 10^{-5}$
Training	Batch size	8
Training	Epochs	100
Model selection	Checkpoint criterion	Highest validation mIoU
Test inference	Voting	Three passes with probability averaging
Operating system	Version	Ubuntu 20.04 LTS
Programming language	Version	Python 3.7
Framework	Version	PyTorch 1.10.1
GPU	Hardware	NVIDIA GeForce RTX 3090, 24 GB
CUDA	Version	11.3

All baseline models used the same spatial partition, point blocks, 4096-point inputs, XYZ + RGB attributes, data-augmentation ranges, optimizer, learning-rate schedule, batch size, and number of training epochs. Architecture-specific feature operators were retained according to the corresponding baseline design. The values reported in Sections 4.2 and 4.4 correspond to one predefined fixed spatial hold-out test set; they are not averages over six rotating folds or repeated random initializations.

4. Results

4.1. Qualitative Demonstration of Bench-Line Extraction

The neighborhood geometry-based method was applied to the UAV LiDAR point cloud of the East Open-Pit Mine to identify candidate bench-line points and generate refined line structures. Figure 6a shows the candidate points overlaid on the UAV orthophoto, whereas Figure 6b presents the refined bench lines in the same spatial reference.



Figure 6. Qualitative bench-line extraction results in the East Open-Pit Mine. (**a**) Candidate bench-line points overlaid on the UAV orthophoto and (**b**) refined bench lines overlaid on the orthophoto. The overlays illustrate the spatial correspondence between the extracted structures and visible platform–slope morphology.

The candidate points are mainly distributed near visible platform–slope transitions and slope-break zones. In these areas, the local relationship between relatively flat platforms and inclined slope faces changes abruptly. The spatial distribution of the candidate points is therefore qualitatively consistent with the expected geometric characteristics of bench lines.

The candidate-point map also contains locally scattered responses, particularly in areas with rough terrain, incomplete point-cloud coverage, irregular surface morphology, or complex intersections between bench structures and haul roads. After non-maximum suppression, connectivity filtering, and terrain-related constraints, the retained points show clearer directional organization along several principal bench transitions. Nevertheless, the visual results also indicate that the geometric criteria do not eliminate all fragmented or ambiguous responses.

Kalman-filter-based line refinement was subsequently applied to the spatially grouped candidate-point sequences. Compared with the original candidate-point distribution, the refined lines appear visually smoother and show fewer small local fluctuations. Several locally separated candidate-point groups are represented as more continuous line segments after recursive refinement. The fitted lines generally follow the main orientation of the visible benches and are predominantly located near platform–slope transition zones.

Figure 6 demonstrates qualitative spatial correspondence between the extracted lines and several principal platform–slope transitions. The results are therefore interpreted as an assessment of morphological and spatial consistency rather than a line-level geometric-accuracy evaluation.

4.2. Quantitative Semantic Segmentation Results

The TIR-PointNet++ configuration was evaluated against the following four representative point-cloud segmentation methods: PointNet [10], PointNet++ [11], RepSurf [20], and Point Transformer [19]. The experiment was designed as a controlled representative comparison rather than an exhaustive benchmark of all recently published point-cloud networks. DGCNN, RandLA-Net, and Point Transformer V3 are discussed in the literature review to describe broader methodological developments but are not included in the present experiment. Accordingly, all comparative rankings in this study are restricted to the selected baseline set. All methods were retrained on the present dataset using the same fixed spatial partition, input attributes, point-block configuration, augmentation procedure, optimizer, learning-rate schedule, batch size, and number of training epochs. The quantitative results are summarized in Table 5.

Table 5. Quantitative comparison of the selected point-cloud segmentation methods on the UAV LiDAR dataset.

Method	OA (%)	mAcc (%)	mIoU (%)	Building IoU (%)	Road IoU (%)	Vegetation IoU (%)
PointNet [10]	92.4	79.5	74.0	92.1	43.6	86.3
PointNet++ [11]	96.3	92.8	90.1	94.7	82.2	93.4
RepSurf [20]	96.9	96.4	90.5	96.0	81.1	94.5
Point Transformer [19]	96.2	97.6	92.7	97.0	85.7	95.5
TIR-PointNet++	97.6	97.2	96.4	98.6	94.8	95.9

Under the reported experimental setting, TIR-PointNet++ achieved the highest OA and mIoU among the evaluated methods, reaching 97.6% and 96.4%, respectively. It also obtained the highest IoU for all three semantic categories. Compared with the original PointNet++, OA increased by 1.3 percentage points and mIoU increased by 6.3 percentage points.

