Transfer learning from building information model-based synthetic data for three-dimensional module detection in point clouds of modular-integrated construction hoisting

Dong Liang[a], Longyong Wu[a], Meng Sun[a], Ruibo Hu[b], Lingming Kong[a], Yipeng Pan[c], Fan Xue[a,d*]

[a] Department of Real Estate and Construction, The University of Hong Kong, Pokfulam, Hong Kong, China
[b] School of Civil and Hydraulic Engineering, Huazhong University of Science and Technology, Wuhan, China
[c] Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong, China
[d] National Center of Technology Innovation for Digital Construction Hong Kong Branch, The University of Hong Kong, Pokfulam, Hong Kong, China
* Corresponding author, Email: xuef@hku.hk, Tel: +852 3917 4174, Fax: +852 2559 9457

This is the peer-reviewed post-print version of the paper:
Liang, D, Wu, L., Sun, M., Hu, R., Kong, L., Pan, Y. & Xue, F. (2025). Transfer learning from building information model-based synthetic data for three-dimensional module detection in point clouds of modular-integrated construction hoisting. *Engineering Applications of Artificial Intelligence*, *164* (A), 113243.
Doi: 10.1016/j.engappai.2025.113243
The final version of this paper is available at: https://doi.org/10.1016/j.engappai.2025.113243.
The use of this file must follow the Creative Commons Attribution Non-Commercial No Derivatives License, as required by Elsevier's policy.

Abstract

Module hoisting monitoring is critical to the efficiency and safety of modular integrated construction (MiC). The three-dimensional point cloud is a promising data source for monitoring MiC module hoisting; however, its development was limited by the scarcity of labeled real-world data, stemming from practical constraints such as difficult collection processes and expensive annotations. To address these issues, this paper proposes a novel transfer learning paradigm using building information model (BIM) for MiC module detection in point clouds. The approach first involves pre-training of a deep learning model in a source domain using a BIM-based synthetic dataset of hoisting from other BIM projects. The model is then fine-tuned with minimal three-dimensional annotations and data augmentation. Experimental results on 2,770 frames of point clouds confirm that the transfer learning paradigm is effective, achieving a recall of 94.6% and an Average Precision at 40 Recall positions (AP_{R40}) (Intersection over Union (IoU) = 0.7) of 88.2%. This paper presents an effective transfer learning paradigm for MiC module detection in complex construction environments, addressing the challenge of limited labeled point cloud data for three-dimensional object detection across diverse construction scenarios. The findings demonstrate that BIM-to-real-world transfer learning significantly reduces the annotation burden for construction monitoring applications while maintaining high detection accuracy.

Keywords: Transfer learning; Point cloud; Modular-integrated Construction; Three-dimensional object detection; Modular-integrated Construction module detection.

1 Introduction

Modular integrated construction (MiC), a cutting-edge building technology, has recently attracted increasing attention in many economies, such as Hong Kong (Pan et al. 2021; Kong et al. 2025). MiC transforms the traditional cast-in-situ construction mode into a three-stage integration process, i.e., manufacturing in the factory, transportation, and final installation at construction sites (Zheng et al. 2020). Most MiC processes can be completed in a controlled factory environment. Meanwhile, centralized module installation replaces the complex on-site construction workflow of cast-in-situ construction. Therefore, MiC offers significant advantages in quality, productivity, and safety. However, MiC brings new challenges for on-site construction management due to the transformation of the construction process (Zhang & Pan 2021).

Hoisting, the most critical aspect of MiC module installation, has become a bottleneck in terms of on-site productivity and safety management (Zhu et al. 2022). The first reason comes from the large size, heavy weight, and unbalanced loading conditions of MiC modules (Wuni et al. 2020). Meanwhile, many MiC projects are conducted in constrained environments, which introduces further difficulties and risks to hoisting operations. In addition, blind hoisting—i.e., the operator's direct line of sight is obscured—further increase the difficulty of the operation and exacerbate the risk factor. In practice, these risk factors directly cause adverse effects on safety and productivity. For instance, collisions between modules and surrounding equipment or workers can cause severe accidents. Swing-induced instability of hoisting modules can delay precise placement and damage MiC modules. Therefore, real-time monitoring of MiC modules hoisting phase is significant for improving construction safety and productivity management (Sun et al. 2025; Zhu et al. 2024). For example, collision risks can be mitigated by detecting the distance between modules and workers or equipment (Liu et al. 2021; Chen et al. 2017). Furthermore, monitoring results can accelerate the final module positioning process, which requires high precision. Given these pressing needs, researchers have begun to explore automated module detection methods to support hoisting monitoring for both safety and productivity.

