

Article

Dynamic Die-Forging Scene Semantic Segmentation via Point Cloud–BEV Feature Fusion with Star Encoding

Xuwen Feng ^{1,2}, Aiming Wang ^{1,*}, Guoying Meng ¹, Yiyang Xu ³, Jie Yang ¹, Xiaohan Cheng ¹ , Yijin Xiong ² and Juntao Wang ^{4,5,6}

¹ School of Mechanical and Electrical Engineering, China University of Mining and Technology-Beijing, Beijing 100083, China

² School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China

³ Beijing China Coal Mine Engineering Co., Ltd., Beijing 100013, China

⁴ State Key Laboratory of Digital Intelligent Technology for Unmanned Coal Mining, Zhangjiakou 076250, China

⁵ China Coal Zhangjiakou Coal Mining Machinery Co., Ltd., Zhangjiakou 076250, China

⁶ Hebei Province High-end Intelligent Mine Equipment Technology Innovation Center, Zhangjiakou 076250, China

* Correspondence: 201418@cumtb.edu.cn; Tel.: +86-134-8871-3431

Abstract

Semantic segmentation of workpieces and die cavities is critical for intelligent process monitoring and quality control in hammer die-forging. However, the field of 3D point cloud segmentation currently faces prominent limitations in forging scenario adaptation: existing state-of-the-art (SOTA) methods are predominantly optimized for road driving or indoor scenes, where targets have stable poses and regular surfaces. They lack dedicated designs for capturing the fine-grained deformation characteristics of forging workpieces and alleviating multi-scale feature misalignment caused by large pose variations—key pain points in forging segmentation. Consequently, these methods fail to balance segmentation accuracy and real-time efficiency required for practical forging applications. To address this gap, this paper proposes a novel semantic segmentation framework fusing 3D point cloud and bird’s-eye-view (BEV) representations for complex die-forging scenes. Specifically, a Star-based encoding module is designed in the BEV encoding stage to enhance capture of fine-grained workpiece deformation characteristics. A hierarchical feature-offset alignment mechanism is developed in decoding to alleviate multi-scale spatial and semantic misalignment, facilitating efficient cross-layer fusion. Additionally, a weighted adaptive fusion module enables complementary information interaction between point cloud and BEV modalities to improve precision. We evaluate the proposed method on our self-constructed simulated and real die-forging point cloud datasets. The results show that when trained solely on simulated data and tested directly in real-world scenarios, our method achieves an mIoU that surpasses RPNNet by 1.1%. After fine-tuning with a small amount of real data, the mIoU further improves by 5%, reaching optimal performance.

Keywords: point cloud; forging; star-operation; alignment mechanism; feature fusion



Academic Editors: Junyu Qi, Dandan Peng, Xiaoxi Hu and Peng Chen

Received: 18 December 2025

Revised: 16 January 2026

Accepted: 18 January 2026

Published: 21 January 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

During the hammer forging process, a workpiece is subjected to frequent pose variations and severe surface deformation under high temperature and pressure conditions. Such non-rigid deformations significantly increase the difficulty of 3D point cloud-based object recognition. While recent point cloud recognition techniques, such as PointNet [1]

and PointNet++ [2], have made considerable progress in semantic understanding, they are primarily designed for environments with relatively stable geometric structures (e.g., roads and indoor scenes). Consequently, these methods are ill-suited for recognizing complex, non-rigid workpieces in industrial manufacturing settings.

Based on different representation paradigms, current 3D point cloud semantic segmentation methods can be broadly categorised into two groups: 3D representation-based methods and 2D projection-based methods. 3D representation-based approaches primarily include methods that process raw point clouds directly and those that utilise voxel representations. Raw point cloud-based methods, such as DGCNN [3] and RandLA-Net [4], typically employ specialised modules to extract features directly from the unordered point sets. DGCNN [3] constructs graph structures to capture local geometric relationships within the point cloud, while RandLA-Net [4] employs an efficient local feature aggregation scheme to handle large-scale point cloud data. Voxel-based methods, exemplified by SparseConv [5] and Cylinder3D [6], first convert point clouds into structured voxel grids and then utilise 3D convolutional operations for feature extraction, enabling effective neighbourhood feature learning and object recognition. SparseConv [5] enhances computational efficiency through sparse convolution, whereas Cylinder3D introduces a cylindrical voxel partitioning strategy to improve long-range contextual modelling. While these 3D representation-based methods effectively preserve spatial geometric information, they often require substantial computational resources during both training and inference, leading to slower processing speeds.

A 2D-projection-based method for point-cloud semantic segmentation first projects the point cloud onto a 2D plane; then, 2D convolutional operations are used to extract planar features and perform semantic understanding on the 2D plane; finally, the 2D semantic features are back-projected into 3D space to achieve semantic understanding of the 3D point cloud. The 2D planar projection can be implemented either as a bird's-eye view (BEV) or a range view (RV). For example, RangeNet++ [7] converts the point cloud into a range image via spherical projection and applies a 2D convolutional neural network for semantic segmentation, achieving good performance. PolarNet [8] adopts a projection in polar coordinates on the BEV and uses convolutional neural networks to perform point-cloud semantic segmentation, yielding excellent results. Projection-based point-cloud semantic segmentation methods offer computational efficiency advantages, but they may suffer from partial loss of spatial information.

To harness the complementary advantages of different point cloud representations, recent research has introduced multi-view fusion strategies, which primarily fall into two categories: (1) the fusion of range view and bird's-eye view representations, where methods like AMVNet [9] integrate BEV and RV features at late stages to mitigate the limitations of individual projection perspectives; (2) the fusion of 3D point clouds with 2D projected views, exemplified by RPNNet [10] that combines raw point cloud, voxel, and range view features during decoding, and CPGNet [11] which employs a joint architecture with dedicated point cloud, RV, and BEV branches to effectively enhance per-point semantic segmentation performance.

Inspired by multi-view fusion frameworks, we introduce a dual-branch semantic segmentation network specifically designed for forging point clouds, which simultaneously exploits raw 3D geometry and structured BEV representations. One branch processes point clouds directly to preserve detailed spatial information, while the other operates in the BEV domain to efficiently encode contextual features. In the BEV encoder, a star-based encoding module is adopted to improve feature expressiveness, and a hierarchical alignment strategy in the decoder ensures robust spatial consistency across scales. To facilitate effective cross-view interaction, we further incorporate a weighted adaptive fusion module that

dynamically balances contributions from both views, resulting in superior segmentation accuracy and robustness over conventional single-view models.

In summary, the main contributions of this work are threefold:

1. We construct a comprehensive semantic dataset of forging point clouds encompassing both simulated and real-world scenarios, providing an essential data foundation for 3D recognition tasks in complex industrial environments.
2. We propose a novel semantic segmentation model that integrates 3D point clouds with Bird's-Eye View (BEV) for forging applications. Specifically, for the BEV branch encoding stage, we design a star-based encoding module and a hierarchical feature alignment mechanism to enhance the encoder's nonlinear mapping capability and feature representation capacity. During the BEV decoding phase, we introduce a multi-level feature offset calibration module to address feature misalignment caused by downsampling operations, enabling effective feature alignment throughout the upsampling process. Furthermore, we develop a weighted adaptive feature fusion module to achieve dynamic integration of cross-view features between the point view and BEV representation, significantly improving the accuracy and robustness of forging point cloud segmentation.
3. The results obtained by training on the synthetic dataset and testing on the real dataset show that PBNNet improves the mIoU metric by 1.1% compared to RPNNet. After further fine-tuning on the real dataset, PBNNet achieves an mIoU of 85.9%, still significantly outperforming RPNNet and PTV3 and maintaining the best performance.

2. Related Work

Point-based Segmentation. PointNet [1] pioneered direct processing of point clouds by using shared MLPs to extract per-point features and symmetric functions to aggregate unordered point sets, establishing an effective paradigm for point cloud segmentation. While subsequent refinements [2,12–15] have demonstrated strong performance on indoor benchmarks, most remain constrained by computational and memory limitations when scaling to large-scale scenarios. To address these challenges, RandLA-Net [4] introduced random sampling with local feature aggregation to minimize information loss, while KPConv [15] achieved notable accuracy through its innovative kernel point convolution. Most recently, Point Transformer V3 [16] has advanced the field by specifically targeting the accuracy-efficiency trade-off in point cloud processing. Despite generally maintaining low parameter counts, point-based methods remain constrained by their notoriously inefficient local neighborhood queries, which lead to substantial computational overhead during feature extraction.