TIR-PointNet++ did not achieve the highest value for every metric. Point Transformer obtained an mAcc of 97.6%, which was 0.4 percentage points higher than the 97.2% obtained by TIR-PointNet++. Because mAcc averages class-specific recall, whereas mIoU penalizes both false-positive and false-negative assignments, the results indicate that Point Transformer achieved slightly higher average class recall, while TIR-PointNet++ achieved higher overall point accuracy and class-overlap quality under the evaluated setting.

For the building class, TIR-PointNet++ achieved an IoU of 98.6%. Buildings in the evaluated areas generally have relatively distinct geometric boundaries and spatially clustered point distributions, which may facilitate their separation from surrounding objects. Point Transformer achieved the second-highest building IoU of 97.0%, followed by RepSurf at 96.0%.

Road segmentation produced larger differences among the evaluated methods. PointNet obtained a road IoU of 43.6%, substantially below its building and vegetation results. PointNet++ increased the road IoU to 82.2%, while RepSurf and Point Transformer achieved 81.1% and 85.7%, respectively. TIR-PointNet++ obtained a road IoU of 94.8%. Roads in the study area are frequently connected to working platforms and exposed surfaces and may also border vegetation. Their local geometry can therefore overlap with that of adjacent objects, making the road category more difficult than the building category.

For vegetation, the five methods achieved IoU values ranging from 86.3% to 95.9%. TIR-PointNet++ obtained the highest value of 95.9%, followed closely by Point Transformer at 95.5%. The relatively small difference between these two methods indicates that

both attention-based contextual modeling and the evaluated PointNet++ configuration represented the irregular and clustered vegetation structures effectively in this dataset.

PointNet showed the lowest OA, mAcc, and mIoU. This result is consistent with its limited explicit representation of hierarchical local neighborhoods. PointNet++ substantially improved all three overall metrics by introducing hierarchical sampling and local feature aggregation. RepSurf achieved a higher OA and mAcc than PointNet++ but only a small increase in mIoU. Point Transformer further increased mAcc and mIoU, demonstrating the value of contextual feature interaction.

Overall, TIR-PointNet++ produced the highest OA, mIoU, and class-wise IoU values among the selected representative baselines, while Point Transformer retained a slight advantage in mAcc.

4.3. Visual Comparison

Visual comparisons were conducted using representative UAV LiDAR scenes from the East Open-Pit Mine. Figure 7 presents the ground-truth labels and the segmentation results obtained using PointNet++, Point Transformer, and TIR-PointNet++. The black rectangles identify representative boundary areas, and the enlarged views in the lower row provide a closer examination of local classification differences.