Although several advanced module detection methods have been explored in studies for safety and productivity purposes (Zhai et al. 2019), manual monitoring remains the predominant approach in current projects. The main reason lies in the fact that two mainstream monitoring

methods, i.e., smart cameras (Zheng et al. 2020) and IoT sensors (Jiang et al. 2022), have limitations in practical applications of module detection. First, smart cameras can only capture 2D information, which cannot achieve accurate spatial positioning and collision detection (Panahi et al. 2023). Second, the application of IoT sensors in complex construction environments often encounters signal vulnerability and delay, which reduces the reliability and real-time performance of hoisting monitoring (Arshad & Zayed 2024). Additionally, the use of IoT sensors is frequently criticized for long installation times and high labor costs. These limitations mean that neither approach can meet the practical requirements of real-time safety and productivity monitoring during module hoisting. Therefore, an effective module detection method must be developed urgently.

A 3D object detection method in point clouds is a potential approach for monitoring MiC hoisting (Shao et al. 2023), since it can provide rich three-dimensional information about modules. In specific, 3D object detection methods in point clouds can determine module locations, rotations, and sizes, which can be further used for safety and productivity analysis. Compared to smart cameras and IoT sensors, this method is inherently more robust, because it not only overcomes the lack of depth information in 2D vision but also avoids the signal instability and latency issues commonly faced by IoT sensor-based monitoring. The data source is typically LiDAR-based point clouds (Wu et al. 2024), which can reflect spatial information from the physical world in real-time. As a result, 3D object detection methods using point clouds are widely applied in dynamic scenarios, such as autonomous driving and robot vision (Schreier et al. 2023; Yang et al. 2018). Among those applications, end-to-end deep learning has become the mainstream architecture for 3D object detection. Significantly, the end-to-end strategy offers two key advantages: reduced manual involvement and greater adaptability to changing conditions.

However, developing a robust end-to-end deep learning model for 3D detection requires a large amount of labeled point cloud data. Generally, creating labeled 3D point cloud datasets is significantly more expensive and complex than generating image datasets (Wu et al. 2023; Li et al. 2025; Dong et al. 2024). The primary historical barrier has been the high cost of hardware. Although hardware costs have recently declined considerably (Meng et al. 2025), the overall cost of data collection in complex environments remains high. This is mainly because multiple LiDAR sources are needed to acquire comprehensive data from different viewpoints. Moreover,

data annotation in three-dimensional space is intrinsically more complex and expensive than that in two-dimensional contexts due to the additional dimension. Specifically, accurate spatial interpretation and specialized annotation tools are required for 3D data labeling (Guo et al. 2021). Therefore, end-to-end 3D object detection methods in point clouds may see wider application in complex construction scenarios if their dependence on substantial labeled datasets is reduced.

Transfer learning (TL) is a promising approach to reduce end-to-end models' dependence on labeled data (Yin et al. 2024; Li et al. 2022). TL is a machine learning technique that leverages data from related or unrelated activities to enhance the performance of a model on a designated task (Weiss et al. 2016). TL initially demonstrated its superior capability in 2D-image-based computer vision tasks. Subsequently, increasing attention has been directed toward improving model performance in 3D object detection (Wang et al. 2024) within point clouds. The foundation of TL for 3D tasks in point clouds lies in learning general 3D structures and patterns using a feature extraction mode trained on existing samples (Xiao et al. 2024). These pre-trained models can then be fine-tuned for domain-specific tasks using smaller datasets. In the construction domain, the widespread adoption of building information modeling (BIM) presents a unique opportunity to advance 3D object detection in point clouds through TL (Frías et al. 2022; Czerniawski & Leite 2019). BIM models can be converted into 3D point cloud data via virtual scanning simulations to generate domain-specific pre-training datasets that closely resemble real-world construction scenarios (Ma et al. 2020; Yang et al. 2024).