Voxel-based Segmentation. Voxel-based methods [17–23] typically convert unstructured point clouds into regular voxel grids and employ 3D convolutional networks for semantic segmentation. Since standard 3D convolutions incur substantial computational and memory costs in high-resolution scenarios, researchers have developed various improvements to alleviate this spatiotemporal complexity. In 2019, Choy et al. proposed Minkowski Convolutional Neural Networks [24], which significantly reduce the computational burden of 3D feature extraction through sparse tensor representations. In 2020, Tang et al. further introduced Sparse Point-Voxel CNNs [5], effectively reducing computational complexity while maintaining representational capacity. That same year, Zhu et al. developed Cylinder3D [6], which utilises asymmetric residual blocks to minimise redundant computations while better preserving geometric features of elongated objects. Building upon these advances, AF2S3Net [25] incorporates an Attention Feature Fusion Module (AF2M) and an Adaptive Feature Selection Module (AFSM) to more effectively integrate local and global contexts while emphasising fine-grained detail features. Nevertheless,

voxel-based methods still face significant information loss at low resolutions, while high resolutions lead to dramatically increased computational complexity, creating an inherent and unavoidable trade-off between accuracy and efficiency in voxel-based modelling.

2D Projection-based Segmentation. Compared to point-based and voxel-based approaches, projection-based methods generally achieve faster inference speeds. The core idea involves projecting sparse 3D point clouds onto regular 2D representation spaces, thereby leveraging well-established 2D convolutional or Transformer architectures for feature extraction. These methods are mainly categorised into two types: Range View (RV) projection and Bird's-Eye View (BEV) projection. In RV-based methods, RangeNet++ [7] was the first to systematically explore the potential of mapping point clouds into range images combined with 2D convolutions. It also introduced a GPU-accelerated k-nearest neighbours (KNN) post-processing scheme to ensure semantic consistency within local point neighbourhoods. Subsequently, RangeViT [26] incorporated pre-trained Vision Transformers (ViT) along with a point-wise semantic refinement module, significantly improving semantic prediction accuracy. Building upon this foundation, RangeFormer [27] established an end-to-end RV framework that integrated a novel network architecture, data augmentation techniques, and post-processing mechanisms. It further introduced a Scalable Training from Range View (STR) strategy, substantially enhancing the applicability and generalisation capability of RV-based methods. In contrast, BEV-based methods focus more on modelling global spatial structures in the bird's-eye plane. PolarNet [8] pioneered the use of polar coordinate BEV representations to mitigate the non-uniform point distribution issues inherent in Cartesian coordinates and designed ring-shaped convolutions to adapt to the polar coordinate space. Following this direction, Panoptic-PolarNet [28] extended semantic segmentation by incorporating instance clustering, achieving panoramic LiDAR point cloud segmentation and demonstrating stronger scene understanding capabilities.

Multi-View Fusion. Given the inherent limitations of single-view representations in expressive power, recent years have witnessed the emergence of various methods [9,29–32] that attempt to fuse features from multiple views. For instance, refs. [29,30] integrate point-level information from both Bird's-Eye View (BEV) and Range View (RV) at the input stage, achieving early fusion of cross-view features. In contrast, AMVNet [9] proposes an uncertainty-aware late fusion strategy, which estimates the uncertainty of predictions from different views and employs an additional network to refine the results, thereby enhancing segmentation consistency. FusionNet [31] designs a point-voxel interactive MLP that efficiently aggregates features between neighbouring voxels and corresponding points, reducing the computational overhead of neighbourhood search while maintaining a favourable balance between accuracy and efficiency on large-scale point cloud data. PVCNN [33] offers a lightweight point-voxel fusion scheme, where the voxel branch provides coarse-grained local structural information while the point branch preserves fine-grained geometric features through point-wise MLPs. RPVNet [10] integrates three distinct representations—points, voxels, and range views—constructing a deep fusion framework with multi-modal interactive information flow and designing a gated fusion module to adaptively combine features from the three branches. Furthermore, CPGNet [11] combines BEV, RV, and raw point clouds to propose a multi-modal fusion model that achieves an improved balance between accuracy and inference speed. Overall, these multi-view fusion methods effectively mitigate the shortcomings of single-view representations by leveraging complementary advantages, demonstrating enhanced robustness and generalisation capability in complex scenarios.

3. Data and Methods

This section first briefly introduces the data acquisition equipment and sample data, then outlines the overall architecture of the proposed model, and provides a detailed description of its core components: the bidirectional projection mechanism between point clouds and Bird's Eye View (BEV), the star-operation feature encoding strategy and the multi-level feature alignment module in the BEV branch, and the final weighted feature fusion module.

3.1. Data Acquisition

To address the failure of traditional visual recognition methods under harsh forging conditions such as intense light and high temperatures, this study employs a Mech-Eye LSR L long-range industrial 3D laser scanner to collect point cloud data on both a high-fidelity simulated forging platform and a real industrial production line. The Mech-Eye LSR L is an industrial-grade 3D vision sensor capable of generating high-quality, detailed 3D point clouds with strong resistance to ambient light interference, a large field of view, and precise geometric measurements even under challenging lighting conditions. It is particularly suitable for factory automation and robotics guidance tasks in environments with strong environmental interference, such as bright reflections and dynamic lighting changes.

As illustrated in Figure 1, during the hammer forging process, 2D images are highly susceptible to environmental disturbances (e.g., drastic illumination variations, thermal radiation, and specular reflections), which can lead to significant distortion or instability of appearance information. This inconsistency greatly increases the difficulty of transferring image-based semantic understanding methods from simulation to reality, thereby limiting their applicability in real production scenarios. In contrast, 3D point clouds exhibit excellent geometric consistency across different scenes: the global topology and local geometric details of the workpiece and die cavity remain highly stable in both simulated and real forging environments. Such robustness makes 3D point clouds an ideal input modality for modeling and recognition tasks, effectively mitigating the challenges caused by appearance changes due to environmental interference.

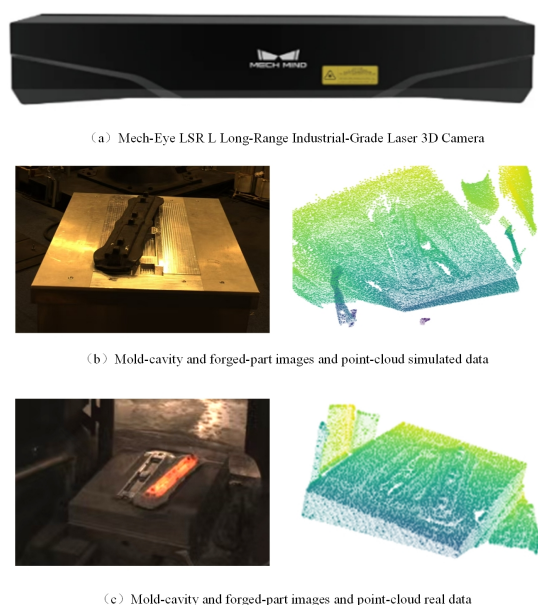


Figure 1. Data acquisition device and collected samples. (a) Diagram of the Mech-Eye LSR L long-range industrial 3D laser camera; (b) Image and point cloud data acquired in the simulated environment; (c) Image and point cloud data obtained from the real-world forging scenario. Comparative analysis between subfigures (b,c) reveals substantial differences in the image data across different environments, while the corresponding forging point clouds maintain high structural consistency.

Based on the above observations, we select point clouds as the core data source for semantic segmentation of die cavities and forged workpieces. By leveraging the structural consistency of point clouds in both simulation and real-world scenarios, we significantly reduce the domain gap between simulated training data and real test data, providing a feasible path toward cross-domain generalization without requiring extensive real-world annotations.

3.2. Framework Overview

The overall architecture of the proposed PBNet is illustrated in Figure 2. The framework consists of two main components: a raw point cloud branch and a Bird's-Eye View (BEV) branch. The raw point branch employs a two-layer Multi-Layer Perceptron (MLP) for initial per-point feature extraction, followed by a point-to-BEV (P2B) projection that maps the point features onto the BEV plane to efficiently construct neighbourhood representations in the 2D domain. The encoder of the BEV branch comprises four stages, each containing multiple stacked star-based encoding modules (SEM) designed to progressively enhance the representation capacity of local features and improve spatial context modelling. In the decoding phase, a multi-level feature alignment module (MFAM) is introduced to alleviate potential spatial misalignment and feature distortion caused by downsampling, ensuring effective integration of multi-level features. Following BEV feature extraction, a BEV-to-point (B2P) operation is applied to back-project the BEV features into 3D space, where they are adaptively fused with the original point features via a weighted feature fusion module (WFFM), yielding semantically enriched output representations. Finally, the fused features are mapped to predefined point-wise categories, achieving high-precision semantic segmentation of forged workpieces.