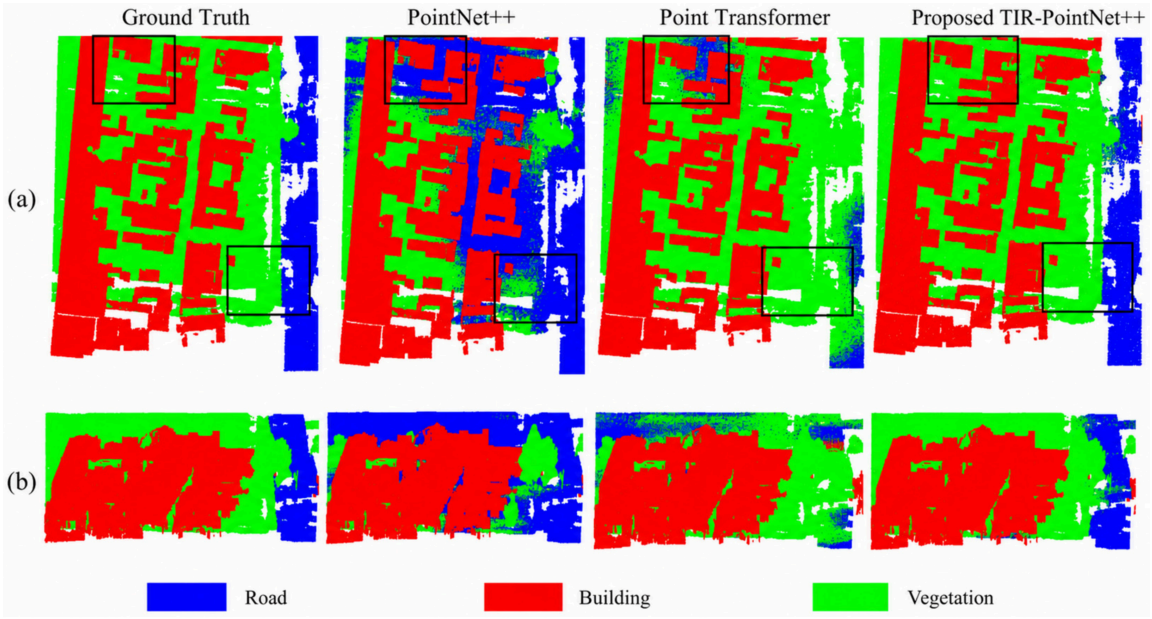


Figure 7. Visual comparison of surface-object segmentation results. (**a**) Full-scene ground-truth labels and segmentation results obtained using PointNet++, Point Transformer, and TIR-PointNet++, with black rectangles identifying representative mixed-object boundary regions; (**b**) enlarged views of the regions marked by the black rectangles in subfigure (**a**). Blue, red, and green denote roads, buildings, and vegetation, respectively.

PointNet++ identifies the main spatial distributions of buildings, roads, and vegetation. However, visible misclassifications remain in several vegetation–road transition regions. Some road points are assigned to vegetation, and parts of irregular vegetation boundaries extend into adjacent road surfaces. These patterns are consistent with the lower road IoU reported for PointNet++ in Table 5.

Point Transformer provides visually improved boundary representation relative to PointNet++ in several areas. Nevertheless, scattered misclassified points remain near building–vegetation interfaces and mixed structural regions. Its visual results are consistent with its relatively high mAcc and mIoU, although its road IoU remains below that of TIR-PointNet++.

TIR-PointNet++ produces visually more compact road regions and fewer obvious cross-class assignments in the highlighted road–vegetation and building–vegetation boundary areas. In the enlarged views, several boundaries appear more continuous and closer to the ground-truth labels than those produced by PointNet++ and Point Transformer.

The visual patterns are consistent with the numerical differences in Table 5, particularly the higher road IoU obtained by TIR-PointNet++.

4.4. Ablation Study

An ablation experiment was conducted to examine the individual and combined contributions of local Transformer self-attention and inverted residual feature transformation. The original PointNet++ network was used as the baseline, and all configurations used the same spatial partition, optimization strategy, training epochs, and hyperparameter settings. All ablation configurations were trained and evaluated using the same fixed spatial partition and checkpoint-selection criterion described in Section 3.4.2. The results are summarized in Table 6.

Table 6. Ablation results for the evaluated TIR-PointNet++ configuration.

Model	Transformer Component	Inverted Residual Component	OA (%)	mIoU (%)
PointNet++ baseline	No	No	96.3	90.1
PointNet++ + T	Yes	No	96.8	94.5
PointNet++ + IR	No	Yes	96.5	94.1
TIR-PointNet++	Yes	Yes	97.6	96.4

Note: T denotes local Transformer self-attention; IR denotes inverted residual feature transformation.

The PointNet++ baseline achieved an OA of 96.3% and an mIoU of 90.1%. When the local Transformer self-attention component was added alone, OA increased to 96.8% and mIoU increased to 94.5%. These values represent numerical increases of 0.5 and 4.4 percentage points, respectively.

When the inverted residual component was added without the Transformer component, OA reached 96.5% and mIoU reached 94.1%, corresponding to increases of 0.2 and 4.0 percentage points over the baseline. The comparatively small change in OA and larger change in mIoU indicate that the improvement was more evident in the average class-overlap metric than in the overall proportion of correctly classified points.