To address the challenges posed by real-time and reliable module detection for monitoring and the scarcity of labeled real-world datasets for 3D detection in point clouds, this paper presents a novel TL paradigm based on BIM for 3D module detection in point clouds during MiC hoisting. It has two specific objectives: (1) to develop a real-time and reliable 3D module detection method in point clouds for MiC hoisting monitoring; (2) to design and validate a BIM-based transfer learning paradigm that leverages synthetic pre-training to achieve accurate and efficient 3D module detection on-site hoisting scenarios. The contributions of this paper are twofold. First, to the best of our knowledge, this study is the first to train a 3D module detection model on point clouds for on-site monitoring. Second, the proposed BIM-based synthetic data-driven TL strategy enables the development of robust 3D module detection

models with significantly reduced reliance on costly and labor-intensive real-world project data labeling.

2 Literature review

2.1 On-site MiC installation process monitoring

Monitoring the on-site installation of processes is critical for on-site management, including progress control, safety management, and quality assurance. In current construction practices, manual monitoring methods are widely adopted for MiC hoisting; however, they are time-consuming and labor-intensive owing to the need to coordinate multiple work surfaces at different heights. Several automated methods have been developed to support MiC hoisting monitoring, which can be broadly categorized into IoT sensor-based and smart camera-based methods. Despite their potential, both methods face limitations in practical construction scenarios. For IoT sensor-based monitoring, Zhou et al. (2021) employed GPS sensors and smart trinity tags to enhance progress tracking and safety management during MiC installation. Additionally, Jiang et al. (2022) developed an ultra-wide-band-based system to achieve digital twinning for on-site MiC assembly. However, signal vulnerability and latency issues with IoT sensors were widely reported in these studies. Furthermore, the requirement to install individual IoT sensors on each object can negatively impact the overall productivity of the MiC hoisting process.

In terms of smart-camera-based process monitoring methods, Zhang et al. (2019) proposed a classification model capable of automatically identifying three module installation stages, i.e., hooking, hoisting, and positioning, for automated progress analysis and control. Chua and Cheah (2024) introduced a deep-learning-based method for automated construction progress monitoring in Prefabricated Prefinished Volumetric Construction (the Singaporean equivalent of MiC), using image-based window count detection and correction. In addition, Zheng et al. (2020) developed a deep learning model to detect the presence and determine the location of modules in images. These camera-based methods have proven effective for automated progress monitoring. However, due to a lack of depth information, 2D detection results are insufficient to meet other management requirements of MiC installation, such as reliable 3D collision detection and precise positioning. Therefore, exploring 3D module detection is essential for comprehensively monitoring module installation processes.

Inspired by the growing demand for 3D detection of the MiC module, Xu and Pan (2023) developed a virtual-prototyping-enabled pseudo-LiDAR point cloud dataset to evaluate the feasibility of 3D MiC module detection. While the detection results within the simulated dataset were promising, the approach has yet to be validated on real-world data. To date, no other researchers have undertaken empirical investigations using real-world construction site data to validate these simulation-based findings. This represents a significant research gap that warrants immediate attention, as the translation from controlled virtual environments to the inherently chaotic and unpredictable conditions of construction sites presents substantial methodological and practical challenges.

2.2 3D object detection and applications in construction

The advancement of 3D detection methods has been significantly driven by developments in deep learning algorithms and sensor technologies. Early approaches primarily focused on processing single-mode LiDAR-based point clouds. Existing deep learning methods for single-mode 3D detection can be broadly categorized into point-based, voxel-based, and hybrid methods. Point-based methods were among the earliest to be developed, with their core strategy being the direct processing of raw, unordered point clouds. For instance, Qi et al. (2017a) introduced PointNet, a pioneering deep learning framework for 3D object detection using point cloud data. Subsequent advancements, such as PointNet++ (Qi et al. 2017b), addressed scalability challenges by hierarchically aggregating local features, facilitating the efficient processing of large-scale scenes. Although point-based methods demonstrated strong performance in 3D detection tasks, their computational efficiency was often criticized, as the processing requirements tended to increase exponentially with increasing scene complexity. To address these limitations, voxel-based methods were developed to achieve a balance of computational efficiency and accuracy. For example, in the VoxelNet algorithm, point clouds are converted into volumetric grids during the data preprocessing stage to improve the computational efficiency (Zhou & Tuzel 2018). Although voxel-based methods have proven effective in reducing computational loads, voxelization always results in the loss of data details. To balance computational precision and efficiency, Lang et al. (2019) proposed a hybrid object detection algorithm, namely, PointPillar, which employs a novel encoder that integrates both point features and voxelization strategies.