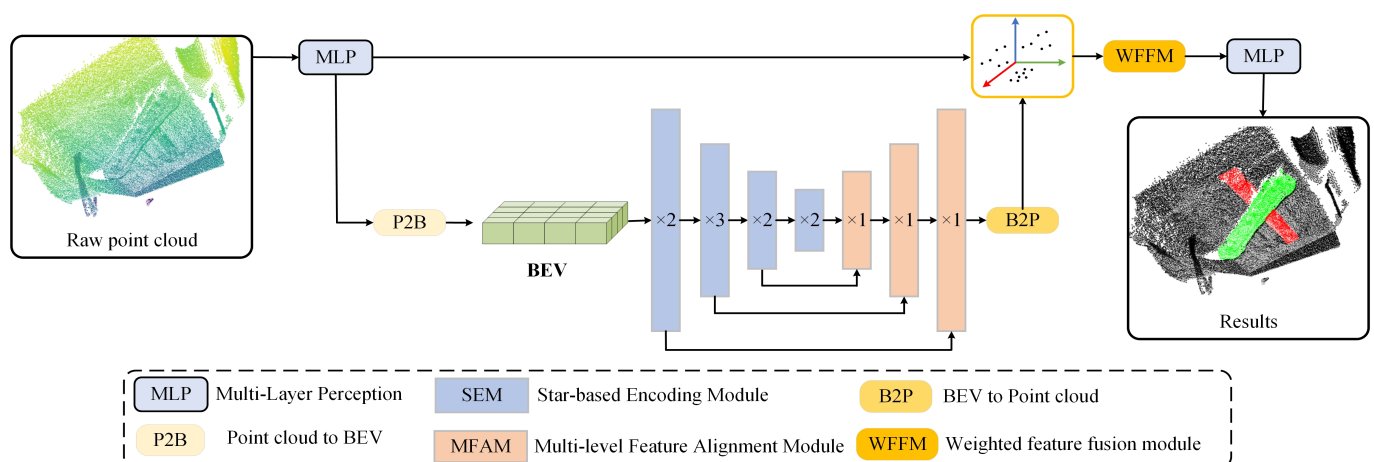


Figure 2. Network architecture of the point cloud and Bird's-Eye View feature fusion model (PBNet). The model consists of an original point cloud branch and a bird's-eye-view (BEV) branch. “ $\times L$ ” denotes the number of repeated layers in each component. The BEV features are constructed via P2B projection. BEV features are then projected back into the 3D space through a B2P operator. Arrows indicate the feature propagation and fusion process, while different colors are used to distinguish different functional modules.

3.3. Point-to-BEV and BEV-to-Point

The Point-to-BEV (P2B) operation [11] is illustrated in Figure 3a, where the black arrows represent max-pooling operations used to aggregate 3D spatial points projected into the same grid cell, thereby obtaining 2D grid coordinates corresponding to the 3D coordinates. The Point-to-BEV projection is formally described as follows: Given a point cloud set $\mathcal{P} = \{p_j \mid j = 1, \dots, N\}$, where N represents the total number of points in $\mathcal{P}$, we take an arbitrary point p_j with 3D coordinates (x_j, y_j, z_j) and project it onto the Bird's-Eye

View (BEV) plane, as illustrated by the green point in Figure 3a, to obtain the corresponding 2D coordinates (u_j, v_j) . During this projection, multiple points from the point cloud set $\mathcal{P}$ can be projected onto the same 2D grid cell. Therefore, let the set $\mathcal{G}_{h,w}$ denote the indices of all points p_j that fall into the same grid cell (h, w) , as shown by the orange points in Figure 3a. Formally, this can be expressed as:

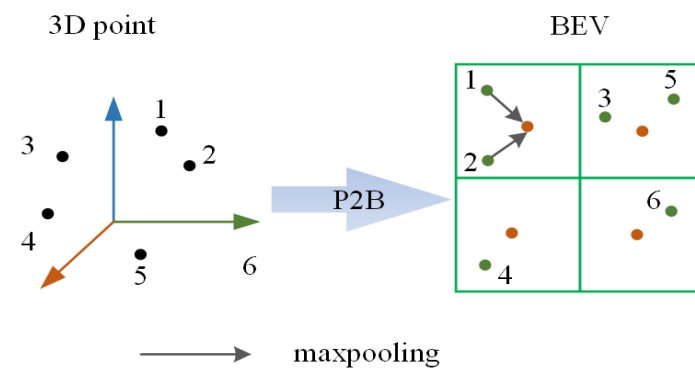
$$\mathcal{G}_{h,w} = \{j \mid h = \lfloor u_j \rfloor, w = \lfloor v_j \rfloor\}, \quad (1)$$

where $\lfloor \cdot \rfloor$ denotes the floor function. After completing the projection of point features, a max-pooling operation is applied to aggregate the point features $F_{j,c}^{3D}$ within $\mathcal{G}_{h,w}$, generating the corresponding BEV feature $F_{h,w,c}^{2D}$. Thus, the operation of projecting 3D points to 2D grid cells is formulated as follows:

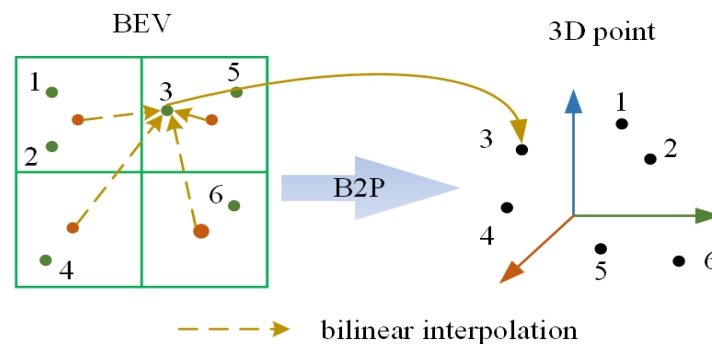
$$F_{h,w,c}^{2D} = \max_{j \in \mathcal{G}_{h,w}} (F_{j,c}^{3D}) \quad (2)$$

Equation (2) summarizes the point feature aggregation method when projecting 3D point clouds onto a 2D plane, as depicted in Figure 3a. In the bird's-eye-view (BEV) representation, 3D points are projected onto the x-y plane and discretized via a rectangular 2D grid. The grid covers the boundary range $(x_{min}, y_{min}, x_{max}, y_{max})$ and is discretized with a spatial resolution r (in meters per pixel), with its width and height are denoted as $W_{bev} = \frac{x_{max} - x_{min}}{r}$ and $H_{bev} = \frac{y_{max} - y_{min}}{r}$, respectively—computed as in Equation (3).

$$\begin{pmatrix} u_j \\ v_j \end{pmatrix} = \begin{pmatrix} \frac{x_j - x_{min}}{x_{max} - x_{min}} \times W_{bev} \\ \frac{y_j - y_{min}}{y_{max} - y_{min}} \times H_{bev} \end{pmatrix} \quad (3)$$



(a) Point-to-BEV operation



(b) BEV-to-point operation

Figure 3. Diagram of (a) P2B and (b) B2P operations.

The BEV-to-Point (B2P) operation [11], serving as the inverse of the Point-to-BEV (P2B) projection, aims to remap 2D features back into 3D space. The specific procedure is illustrated in Figure 3b. This operation consists of two main steps: First, the corresponding 2D grid coordinates (u_j, v_j) are determined for each point, as shown by the orange points in Figure 3b. Then, bilinear interpolation is performed within the four adjacent grid cells surrounding these coordinates. The mathematical formulation is given by:

$$F_{j,c}^{3D} = \sum_{m=0}^1 \sum_{n=0}^1 w_{m,n,j} F_{h+m,w+n,c}^{2D} \quad (4)$$

$$w_{m,n,j} = (1 - |u_j - (h + m)|)(1 - |v_j - (w + n)|)$$

In Equation (4), h and w represent $\lfloor u_j \rfloor$ and $\lfloor v_j \rfloor$ respectively, where $\lfloor \cdot \rfloor$ denotes the floor operation. It should be noted that adjacent grid cells falling outside the valid 2D grid boundaries are treated as zero vectors (zero-padding). Importantly, this computation is performed independently for each point and each feature channel, making it particularly suitable for parallel implementation on CUDA-enabled architectures.