The complete TIR-PointNet++ configuration achieved an OA of 97.6% and an mIoU of 96.4%, corresponding to increases of 1.3 and 6.3 percentage points over the PointNet++ baseline. It also outperformed both single-component variants. These results indicate that local self-attention and inverted residual feature transformation provided complementary improvements under the evaluated configuration.

5. Discussion

5.1. Geometric Basis and Performance Characteristics of Bench-Line Extraction

Bench lines in open-pit mines are structural discontinuities formed near the intersections between relatively flat working platforms and inclined slope faces. This geometric characteristic distinguishes them from general object outlines and many texture-dependent surface boundaries. Within a local point-cloud neighborhood, a platform–slope transition may produce a change in normal direction, an increased curvature-related response, a larger local plane-fitting residual, and an uneven angular distribution of neighboring points. These properties provide the geometric rationale for using neighborhood analysis to identify candidate bench-line points.

The use of multiple local descriptors is intended to reduce dependence on a single elevation or slope attribute. Elevation alone may not distinguish bench transitions when roads cross different elevation levels, whereas a slope-based response can also occur on rough ground, dumping surfaces, road shoulders, and other non-bench structures. Normal estimation and RANSAC plane fitting define the local surface frame used for angular-gap analysis, while curvature and fitting residuals describe the degree of local surface variation. In the implemented decision sequence, the angular-gap threshold is the explicit candidate-point criterion, followed by slope, height-difference, non-maximum-suppression, and connectivity constraints.

The geometric settings were selected according to the relationship between point spacing, voxel scale, and bench morphology. At an average density of approximately 25 points/m^2, the nominal horizontal point spacing was approximately 0.2 m. The 0.5 m voxel size reduced redundant sampling while retaining principal bench-scale transitions; the 30-point neighborhood preserved local sensitivity for covariance analysis, whereas the 200-point neighborhood increased the stability of RANSAC plane fitting. The 1.0 m NMS radius and 2.0 m connectivity distance were defined relative to the voxel scale, while the slope, height-difference, direction, and minimum-length constraints suppressed flat or fragmented responses. The 120° angular-gap threshold identified strongly asymmetric neighborhoods in the present dataset, but these settings should be verified when point density, acquisition geometry, or bench morphology changes substantially.

Kalman-filter-based refinement recursively smooths the ordered candidate-point sequences. As shown in Figure 6, the refined lines contain fewer short-scale fluctuations and appear more continuous than the initial candidate-point distribution. This behavior is consistent with the prediction–update mechanism of the filter, although comparison with spline, moving-least-squares, and other line-reconstruction strategies is still required to determine its relative performance.

Several conditions may produce missed detections or false responses in Module A. Sparse sampling and occlusion near steep slopes can weaken normal and curvature estimates. Rough dumping surfaces and road shoulders can produce geometric discontinuities resembling bench transitions. Haul roads crossing slope faces may interrupt the continuity of actual bench lines, while vegetation and equipment can introduce non-terrain points into the local neighborhoods. Closely spaced or highly eroded benches may also be difficult to distinguish after downsampling. These cases explain why the candidate-point map still contains fragmented and ambiguous responses.

The present evaluation focuses on the spatial correspondence between the extracted lines and visible platform–slope morphology. Figure 6 indicates that the method captures several principal bench transitions while retaining some fragmented responses in complex areas. A reference-based evaluation using completeness, correctness, positional distance, and line-continuity metrics would provide a more comprehensive assessment in future studies.

5.2. Interpretation of the Semantic Segmentation Performance

The semantic-segmentation results indicate that the evaluated TIR-PointNet++ configuration achieved the highest OA, mIoU, and class-wise IoU values among the selected baseline methods. Its OA and mIoU reached 97.6% and 96.4%, respectively. Compared with the original PointNet++ configuration, these values increased by 1.3 and 6.3 percentage points. The following discussion explains the observed results in relation to the network design, while recognizing that the experiments were conducted using one spatial hold-out configuration.