Multi-modal fusion, such as the integration of LiDAR with thermal sensors or RGB cameras, has recently emerged as a mainstream architecture in 3D object detection. The primary advantage of multi-modal fusion is that the sensors can complement each other. Compared to single-mode LiDAR-based methods, multi-modal fusion methods have demonstrated superior performance in terms of 3D object detection accuracy. Therefore, they have been widely used in 3D object detection applications such as autonomous driving and robotic vision. For instance, a 3D object detection method that integrates LiDAR and RGB camera data improved the state-of-the-art performance on the KITTI benchmark dataset (Liang et al. 2019). However, the integration of multiple sensors inevitably leads to increased computational complexity (Li et al. 2024).

3D object detection in point clouds has served as a fundamental enabler of automation in the construction industry. This is because most automation technologies, such as robotics, rely heavily on real-time spatial information derived from 3D object detection at construction sites (Teizer et al. 2005). For example, Teizer (2008) employed a 3D range imaging-based method to detect workers within point clouds, demonstrating the feasibility of using 3D object detection for worker safety management in indoor environments. Building on this, Son et al. (2010) proposed a systematic 3D range imaging-based object detection method to support heavy equipment operations. However, the inherent low resolution of imaging sensors limited the ability of these methods to capture fine details. Furthermore, Sharif et al. (Sharif et al. 2017) introduced an automated approach for detecting 3D objects in construction point clouds but encountered challenges owing to incomplete data and the presence of highly cluttered scenes. Chen et al. (2018) evaluated several state-of-the-art 3D descriptors for construction object recognition in point clouds, considering factors such as varying levels of detail, noise, and degrees of occlusion. Limited studies are focusing on 3D object detection in point clouds for construction scenarios. Meanwhile, the majority rely on rule-based methods. Compared to end-to-end deep learning methods, rule-based methods exhibit limited adaptability to the dynamic, temporary, and often disordered nature of real-world construction environments (Liang et al. 2025). Therefore, researchers still need to invest a lot of effort to explore effective and feasible 3D object detection methods, especially end-to-end methods, in the construction industry.

The end-to-end 3D object detection model offers better adaptability to complex environments; however, developing an effective and robust end-to-end deep learning model requires a large

amount of labeled data for training (Oh et al. 2023). Two large publicly available datasets for deep learning model development in autonomous driving are Waymo (Sun et al. 2020) and Kitti (Geiger et al. 2013). However, there are no relevant datasets in the construction industry. Creating comprehensive point cloud datasets presents greater challenges than developing image datasets (Wu et al. 2023), which can be explained by several factors. First, the LiDAR sensors required for point cloud acquisition have historically been expensive. Despite the substantial decrease in LiDAR cost in recent years, the annotation process for 3D data inherently requires more sophisticated spatial comprehension and specialized tools than 2D annotation (Guo et al. 2021). Furthermore, extensive efforts are required to collect high-quality point cloud data in complex environments such as construction sites, as LiDAR sensors must be positioned at multiple locations to capture complete spatial information. Therefore, reducing reliance on the model on large-scale annotated datasets is essential for the widespread adoption of end-to-end 3D object detection in construction applications.

2.3 Transfer learning (TL) application in construction

TL, a machine learning technique, leverages the knowledge acquired by a pre-trained model on a source task for application to a related target task with limited labeled data. TL aims to improve model generalization capability in areas where acquiring labeled datasets is labor-intensive and costly (Zhao et al. 2024a; Li et al. 2025b). TL was first applied to 2D computer vision tasks. For example, several object detection models, such as Faster R-CNN, YOLO, and SSD, are pre-trained on large-scale datasets such as ImageNet or COCO and fine-tuned for specific applications (Pan & Yang 2010).