3.4. BEV Branch Encoding Module

3.4.1. Star-Based Encoding Module

The encoder of the BEV branch is built upon a star-based encoding module, as shown in Figure 4. Following the design principle of residual networks, the module introduces a star operation (i.e., element-wise multiplication) to nonlinearly project input features into a high-dimensional representation space, without incurring a significant increase in model complexity.

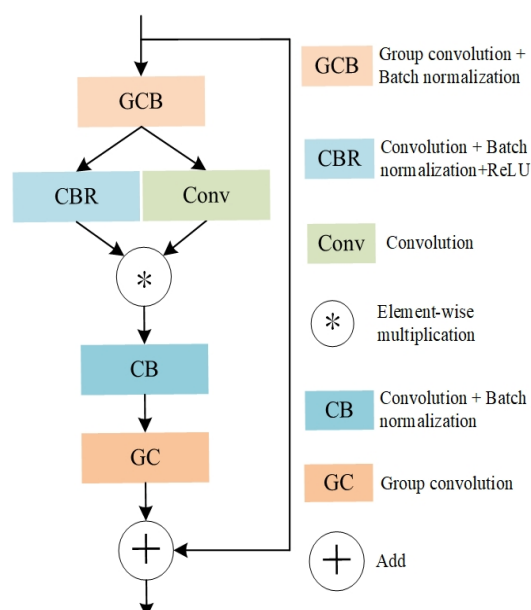


Figure 4. Star-based Encoding Module. This module incorporates grouped convolutional operations and a Star-based operator (*) to effectively capture representative point cloud features, thereby further enhancing encoding efficiency.

Concretely, the input features are first processed by a grouped convolution followed by batch normalization (GCB) to enable channel-wise grouping and reduce computational overhead. The resulting features are then split into two parallel branches: one branch applies a convolution, batch normalization, and ReLU (CBR) block to enhance nonlinearity, while the other branch employs a standard convolution to capture local features. The outputs of the two branches are fused via a star operation (*), performing

element-wise multiplication to facilitate adaptive feature interaction and modulation in the high-dimensional space.

The modulated features are subsequently refined by a convolution with batch normalization (CB) and a grouped convolution (GC) for feature aggregation and dimensionality reduction. Finally, a residual connection adds the original input to the output of the module, which effectively mitigates vanishing gradients and stabilizes training. Notably, the star operation implicitly corresponds to a mapping into an infinite-dimensional space, as theoretically demonstrated by Xu et al. [34]; further analysis is provided in the Appendix A.

3.4.2. Dual-Branch Subsampling Module

To more comprehensively preserve critical detail information during the image downsampling process, this paper designs a Dual-branch Downsampling Module (DDM), as illustrated in Figure 5. Conventional single-path downsampling methods often suffer from local detail loss due to information compression when reducing the resolution of feature maps. To address this issue, the DDM employs two parallel, functionally complementary paths working cooperatively: Branch I utilizes a 3×3 convolutional layer with a stride of 2 to achieve spatial downsampling while retaining relatively rich contextual structure; Branch II first applies a 1×1 convolution to extract local fine-grained features and then performs dimensionality reduction via max-pooling to enhance and preserve the most discriminative local responses. Finally, the feature maps output from the two paths are fused along the channel dimension, thereby improving the representational capacity of the features while compressing resolution, and providing a more comprehensive and discriminative feature representation for subsequent semantic segmentation tasks.

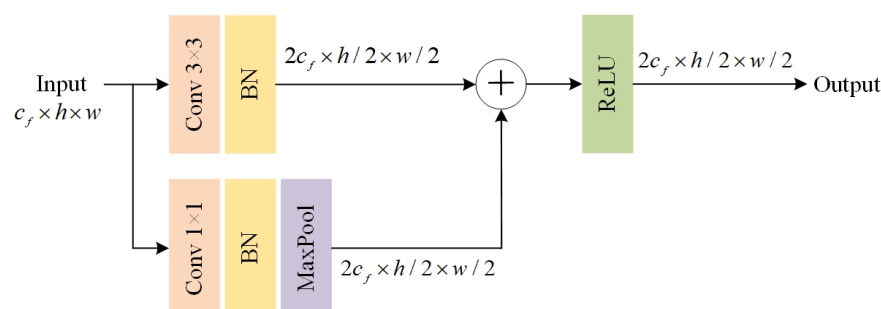


Figure 5. Dual-branch downsampling module. Here, c_f , h , and w denote the number of channels, height, and width of the 2D feature map, respectively. BN denotes Batch Normalization. Different colors represent different feature branches, and the “+” symbol denotes feature fusion by element-wise addition.

3.5. Decoder Module

3.5.1. Multi-Level Feature Alignment Module

The decoder of the BEV branch is composed of a hierarchical feature-alignment module, whose architecture is shown in Figure 6. To effectively preserve edge details, this module first applies convolution and bilinear interpolation to the high-level, low-resolution feature map $F_i \in R^{c_i \times h_i \times w_i}$, to unify its channel number and spatial resolution. Then, the upsampled F_i is concatenated with the channel-unified $F_j \in R^{c_m \times h_j \times w_j}$, and based on this concatenated feature, the module predicts the offsets $\Delta_i \in R^{2 \times h_j \times w_j}$ and $\Delta_j \in R^{2 \times h_j \times w_j}$ between the upsampled F_i and F_j . The obtained offsets Δ_i and Δ_j are then used to align the high-level feature F_i with the low-level feature F_j , respectively.

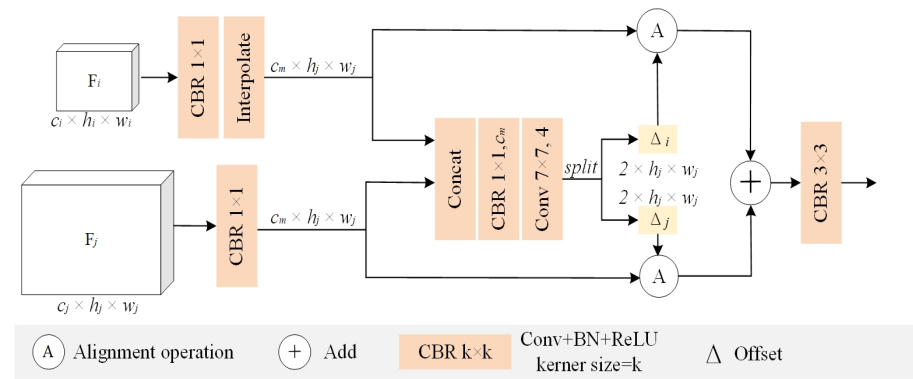


Figure 6. Multi-level feature alignment module. This module takes adjacent decoder-level features as input: a higher-level feature F_i with dimensions $c_i \times h_i \times w_i$ and a lower-level feature F_j with dimensions $c_j \times h_j \times w_j$, where c_i and c_j denote the number of channels, and the spatial resolutions satisfy $2h_i = h_j$ and $2w_i = w_j$. Here, Δ_i and Δ_j represent the offsets of F_i and F_j , respectively.

Once the offset maps are obtained, the feature alignment and aggregation can be performed according to the following formula:

$$A^j = U(\text{upsample}(F_i), \Delta_i) \oplus U(F_j, \Delta_j) \quad (5)$$

where A_j denotes the aligned aggregated feature, $\text{upsample}(\cdot)$ represents the bilinear interpolation function, and $U(\cdot, \cdot)$ indicates the alignment operation. Assume that the spatial coordinates of each position to be aligned on feature map F are $\{(1, 1), (1, 2), \dots, (H, W)\}$, and the offset map is $\Delta \in \mathbb{R}^{2 \times H \times W}$. U_{hw} is the output of the alignment function $U(F, \Delta)$, which is defined as follows:

$$U_{hw} = \sum_{h'=1}^H \sum_{w'=1}^W F_{h'w'} \cdot \max(0, 1 - |h + \Delta_{hw}^1 - h'|) \cdot \max(0, 1 - |w + \Delta_{hw}^2 - w'|) \quad (6)$$

where the feature value at position $(h + \Delta_{hw}^1, w + \Delta_{hw}^2)$ on the feature map F is obtained via bilinear interpolation. Here, Δ_{hw}^1 and Δ_{hw}^2 represent the learned 2D spatial offsets corresponding to the position (h, w) . It should be noted that if the offsets Δ are not learned and both Δ_{hw}^1 and Δ_{hw}^2 in Equation (6) are set to zero, the alignment function will not modify the input feature F , and thus the output U will be identical to F .