PointNet++ provides the hierarchical backbone of the network. Its set-abstraction modules organize irregular points into local neighborhoods and progressively learn features at different spatial scales [11]. This structure is more appropriate for local geometric variation than the predominantly global aggregation used in PointNet. The substantial difference between PointNet and PointNet++ in Table 5 is consistent with the importance of hierarchical neighborhood representation for the evaluated outdoor point cloud.

However, the original PointNet++ configuration may have difficulty distinguishing classes with similar local geometry. Haul roads and working platforms can both contain smooth, low-curvature surfaces. Vegetation may overlap spatially with road shoulders, slope boundaries, or building margins, particularly where point density is uneven. In such neighborhoods, class decisions cannot always be made from the geometry of an individual point or a simple pooled local descriptor.

The local self-attention component assigns different weights to neighboring features according to their learned relationships. Its purpose is to strengthen contextual interaction within each grouped neighborhood rather than to apply global attention to the complete mine point cloud. This configuration may help distinguish a road point surrounded by other road-surface points from a geometrically similar point located at a vegetation boundary. Similar motivations underlie Point Transformer and other attention-based point-cloud networks [19]. The ablation result shows that adding the local attention component increased mIoU from 90.1% to 94.5%. This numerical change is consistent with improved neighborhood-context representation.

The inverted residual component introduces an expansion–transformation–compression path into the local feature encoder. Such structures are established network-design mechanisms and have also been used in PointNeXt to enhance PointNet++-based feature learning [21]. In the present experiment, adding the inverted residual component alone increased mIoU from 90.1% to 94.1%. This result suggests that additional channel transformation was useful for the evaluated dataset, but it does not demonstrate that the component is universally preferable to other residual or multilayer-perceptron designs.

When both components were included, mIoU reached 96.4%, which exceeded the values obtained using either component alone. The numerical pattern is consistent with complementary roles as follows: local attention modifies how neighborhood information is weighted, whereas inverted residual transformation modifies how the aggregated features are represented across channels. The ablation pattern suggests that local attention primarily improves neighborhood interaction, whereas inverted residual transformation enhances channel-wise feature representation.

The largest class-wise difference between TIR-PointNet++ and the selected baselines occurred for roads. Road IoU increased from 82.2% for PointNet++ and 85.7% for Point Transformer to 94.8% for TIR-PointNet++. The visual comparison similarly shows fewer vegetation–road assignments in the highlighted regions. These results suggest that the evaluated configuration was effective in mixed road-boundary neighborhoods, although the magnitude of the improvement may also be influenced by the spatial distribution, annotation quality, and class composition of the fixed test set. Verification of additional mining areas is therefore needed.

TIR-PointNet++ did not obtain the highest value for every metric. Point Transformer achieved an mAcc of 97.6%, compared with 97.2% for TIR-PointNet++. mAcc averages class-specific recall, whereas mIoU considers both false-positive and false-negative assignments. The results therefore indicate that Point Transformer obtained a slightly higher average class recall, while TIR-PointNet++ achieved higher overall accuracy and overlap quality. This difference reinforces the need to evaluate multiple metrics rather than describe one method as uniformly superior.

The experimental comparison included four representative baselines covering global point-set encoding, hierarchical neighborhood learning, surface representation, and attention-based modeling. The reported advantages should therefore be interpreted within this comparison set rather than as a comprehensive ranking of all recent point-cloud segmentation networks.

5.3. Complementary Outputs and Potential Applications

The dual-module workflow provides two complementary forms of information from the same UAV LiDAR point cloud. The geometry-based branch delineates candidate bench-line structures that describe principal platform–slope transitions, whereas the semantic-segmentation branch assigns point-wise labels to buildings, roads, and vegetation. The two branches operate independently and address different mapping requirements.

The extracted bench-line structures can provide auxiliary geometric information for mine-surface mapping and bench-morphology interpretation. The semantic labels can support haul-road delineation and the identification of facilities and vegetation. Together, these outputs provide structural and object-level information that may facilitate mine-map updating and surface interpretation. Their incorporation into operational monitoring or mine-management systems requires application-specific validation using additional survey periods and mining areas.

5.4. Limitations and Future Work

The present study has several limitations.