In the construction industry, TL has been applied in numerous 2D computer vision applications (Wang & Gan 2023), including safety monitoring (Fang et al. 2020), defect detection (Gong et al. 2020; Li et al. 2025a), unsafe behavior recognition (Jiang & Ding 2024), and progress tracking (Zheng et al. 2020). For safety monitoring purposes, Kolar et al. (2018) developed a pre-trained model that achieved high safety guardrail detection accuracy in a real-world dataset. Meanwhile, Chen et al. (2020) proposed a TL paradigm to detect semantic regions in site images and identify construction events using other image datasets without modification or retraining. In addition, Zoubir et al. (2022) employed three TL models by fine-tuning them with the state-of-the-art visual geometry group network to classify concrete bridge defects using limited training samples. These studies demonstrated that TL can facilitate the

development of robust and precise 2D detection models with limited construction-specific labeled datasets.

Nevertheless, the application of TL to 3D object detection in point clouds remains understudied in the construction industry, despite its demonstrated effectiveness in autonomous driving and robotic vision. Specifically, several models, such as PointNet, PointNet++, and 3DSSD (Zhao et al. 2024b; Jaquier et al. 2023), have been developed using pre-trained weights obtained from synthetic datasets such as ShapeNet for domain-specific 3D detection applications. However, no synthetic or real-world dataset currently exists for TL in the construction domain.

BIM, mandated in most current construction workflows, provides a solid foundation for developing domain-specific pre-training datasets for 3D object detection in the construction industry (Hong et al. 2020). BIMs encompass rich geometric details and semantic attributes designed to reflect real-world construction environments, including building elements and spatial relationships. Therefore, converting numerous construction scenes from existing BIMs into point cloud data presents an opportunity to create suitable pre-training materials.

In the construction industry, there were already several studies that use BIM to produce synthetic data for model pre-training for many kinds of deep learning applications, such as semantic segmentation, 2D image classification, and 2D pose recognition. In terms of semantic segmentation of point clouds, Ma et al. (2020) used synthetic BIM-based point clouds to enhance training datasets for semantic segmentation in indoor environments. A similar synthetic data-enhanced technique was also utilized by Yang et al. (2024) to improve the semantic segmentation performance for bridge structures. The experimental results confirm that synthetic data can supplement real data, especially when only a limited amount of real point clouds is available.

For 2D image classification or object detection, Frías et al. (2022) exploited a BIM object-based synthetic data generation method toward indoor point cloud classification using deep learning. Experimental results show that the 2D images generated from these synthetic 3D point clouds at different angles can be correctly classified by the 2D CNN network. Zheng et al. (2020) proposed a transfer learning-enabled 2D MiC module detection based on the images

extracted from BIMs. The results showed that the synthetic image dataset effectively improved the detection accuracy with limited real-world data.

Regarding the pose recognition or estimation, Fu et al. (2025) introduced the synthetic data from a Unity-based BIM project to enhance the 3D pose recognition of construction workers in the real world. Experiments verify that the model jointly trained with synthetic and real data outperforms a model trained on real data alone. Similarly, Pham & Han (2024) used the synthetic data to improve the model performance for excavator pose estimation.

However, there is no study for enhancing 3D object detection performance by using BIM-based synthetic data in the construction industry. Therefore, more efforts should be put into this part in the future since a 3D real-world dataset is more difficult to collect than 2D images.

In summary, monitoring the module hoisting process is crucial for on-site MiC installation in terms of efficiency and safety. However, existing IoT sensor-based and smart camera-based methods are inadequate for monitoring this process. Inspired by its success in autonomous driving and robotic vision, 3D object detection in point clouds is a promising approach for monitoring MiC module hoisting. Nevertheless, the scarcity of labeled data remains a significant obstacle to applying end-to-end 3D object detection methods in construction. TL can reduce the dependence of these methods on labeled datasets. Furthermore, BIM provides a valuable resource for creating pre-trained datasets for 3D detection in construction scenarios.

3 Methodology

3.1 Overview

Fig. 1 shows the proposed TL paradigm, which leverages the target domain (real-world scenario) and the source domain (BIM project). The model development process consists of two domains:

(1) Source domain: A BIM project of MiC is developed in Unity to create a foundation of synthetic data. The project is subsequently converted into labeled point cloud datasets through virtual simulation. These labeled data are then used to train a PointRCNN model. A pre-trained model (Result 1) is obtained through this comprehensive processing pipeline and serves as the foundation for TL to the target domain.