3.5.2. Weighted Feature Fusion Module

After completing BEV feature extraction, the BEV features are projected back into 3D space via the B2P operation. To further integrate point-view and bird's-eye view features, we design a weighted feature fusion module, as shown in Figure 7. This module adaptively fuses the B2P-projected features F_{BEV} with the original point-view features F_{point} to mitigate potential feature loss during the projection process. Specifically, given the point-view features F_{point} and the bird's-eye view features F_{BEV} , the weighted fusion module is computed as follows:

$$F_{\text{out}} = \lambda F_{\text{BEV}} \oplus (1 - \lambda) F_{\text{point}} \quad (7)$$

$$\lambda = H(F_{\text{point}}, F_{\text{BEV}})$$

In the Equation (7), F_{out} denotes the output feature, F_{point} represents the original point features, F_{BEV} corresponds to the back-projected Bird's-Eye View features, λ is the adaptive weight, and $H(\cdot)$ indicates the weight computation function, whose implementation is illustrated in Figure 7.

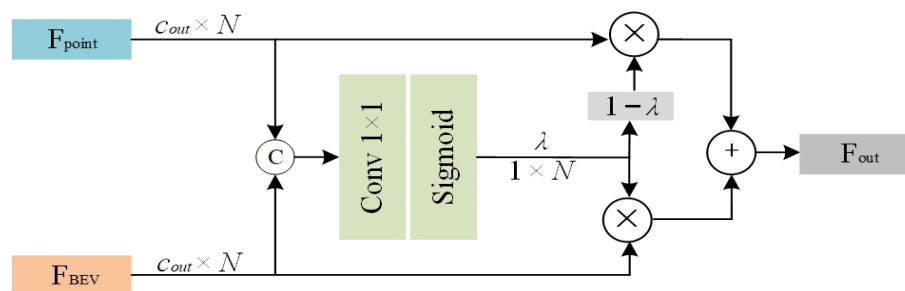


Figure 7. Weighted feature fusion module. The input features consist of the point-view feature F_{point} and the Bird's-Eye View feature F_{BEV} . An adaptive weight λ is computed using a sigmoid function at an intermediate stage to perform point-wise feature fusion. Different colors and symbols are used for visual distinction.

3.6. Loss Function

The segmentation prediction is obtained by applying a fully connected (FC) layer to the output features of the Weighted Feature Fusion Module (WFFM). Due to the pronounced class imbalance in the point cloud semantic segmentation dataset of die cavities and forgings, this work adopts a weighted cross-entropy (WCE) loss function together with the Lovász-Softmax loss to jointly supervise the model training. The overall loss function is defined as follows:

$$L_{\text{total}} = L_{\text{CE}} + 4L_{\text{CE}}^{25\%} + 3L_{\text{LS}} \quad (8)$$

$$L_{\text{CE}} = -\frac{1}{N} \sum_{n=1}^N \sum_{c=1}^C y_n^c \log(\hat{y}_n^c) \quad (9)$$

$$L_{\text{LS}} = \frac{1}{C} \sum_{c=1}^C \Delta J_c(m(c))$$

$$m_i(c) = \begin{cases} 1 - x_i(c), & \text{if } c = y_i(c) \\ x_i(c), & \text{otherwise} \end{cases} \quad (10)$$

In Equation (8), $\mathcal{L}_{\text{total}}$ denotes the total supervised loss of the network, $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss whose explicit form is given in Equation (9) and L_{LS} represents the Lovász-softmax loss, computed as shown in Equation (10). In Equation (9), y_n^c indicates the semantic ground-truth label, and $\hat{y}_n^c$ is the predicted probability for class c at the n -th point. To facilitate more accurate classification of hard samples, an additional loss term $L_{\text{CE}}^{25\%}$ considers only the top 25% of points with the highest loss values. Furthermore, to maximize the Intersection over Union (IoU) score, the Lovász-softmax loss function proposed by Berman et al. [35] is adopted, which is formulated in Equation (10). In Equation (10), ΔJ_c is defined as the Lovász extension of the Jaccard index, C denotes the number of classes, and $x_i(c) \in [0, 1]$ together with $y_i(c) \in \{-1, 1\}$ represent the predicted probability and the ground-truth label, respectively, for class c at pixel i .

3.7. Evaluation Metric

To systematically evaluate the performance of the proposed PBNet and the compared models, this paper adopts the widely used evaluation metric in semantic segmentation the mean Intersection over Union (mIoU) [36]. This metric comprehensively reflects the segmentation accuracy across different categories by calculating the overlap between the predicted regions and the ground-truth annotations for each class and then averaging over all classes. The mIoU is computed as follows:

$$mIoU = \frac{1}{C} \sum_{c=1}^C \frac{TP_c}{TP_c + FP_c + FN_c} \quad (11)$$

where TP_c , FP_c , and FN_c denote the true positives, false positives, and false negatives for class c , respectively, and C represents the total number of classes.

4. Experiments

The performance of the proposed PBNet is evaluated on both simulated and real-world forge point cloud semantic datasets, demonstrating its effectiveness across different environments.

Simulated Forging Point Cloud Semantic Dataset. To systematically replicate the randomness of workpiece placement and the complexity of working conditions in forging workshops, this study first constructs a simulated forging point cloud semantic dataset. This part is carried out in a controlled laboratory environment, where active introduction of artificial occlusion, multi-angle lighting variations, and background interference simulates extreme observation conditions that may occur in real production lines, such as partial occlusion, glare, and low light. Data collection covers a variety of typical forging shapes and placement poses, ultimately forming a high-quality dataset comprising 350 point cloud frames, with each frame containing approximately 10,000 points. This simulated dataset not only provides diverse training samples for the algorithm but also establishes a controllable and reliable experimental foundation for validating the model's generalization capability in subsequent real-world scenarios.

Real-scene Forging Point Cloud Semantic Dataset. To further enhance the engineering applicability and scenario coverage of the dataset, the research team conducted on-site point cloud acquisition in an operational forging workshop using a mobile scanning platform. Data were collected during key processes, including heating, forging, and cooling, under actual production conditions. This portion of data originates entirely from real manufacturing environments and incorporates challenges such as complex background interference and deformations caused by high-temperature forging. A total of 350 point cloud frames were captured on-site. Together with the laboratory-simulated data, they form a composite forging point cloud semantic dataset of 700 frames.

In the experiments, the data are partitioned and used as follows. First, to validate the model's cross-domain generalisation capability, training is conducted on the simulated dataset and evaluation is performed on the real-scene dataset. Second, to further improve the model's performance in practical scenarios, 350 simulated and 70 real samples are selected and combined to form a training set of 420 samples; the remaining real-scene data is reserved for the final performance evaluation.

4.1. Experimental Setup

Network Setup. As shown in Figure 2, the feature extraction modules in PBNet share the same operations but differ in their parameters. In the point branch of PBNet, the MLP at the initial stage takes the 3D coordinates (x, y, z) as input (3 channels) and outputs features with 64 channels. In the BEV branch, the BEV resolution is set to $W = 640$ and $H = 640$. This branch consists of three downsampling and three upsampling stages. The numbers of feature channels in successive stages are 64, 64, 128, 256, 128, 96, and 64, respectively. Consequently, the input to the Weighted Feature Fusion Module (WFFM) are the features from both branches, each with 64 channels, and its output also maintains 64 channels. Finally, point-wise classification is performed through an MLP layer.

Training Details. All experiments were conducted on an NVIDIA RTX 3090 GPU using the PyTorch framework (version 1.12.1) with FP32 precision. The PBNet was trained from random initialisation for 30 epochs with a batch size of 4. Training on two GPUs took approximately 5 h. The optimiser used was Stochastic Gradient Descent (SGD) with a weight decay of 0.001, a momentum of 0.9, and an initial learning rate of 0.02, which was multiplied by 0.1 every 10 epochs. For data augmentation, the following strategies were applied: random rotation around the *z-axis*, random global scaling sampled from [0.95, 1.05], random flipping along the *x-axis* and *y-axis*, and the addition of random Gaussian noise following $N(0, 0.02)$.