First, the evaluation was conducted at the East Open-Pit Mine, and differences in terrain morphology, bench dimensions, surface roughness, point density, and acquisition geometry may affect transferability to other sites. Multi-mine and multi-temporal datasets are therefore required to assess generalization under different environmental and operational conditions.

Second, the semantic-segmentation task contains three categories—buildings, roads, and vegetation—which do not represent the complete composition of an active open-pit mine. Future datasets should incorporate additional mine-specific categories, including working platforms, slope faces, exposed rock, waste dumps, coal piles, water bodies, conveyor facilities, and mining equipment.

Third, the bench-line results were evaluated through spatial and morphological comparison. As an initial quantitative extension, two or three representative bench-line segments should be independently digitized on the orthophoto by an operator not involved in method development, and the mean and standard deviation of perpendicular distances should be reported. A broader benchmark should subsequently use independently interpreted or surveyed reference lines to evaluate completeness, correctness, F1-score, Hausdorff distance, and continuity-related measures and to compare the geometry-based branch directly with existing open-pit step-line extraction methods.

Fourth, semantic-segmentation performance was assessed using a fixed spatial hold-out setting. As a minimum stability check, future experiments should include at least three independent training runs with different random initializations on the same fixed hold-out and report the results as mean $\pm$ standard deviation. Alternative spatial partitions should then be evaluated to quantify sensitivity to data division. Parameter-sensitivity analysis would also help determine how neighborhood size, angular-gap threshold, voxel scale, and network configuration affect the results.

Fifth, computational cost was not evaluated in the present study. Future experiments should report model parameters, floating-point operations, GPU memory consumption,

inference time, and point-processing throughput to assess suitability for large-area and frequent-update applications.

Finally, both modules may be affected by uneven point density, steep-slope occlusion, scanning-angle variation, seasonal vegetation, mobile equipment, incomplete ground returns, and annotation uncertainty. Future work should examine these conditions systematically and investigate feature-level or decision-level interaction between the structural and semantic branches.

6. Conclusions

This study developed a dual-module workflow for extracting bench-line structures and segmenting typical surface objects from UAV LiDAR point clouds in an open-pit mine. The same preprocessed point cloud is independently processed by a neighborhood geometry-based branch and a TIR-PointNet++-based semantic-segmentation branch. The former identifies structural transitions between working platforms and slope faces, while the latter assigns point-wise labels to buildings, roads, and vegetation. The workflow therefore provides complementary geometric and semantic information through two task-specific branches rather than an end-to-end fusion network.

The bench-line branch combines normal-vector and curvature descriptors, RANSAC-based local plane fitting, angular-gap analysis, regional grouping, and Kalman-filter-based line refinement. The extracted lines showed visually coherent correspondence with the principal platform–slope transitions represented in the orthophoto, and the refined line sequences appeared smoother and more continuous than the original candidate-point distribution. The bench-line results were evaluated qualitatively through their spatial correspondence with visible platform–slope morphology and therefore represent a morphological demonstration rather than a line-level geometric-accuracy assessment. For semantic segmentation, TIR-PointNet++ achieved an OA of 97.6%, an mAcc of 97.2%, and an mIoU of 96.4% under the fixed spatial hold-out configuration. The corresponding IoU values for buildings, roads, and vegetation were 98.6%, 94.8%, and 95.9%, respectively. It obtained the highest OA, mIoU, and class-wise IoU values among the selected baselines, whereas Point Transformer achieved a slightly higher mAcc of 97.6%. The ablation results further showed that both local self-attention and inverted residual feature transformation contributed to the performance improvement observed in the evaluated configuration.

The complementary outputs have potential value for mine-map updating, interpretation of platform–slope relationships, haul-road mapping, and identification of buildings and vegetation. Nevertheless, the present findings are limited to one open-pit mine, three semantic categories, a single fixed spatial hold-out experiment, and qualitative bench-line evaluation. Future work should construct independently interpreted bench-line references, extend the dataset to different mines and acquisition periods, incorporate additional mine-specific object classes, conduct repeated-initialization and alternative spatial-partition experiments, and evaluate parameter sensitivity and computational cost.