(2) Target domain: Point cloud data are collected in real-world construction environments. The collected raw point cloud data are then preprocessed and labeled. A small portion of the labeled data is selected as a training set, while the remainder is used as a test set. Data augmentation is applied to the training set to enhance model performance. Finally, the pre-trained source domain model is fine-tuned using real-world data within the same PointRCNN architecture. The proposed TL paradigm can quickly adapt to real-world construction environments without extensive manual labeling.

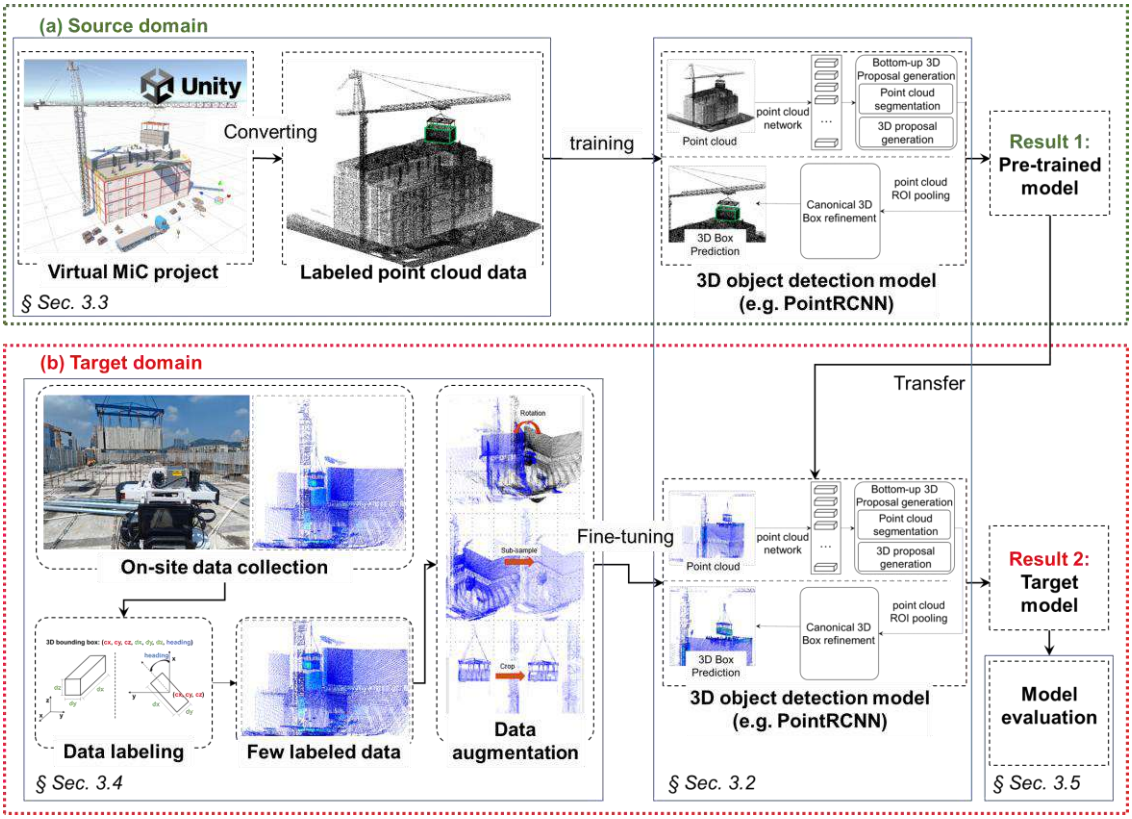


Fig. 1 Proposed transfer learning paradigm

3.2 3D object detection model

In general, the deep learning architecture in the presented TL paradigm can utilize any 3D object detection algorithms. The PointRCNN algorithm (Shi et al. 2019), a point-based 3D detection method, is selected as a benchmark model here. More algorithms will be tested in the sensitivity analysis in Section 4.3. PointRCNN directly processes raw point clouds, allowing for the retention of more original point cloud information and achieving higher detection accuracy. Therefore, this paper adopts the PointRCNN algorithm as the deep learning architecture. The two-stage architecture of the PointRCNN algorithm is shown in Fig. 2.