4.2. Quantitative Results

To validate the domain discrepancy between simulated and real-world data, we first trained our model on the simulated dataset. Then we evaluated it on the real-world forging semantic dataset. The comparative methods include RandLA-Net [4], PTv3 [16], Cylinder3D [6], and the multi-view fusion architectures RPNNet [10] and CPGNet [11], with their core designs described as follows: RandLA-Net adopts a pure point-cloud processing architecture that achieves lightweight computation through random sampling and local feature aggregation; PTv3 (Point Transformer v3) models global geometric relationships in point clouds based on a self-attention mechanism, yielding strong expressive capability; Cylinder3D projects the point cloud into a cylindrical coordinate system and integrates 2D convolution with 3D voxel feature extraction, showing excellent performance in tasks requiring geometric consistency; RPNNet and CPGNet are both multi-view fusion architectures: the former jointly utilizes point-view and voxel features, while the latter fuses point cloud features with BEV semantics, enhancing segmentation accuracy through complementary cross-view information.

As shown in Table 1, under the cross-domain “sim-train/real-test” setting, the proposed PBNet achieves the best performance in semantic segmentation of both forging workpieces and die cavities, with a mean Intersection-over-Union (mIoU) of 80.9%. This result significantly outperforms existing point-based, voxel-based, and multi-view fusion methods, surpassing the previous best model, RPNNet, by 1.1 percentage points. Moreover, PBNet maintains the fastest inference speed (30 ms) while delivering the highest accuracy, substantially outperforming other high-performance models of the same type (e.g., 50 ms for CPGNet and 165 ms for RPNNet). This indicates that our dual-branch fusion architecture of point cloud and Bird’s-Eye View (BEV) can effectively extract and integrate robust cross-view features, thereby largely meeting the real-time processing requirements of industrial scenarios.

Table 1. Test results on the real-world forging dataset after training on the simulated forging dataset.

Methods	Publication	Forging (%)	Cavity (%)	Speed (ms)	mIoU (%)
KPConv [15]	2019	71.8	71.0	—	71.4
RandLA-Net [4]	2020	74.2	73.1	420	73.7
PointNext [37]	2022	75.3	74.2	—	74.8
OctFormer [38]	2023	77.4	76.3	91	76.9
PTv3 [16]	2024	79.1	78.6	70	78.85
SparseConvNet [39]	2018	70.1	69.5	200	69.8
SPVNAS [5]	2020	78.2	77.9	160	78.0
Cylinder3D [6]	2020	79.5	78.4	170	78.9
RPNNet [10]	2021	80.1	79.4	165	79.8
CPGNet [11]	2022	79.5	78.9	50	79.2
PBNet (Ours)	-	81.6	80.3	30	80.9

To further investigate the model’s adaptability to real-world data, we fine-tuned it using 20% of the real-world forging semantic data. The results are presented in Table 2.

After fine-tuning, all compared methods show notable performance improvements, while PBNet still maintains a comprehensive lead: its mIoU rises to 85.9%, which is approximately 5 percentage points higher than the result obtained with only simulated data, and clearly exceeds the current best counterparts, RPVNet (84.8%) and CPGNet (84.2%). In terms of per-category metrics, PBNet reaches 86.6% for forgings and 85.3% for cavities, both being the highest among all methods. Furthermore, PBNet's inference speed remains consistently fast at 30 ms, far ahead of other models at similar accuracy levels. These results demonstrate that the proposed dual-branch fusion architecture not only possesses excellent cross-domain generalisation ability but can also rapidly adapt to real-world scenarios with only a small amount of real samples. It achieves a superior balance between accuracy and efficiency, providing a reliable solution for industrial real-time point-cloud semantic segmentation.

Table 2. Evaluation results after fine-tuning using partial real-world data.

Methods	Publication	Forging (%)	Cavity (%)	Speed (ms)	mIoU (%)
RandLA-Net [4]	2020	78.4	77.6	420	78.0
OctFormer [38]	2023	82.6	80.9	91	81.8
PTv3 [16]	2024	85.2	83.6	70	84.4
SparseConvNet [39]	2018	74.3	72.1	200	73.2
SPVNAS [5]	2020	83.6	82.9	160	83.25
Cylinder3D [6]	2020	84.7	83.5	170	84.1
RPVNet [10]	2021	85.1	84.4	165	84.8
CPGNet [11]	2022	84.5	83.9	50	84.2
PBNet (Ours)	-	86.6	85.3	30	85.9

4.3. Qualitative Results

A qualitative evaluation was conducted on the fine-tuned models, and the results are presented in Figure 8. This figure illustrates three representative samples (labeled (1)–(3)) and compares the segmentation results of the proposed PBNet with those of the current state-of-the-art methods PTv3 [16] and CPGNet [11]. To intuitively highlight segmentation errors, error results are also provided, where red dots indicate regions where the predictions are inconsistent with the ground truth.

Specifically, in sample (1), PTv3 and CPGNet exhibit substantially more errors than PBNet, with the majority of incorrect regions concentrated at the interface between the forging workpiece and the die cavity, i.e., where the forging occludes the cavity, indicating that both methods have difficulty handling local occlusions. In sample (3), the forging and cavity deeply overlap, forming complex occlusions; PTv3 produces extensive misclassification of forging points, and CPGNet even incorrectly labels portions of the background as forging targets. In contrast, PBNet accurately separates the two and preserves clear and continuous segmentation boundaries at occlusion edges. In sample (2), although the forging and cavity have a small contact area and the overall structure remains intact, PTv3 and CPGNet still generate a large number of misclassified points along edges and fine structures, reflecting their sensitivity to geometric details and insufficient stability. PBNet, on the other hand, yields significantly fewer prediction errors and demonstrates stronger robustness and structural consistency in both complex boundary areas and regions with minimal interference.

Taken together, the qualitative analysis in Figure 8 shows that, compared with PTv3 and CPGNet, the proposed PBNet exhibits significant advantages when confronted with typical industrial challenges such as occlusions, background interference, and complex geometric structures. It achieves more robust and accurate semantic segmentation in real forging scenarios, thereby substantiating the reliability and superiority of the proposed method for practical applications.

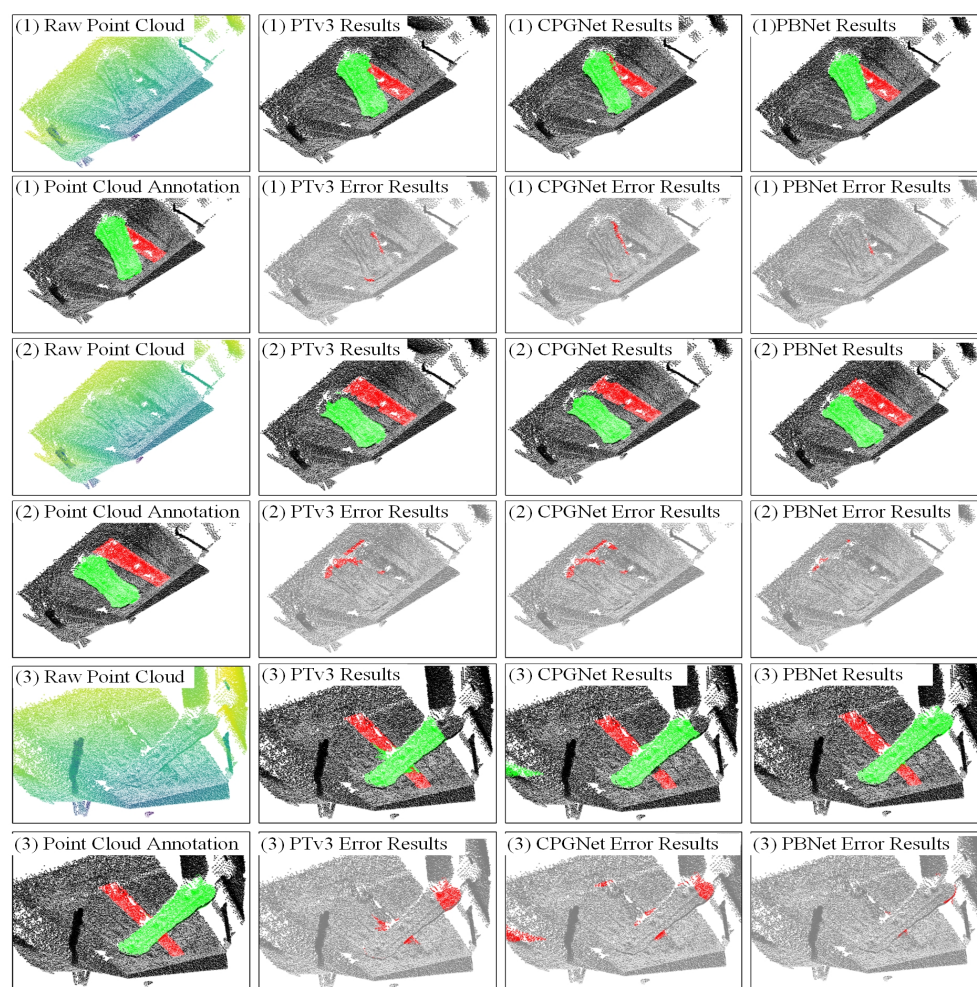


Figure 8. Qualitative comparison of semantic segmentation results on real-scene forging point clouds. The figure presents two sets of visual comparisons for point cloud segmentation. Labels (1)–(3) correspond to three different test samples, where (1), (2), and (3) denote the first, second, and third samples, respectively. In the result maps of each method, green points denote forgings, and red points denote cavities; other colors indicate background or non-target points. In the error maps, red points indicate misclassified points.