Author Contributions: Conceptualization, S.G., N.Q. and J.G.; methodology, S.G., N.Q. and D.Y.; software, S.G. and D.Y.; validation, S.G., N.Q., X.L., W.Z., Y.F. and D.Y.; formal analysis, S.G., N.Q. and X.L.; investigation, S.G., X.L., W.Z. and Y.F.; resources, S.G., J.G. and Y.F.; data curation, S.G., W.Z. and Y.F.; writing—original draft preparation, S.G. and N.Q.; writing—review and editing, N.Q., J.G., X.L., W.Z., Y.F. and D.Y.; visualization, S.G., N.Q. and X.L.; supervision, N.Q. and J.G.; project administration, N.Q. and J.G.; funding acquisition, N.Q. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant Nos. 42504028, 42304046 and 42404016), the Basic Research Program of Jiangsu Province (Grant No. BK20241665), the Fundamental Research Funds for the Central Universities (Grant Nos. 2024ZDPYCH1003 and 2025QN1109).

Institutional Review Board Statement: Not applicable. This study did not involve humans or animals.

Informed Consent Statement: Not applicable. This study did not involve human participants.

Data Availability Statement: The UAV LiDAR and auxiliary geospatial data used in this study contain mine-related spatial information and are subject to enterprise data-management and mine-safety restrictions; therefore, they are not publicly available. All figures, tables, and numerical results reported in this study are included in the article.

Acknowledgments: The authors would like to thank China Coal Pingshuo Group Co., Ltd. for providing support during the UAV LiDAR data acquisition and field investigation. The authors also acknowledge the technical assistance provided during point-cloud data processing, field verification, and experimental implementation.

Conflicts of Interest: Author Shanfeng Ge was employed by the company China Coal Pingshuo Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Tsachouridis, S.; Pavloudakis, F.; Sachpazis, C.; Tsioukas, V. Monitoring Slope Stability: A Comprehensive Review of UAV Applications in Open-Pit Mining. *Land* **2025**, *14*, 1193. [CrossRef]
2. Le Roux, R.; Sepehri, M.; Khaksar, S.; Murray, I. Slope Stability Monitoring Methods and Technologies for Open-Pit Mining: A Systematic Review. *Mining* **2025**, *5*, 32. [CrossRef]
3. Weinmann, M.; Jutzi, B.; Mallet, C. Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [CrossRef]
4. Hackel, T.; Wegner, J.D.; Schindler, K. Fast Semantic Segmentation of 3D Point Clouds with Strongly Varying Density. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *III-3*, 177–184. [CrossRef]
5. Ni, H.; Lin, X.; Ning, X.; Zhang, J. Edge Detection and Feature Line Tracing in 3D-Point Clouds by Analyzing Geometric Properties of Neighborhoods. *Remote Sens.* **2016**, *8*, 710. [CrossRef]
6. Wang, Z.; An, S.; Zou, J.; Zhang, Z. Step Line Extraction from Point Cloud Data of Open-Pit Mine. *J. Northeast. Univ. (Nat. Sci.)* **2021**, *42*, 1323–1328. [CrossRef]
7. Mao, Y.; Wang, H.; Cao, W.; Fu, Y.; Fu, Y.; He, L.; Bao, N. Extraction of Step-Feature Lines in Open-Pit Mines Based on UAV Point-Cloud Data. *Sensors* **2022**, *22*, 5706. [CrossRef] [PubMed]
8. Cui, P.; Meng, X.; Zhang, W. Step Feature Line Extraction from Large-Scale Point Clouds of Open-Pit Mine Based on Structural Characteristics. *Sci. Rep.* **2024**, *14*, 18423. [CrossRef] [PubMed]
9. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [CrossRef] [PubMed]
10. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; IEEE: New York, NY, USA, 2017; pp. 652–660.
11. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5099–5108.
12. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*, 146. [CrossRef]
13. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution on X-Transformed Points. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 820–830.
14. Wu, W.; Qi, Z.; Fuxin, L. PointConv: Deep Convolutional Networks on 3D Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019*; IEEE: New York, NY, USA, 2019; pp. 9621–9630. [CrossRef]
15. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019*; IEEE: New York, NY, USA, 2019; pp. 6411–6420. [CrossRef]

16. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, NY, USA, 2020; pp. 11108–11117.
17. Landrieu, L.; Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018*; IEEE: New York, NY, USA, 2018; pp. 4558–4567.
18. Choy, C.; Gwak, J.; Savarese, S. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019*; IEEE: New York, NY, USA, 2019; pp. 3075–3084.
19. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.S.; Koltun, V. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021*; IEEE: New York, NY, USA, 2021; pp. 16259–16268.
20. Ran, H.; Liu, J.; Wang, C. Surface Representation for Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022*; IEEE: New York, NY, USA, 2022; pp. 18942–18952. [CrossRef]
21. Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; Ghanem, B. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 23192–23204. [CrossRef]
22. Sevinc, H.K.; Karas, I.R. Deep Learning-Based Semantic Segmentation of Airborne LiDAR Point Clouds Using a Transformer-Enhanced PointNet++ Architecture. *Geomatics* **2026**, *6*, 43. [CrossRef]
23. Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; Zhao, H. Point Transformer V3: Simpler, Faster, Stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024*; IEEE: New York, NY, USA, 2024; pp. 4840–4851.
24. Fan, S.; Dong, Q.; Zhu, F.; Lv, Y.; Ye, P.; Wang, F.-Y. SCF-Net: Learning Spatial Contextual Features for Large-Scale Point Cloud Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021*; IEEE: New York, NY, USA, 2021; pp. 14504–14513.
25. Lu, T.; Wang, L.; Wu, G. CGA-Net: Category Guided Aggregation for Point Cloud Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021*; IEEE: New York, NY, USA, 2021; pp. 11693–11702.
26. Qiu, S.; Anwar, S.; Barnes, N. Semantic Segmentation for Real Point Cloud Scenes via Bilateral Augmentation and Adaptive Fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021*; IEEE: New York, NY, USA, 2021; pp. 1757–1767.
27. Varney, N.; Asari, V.K.; Graehling, Q. DALES: A Large-Scale Aerial LiDAR Data Set for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020*; IEEE: New York, NY, USA, 2020; pp. 186–187.
28. Kölle, M.; Laupheimer, D.; Schmohl, S.; Haala, N.; Rottensteiner, F.; Wegner, J.D.; Ledoux, H. The Hessigheim 3D (H3D) Benchmark on Semantic Segmentation of High-Resolution 3D Point Clouds and Textured Meshes from UAV LiDAR and Multi-View-Stereo. *ISPRS Open J. Photogramm. Remote Sens.* **2021**, *1*, 100001. [CrossRef]
29. Tan, W.; Qin, N.; Ma, L.; Li, Y.; Du, J.; Cai, G.; Yang, K.; Li, J. Toronto-3D: A Large-Scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020*; IEEE: New York, NY, USA, 2020; pp. 202–203.
30. Hu, Q.; Yang, B.; Khalid, S.; Xiao, W.; Trigoni, N.; Markham, A. Towards Semantic Segmentation of Urban-Scale 3D Point Clouds: A Dataset, Benchmarks and Challenges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021*; IEEE: New York, NY, USA, 2021; pp. 4977–4987.
31. Borja-Jaimes, V.; Valdez-Martínez, J.S.; Beltrán-Escobar, M.; Ramírez-Zúñiga, G.; Reyes-Mayer, A.; Calixto-Rodríguez, M. Robust Backstepping-Sliding Control of a Quadrotor UAV with Disturbance Compensation. *Computation* **2026**, *14*, 51. [CrossRef]
32. Borja-Jaimes, V.; García-Morales, J.; Escobar-Jiménez, R.F.; Guerrero-Ramírez, G.V.; Adam-Medina, M. A Backstepping Sliding Mode Control of a Quadrotor UAV Using a Super-Twisting Observer. *Appl. Sci.* **2025**, *15*, 10120. [CrossRef]
33. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
35. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018*; IEEE: New York, NY, USA, 2018; pp. 4510–4520. [CrossRef]
36. Zhou, Q.-Y.; Park, J.; Koltun, V. Open3D: A Modern Library for 3D Data Processing. *arXiv* **2018**, arXiv:1801.09847.
37. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8024–8035.

38. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
39. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.