4.4. Ablation Study

Impact of network modules. To assess the individual contributions of each module to the network's overall performance, we conducted an ablation study. Specifically, the model was trained using the entire synthetic point-cloud semantic dataset along with 20% of the real-world forging point-cloud semantic data, while the remaining 80% of the real forged point clouds were reserved as a validation set. Based on this setup, we evaluated the following components separately: the star-based encoding module (SEM), the dual-branch downsampling module (DDM), the multi-level feature alignment module (MFAM), and the weighted feature fusion module (WFFM). The results are summarised in Table 3. The first row shows the full model with all modules included, achieving a best mIoU of 85.9%. In the second row, SEM was replaced with a standard ResBlock [40], while all other structures remained unchanged—mIoU dropped to 80.1%, a decrease of 5.8%. This indicates that SEM offers substantially superior feature extraction compared to a standard residual block. In the third row, DDM was substituted by traditional MaxPooling downsampling; the mIoU fell to 82.1%, a 3.8% drop relative to the full model, demonstrating that DDM more effectively preserves multi-scale contextual information. The fourth row removed the alignment mechanism of MFAM in the decoder, replacing it with simple feature concatenation; mIoU

decreased to 81.3%, a reduction of 4.6%, confirming the importance of feature alignment in fusing multi-level information. In the fifth row, WFFM's weighted fusion was replaced with simple concatenation, resulting in an mIoU of 83.2%, a 2.7% decline—indicating that weighted fusion more effectively selects and combines the most salient features.

In summary, all modules positively contribute to improving PBNet's performance. Among them, SEM and MFAM have particularly significant impacts on overall performance, further validating the effectiveness and necessity of the module designs proposed herein.

Table 3. Impact of each network module.

NO.	SEM	DDM	MFAM	WFFM	mIoU
1	✓	✓	✓	✓	85.9
2	✗	✓	✓	✓	80.1
3	✓	✗	✓	✓	82.1
4	✓	✓	✗	✓	81.3
5	✓	✓	✓	✗	83.2

Note: A checkmark (✓) indicates the presence of the corresponding module.

Encoder Backbone Impact. To thoroughly evaluate the performance advantages of the proposed Star-Shaped Encoding Module (SEM), we conducted comparative experiments replacing SEM in PBNet with several current mainstream lightweight encoding modules. Specifically, we substituted SEM with MobileOne Block [41], ShuffleNet Block [42], MobileNetV3 Block [43], and Ghost Block [44], respectively, while keeping the rest of the network architecture and training configuration unchanged. The comparative results are presented in Table 4. Among all compared modules, SEM achieved uniformly superior performance. Its overall mIoU reached 85.9%, outperforming the second-best MobileOne Block by 2.6%, and markedly surpassing the ShuffleNet, MobileNetV3, and Ghost alternatives. For the specific sub-categories “Forging” and “Cavity”, SEM likewise maintained the highest accuracy, demonstrating its superior encoding capability in capturing the critical geometric features of forged-part point clouds. This comparative experiment clearly indicates that SEM significantly outperforms other mainstream lightweight designs in accuracy and is a key component ensuring PBNet's high-precision segmentation.

Table 4. Impact of Star-Based Encoding Module (SEM).

Module	Forging (%)	Cavity (%)	mIoU (%)
MobileOne Block [41]	83.8	82.7	83.3
ShuffleNet Block [42]	81.8	80.1	80.9
MobileNetV3 Block [43]	80.2	79.1	79.7
Ghost Block [44]	79.4	78.7	79.1
SEM	86.6	85.3	85.9

Random Gaussian Noise Impact. As shown in Table 5, when the noise variance is set to $\delta^2 = 0.02$, the model achieves the highest mIoU on the test set (85.9%), representing a clear improvement over the noise-free baseline (84.3%). In contrast, excessively strong noise ($\delta^2 > 0.02$) leads to a degradation in performance, indicating that overly large perturbations can disrupt the effective structural information in the input data. Therefore, selecting $\delta^2 = 0.02$ as the noise intensity strikes a favorable balance between enhancing model robustness and preserving feature integrity, validating the rationality and effectiveness of this setting.

Table 5. The Effect of Random Gaussian Noise $N(\mu = 0, \delta^2)$.

δ^2	0.0	0.01	0.02	0.03	0.04	0.05
mIoU (%)	84.3	84.6	85.9	85.5	85.0	84.7

5. Conclusions

In this paper, we propose a novel semantic segmentation method tailored for the die-forging environment, PBNet, which effectively addresses the misclassification issues caused by drastic changes in workpiece pose and non-rigid surface deformations during the forging process. To tackle the challenges of complex geometric structures and significant domain shifts inherent to this scenario, PBNet incorporates three key architectural innovations: First, a Star-based Encoding Module (SEM) is introduced in the Bird's-Eye-View (BEV) branch, significantly enhancing local-to-global feature modeling in BEV space; compared with the MobileOne backbone, this yields a 2.6% improvement in mIoU. Second, during decoding, a Multi-level Feature Alignment Module (MFAM) is designed to effectively mitigate spatial misalignment among multi-scale features caused by deformation. Finally, at the fusion stage, a Weighted Feature Fusion Module (WFFM) is proposed to achieve adaptive fusion of point cloud geometric features and BEV semantic representations. We conduct a comprehensive evaluation on our self-constructed simulated and real die-forging point cloud datasets. Experiments demonstrate that PBNet, trained solely on simulated data and directly tested in real scenarios, achieves an mIoU approximately 2% higher than the state-of-the-art PointTransformer v3 (PTv3). With minor fine-tuning using a small amount of real data, performance improves further, outperforming all compared methods. These results fully validate the robustness and generalization ability of PBNet in industrial scenarios with strong deformations and cross-domain challenges, providing support for precise 3D semantic understanding in die-forging environments.

Author Contributions: Conceptualization, X.F. and A.W.; methodology, X.F.; software, X.F. and J.Y.; validation, X.C. and A.W.; formal analysis, Y.X. (Yiyang Xu) and Y.X. (Yijin Xiong); investigation, X.F. and A.W.; resources, X.F.; data curation, J.Y.; writing—original draft preparation, G.M.; writing—review and editing, J.W.; visualization, J.W.; funding acquisition, A.W. and X.C. All authors have read and agreed to the published version of the manuscript.

Funding: The National Natural Science Foundation of China under Grant number: 52374167 (Funder: A.W.) and the Fundamental Research Funds for the Central Universities under Grant number: 2024ZKPYJD02 (Funder: X.C.).

Data Availability Statement: All relevant data are available from the corresponding author upon request.

Conflicts of Interest: Author Yiyang Xu was employed by Beijing China Coal Mine Engineering Co., Ltd.; Author Juntao Wang was employed by the State Key Laboratory of Digital Intelligent Technology for Unmanned Coal Mining, China Coal Zhangjiakou Coal Mining Machinery Co., Ltd., and the Hebei Province High end Intelligent Mine Equipment Technology Innovation Center. 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.

Appendix A

The encoder for the BEV branch comprises star-based encoding modules, as illustrated in Figure 4. In a single-layer neural network, the star operation is generally formulated as $(W_1^T X + B_1) * (W_2^T X + B_2)$, which integrates features from two distinct linear transformations via element-wise multiplication. To simplify notation, the weight matrix and bias term can be combined into a single entity denoted as $\hat{W} = [W, B]$, while the input is

extended as $\hat{X} = [X, 1]$. This allows the star operation to be rewritten as $(\hat{W}_1^T \hat{X}) * (\hat{W}_2^T \hat{X})$. For analytical convenience, this paper focuses on the scenario involving a single output channel and a single input element, specifically with $w_1, w_2, x \in \mathbb{R}^{(c_d+1) \times 1}$, where c_d is the number of input channels. The framework can be readily extended to multiple output channels, i.e., $\hat{W}_1, \hat{W}_2 \in \mathbb{R}^{(c_d+1) \times (c_d+1)}$, and can also handle multiple input feature elements, in which case $\hat{X} \in \mathbb{R}^{(c_d+1) \times n}$. In summary, the general form of the star operation [34] can be uniformly expressed as follows:

$$w_1^T x * w_2^T x = \underbrace{\alpha_{(1,1)} x^1 x^1 + \cdots + \alpha_{(4,5)} x^4 x^5 + \cdots + \alpha_{(c_d+1, c_d+1)} x^{c_d+1} x^{c_d+1}}_{(c_d+2)(c_d+1)/2 \text{ items}} \quad (\text{A1})$$

$$\alpha_{(i,j)} = \begin{cases} w_1^i w_2^j & \text{if } i = j, \\ w_1^i w_2^j + w_1^j w_2^i & \text{if } i \neq j. \end{cases} \quad (\text{A2})$$

In Equation (A1), x_k denotes the k -th component of the input feature vector $x = (x^1, x^2, \dots, x^{c_d+1})^T$. In Equation (A2), the indices i and j are used to traverse the channels of the weight matrices $\mathbf{w}_1$ and $\mathbf{w}_2$, while α denotes a per-element coefficient, where i and j to index the channels, and α denotes a per-element coefficient.

After reformulating the star-based operation described in Equation (A1), we can extend it into a combination of $\frac{(c_d+2)(c_d+1)}{2}$ distinct terms. It is worth noting that, except for the term $\alpha_{(c_d+1, c_d+1)} x^{c_d+1} x^{c_d+1}$, all other terms exhibit a nonlinear relationship with the input $\mathbf{x}$, which suggests they correspond to several independent implicit dimensions. Although we employ a computationally efficient star-based operation in the d -dimensional space, it effectively achieves feature representation in a higher-dimensional implicit feature space $\frac{(c_d+2)(c_d+1)}{2} \approx \left(\frac{c_d}{\sqrt{2}}\right)^2$ (where $c_d \gg 2$). This significantly expands the feature dimensionality without introducing any additional computational overhead in a single-layer structure. Assuming the initial network layer has a width of c_d , applying a single star-based operation yields the result shown in Expression (A1), mapping the features into a representation within the implicit feature space $R^{\left(\frac{c_d}{\sqrt{2}}\right)^2}$. Furthermore, stacking multiple such layers in a recursive manner leads to an exponential growth of the implicit dimensionality, ultimately approaching infinity. Let O_l denote the output of the l -th star-based operation; it can be expressed as follows:

$$O_l = W_{l,1}^T O_{l-1} * W_{l,2}^T O_{l-1} \in R^{\left(\frac{c_d}{\sqrt{2}}\right)^{2^l}} \quad (\text{A3})$$

In other words, by stacking l layers of the star-shaped operation, we can implicitly obtain an extremely high-dimensional representation in the feature space $\mathbb{R}$. For example, given a 10-layer isotropic network with a width of 128 per layer, the implicit feature dimension achieved through the star-shaped operation can reach approximately 90^{1024} dimensions, which is already close to being considered infinite-dimensional. Therefore, even with only a few stacked layers, the star-shaped operation is capable of expanding the implicit feature dimension exponentially.

References

1. 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; pp. 652–660.
2. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.

3. Wang, Y. DGCNN: Learning Point Cloud Representations by Dynamic Graph CNN. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2020.
4. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigi, 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; pp. 11108–11117.
5. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching efficient 3d architectures with sparse point-voxel convolution. In *Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020*; *Proceedings, Part XXVIII*; Springer: Cham, Switzerland, 2020; pp. 685–702.
6. Zhou, H.; Zhu, X.; Song, X.; Ma, Y.; Wang, Z.; Li, H.; Lin, D. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv* **2020**, arXiv:2008.01550.
7. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. Rangenet++: Fast and accurate lidar semantic segmentation. In *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 4213–4220.
8. Zhang, Y.; Zhou, Z.; David, P.; Yue, X.; Xi, Z.; Gong, B.; Foroosh, H. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020; pp. 9601–9610.
9. Liong, V.E.; Nguyen, T.N.T.; Widjaja, S.; Sharma, D.; Chong, Z.J. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. *arXiv* **2020**, arXiv:2012.04934.
10. Xu, J.; Zhang, R.; Dou, J.; Zhu, Y.; Sun, J.; Pu, S. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, QC, Canada, 10–17 October 2021; pp. 16024–16033.
11. Li, X.; Zhang, G.; Pan, H.; Wang, Z. Cpgnet: Cascade point-grid fusion network for real-time lidar semantic segmentation. In *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 11117–11123.
12. Atzmon, M.; Maron, H.; Lipman, Y. Point convolutional neural networks by extension operators. *arXiv* **2018**, arXiv:1803.10091. [\[CrossRef\]](#)
13. 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; pp. 9621–9630.
14. 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. (TOG)* **2019**, *38*, 1–12. [\[CrossRef\]](#)
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; pp. 6411–6420.
16. 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; pp. 4840–4851.
17. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. *arXiv* **2015**, arXiv:1512.03012.
18. Maturana, D.; Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 922–928.
19. Qi, C.R.; Su, H.; Nießner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016; pp. 5648–5656.
20. Choy, C.B.; Xu, D.; Gwak, J.; Chen, K.; Savarese, S. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016; pp. 628–644.
21. Riegler, G.; Osman Ulusoy, A.; Geiger, A. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 3577–3586.
22. Zhou, Y.; Tuzel, O. Voxnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
23. Wang, Z.; Lu, F. VoxSegNet: Volumetric CNNs for semantic part segmentation of 3D shapes. *IEEE Trans. Vis. Comput. Graph.* **2019**, *26*, 2919–2930. [\[CrossRef\]](#) [\[PubMed\]](#)
24. 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; pp. 3075–3084.

25. Cheng, R.; Razani, R.; Taghavi, E.; Li, E.; Liu, B. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12547–12556.
26. Ando, A.; Gidaris, S.; Bursuc, A.; Puy, G.; Boulch, A.; Marlet, R. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5240–5250.
27. Kong, L.; Liu, Y.; Chen, R.; Ma, Y.; Zhu, X.; Li, Y.; Hou, Y.; Qiao, Y.; Liu, Z. Rethinking range view representation for lidar segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 228–240.
28. Zhou, Z.; Zhang, Y.; Foroosh, H. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13194–13203.
29. Zhou, Y.; Sun, P.; Zhang, Y.; Anguelov, D.; Gao, J.; Ouyang, T.; Guo, J.; Ngiam, J.; Vasudevan, V. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In Proceedings of the Conference on Robot Learning, Virtual, 16–18 November 2020; pp. 923–932.
30. Wang, Y.; Fathi, A.; Kundu, A.; Ross, D.A.; Pantofaru, C.; Funkhouser, T.; Solomon, J. Pillar-based object detection for autonomous driving. In *Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Springer: Cham, Switzerland, 2020; pp. 18–34.
31. Zhang, F.; Fang, J.; Wah, B.; Torr, P. Deep fusionnet for point cloud semantic segmentation. In *Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Springer: Cham, Switzerland, 2020; pp. 644–663.
32. Gerdzhev, M.; Razani, R.; Taghavi, E.; Bingbing, L. Tornado-net: Multiview total variation semantic segmentation with diamond inception module. In *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 9543–9549.
33. Liu, Z.; Tang, H.; Lin, Y.; Han, S. Point-voxel cnn for efficient 3d deep learning. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
34. Ma, X.; Dai, X.; Bai, Y.; Wang, Y.; Fu, Y. Rewrite the stars. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 5694–5703.
35. Berman, M.; Triki, A.R.; Blaschko, M.B. The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4413–4421.
36. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [\[CrossRef\]](#)
37. 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.
38. Wang, P.S. Octformer: Octree-based transformers for 3d point clouds. *ACM Trans. Graph. (TOG)* **2023**, *42*, 1–11. [\[CrossRef\]](#)
39. Graham, B.; Engelcke, M.; Van Der Maaten, L. 3d semantic segmentation with submanifold sparse convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9224–9232.
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
41. Vasu, P.K.A.; Gabriel, J.; Zhu, J.; Tuzel, O.; Ranjan, A. Mobileone: An improved one millisecond mobile backbone. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7907–7917.
42. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
43. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
44. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.

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.