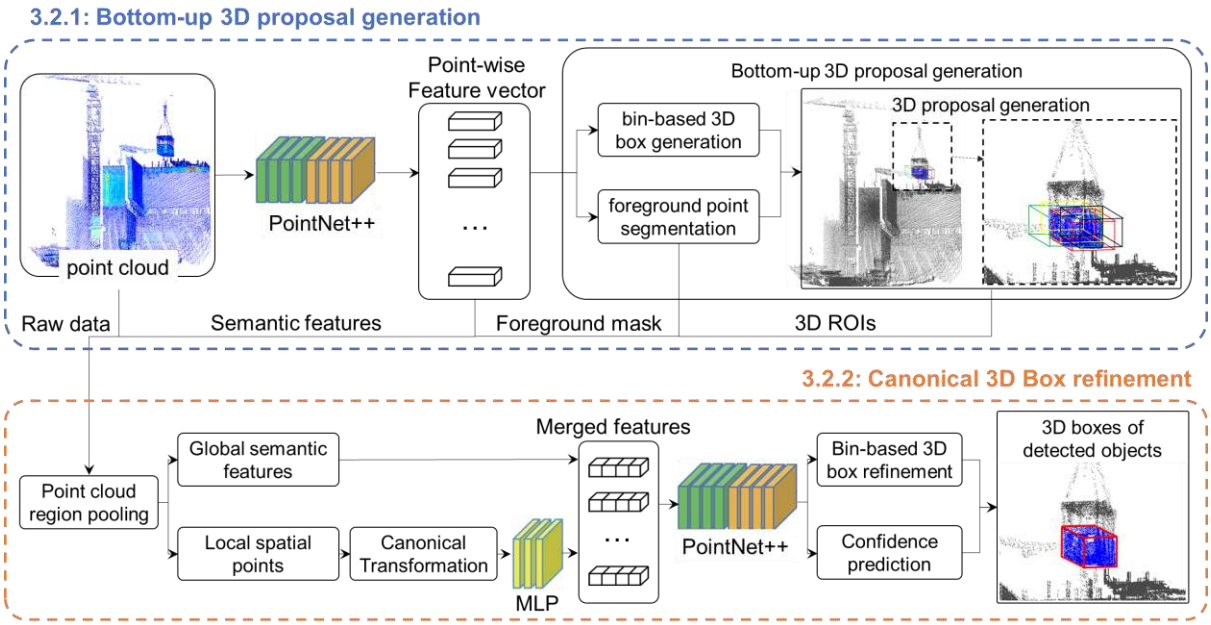


Fig. 2 Architecture of PointRCNN for 3D object detection from point cloud

3.2.1 Bottom-up 3D Proposal Generation

The bottom-up 3D proposal generation stage of PointRCNN is a carefully structured process designed to directly extract high-quality 3D proposals from raw point cloud data. This stage begins with the segmentation of the entire point cloud scene into foreground and background points. PointNet++ is employed as a backbone architecture to extract distinctive point-wise features. These extracted features serve as the foundation for subsequent operations.

A segmentation head is subsequently integrated into the backbone network to estimate a foreground mask, which effectively identifies objects in regions of interest. This segmentation procedure is critical for detecting possible objects and guiding the generation of accurate 3D proposals.

Parallel to the identification of foreground points, a bin-based box regression head is incorporated to generate 3D bounding box proposals. These proposals are characterized by parameters such as the center coordinates (cx, cy, cz), object dimensions (dx, dy, dz), and the heading angle representing orientation in the top view. To ensure robust and precise proposal generation, a bin-based regression loss is employed. This loss function discretizes the potential values of the bounding box parameters into predefined bins, enhancing learning through the integration of bin classification and residual regression within each bin. For localization, bin-

based classification with cross-entropy loss is applied to the X- and Z-axes, while the Y-axis, owing to its limited range of vertical fluctuation, utilizes a smooth L1 loss. The orientation and size estimations are addressed using bin-based techniques. This approach significantly improves the proposal generation efficiency by limiting the search space and expediting convergence during training.

3.2.2 Canonical 3D Box Refinement

The canonical 3D box refinement stage, a critical component of PointRCNN, enhances the proposals generated in the first stage to achieve state-of-the-art detection accuracy. Once the 3D bounding box proposals are obtained, a point cloud region pooling operation is performed for each proposal. This operation focuses on the region corresponding to each proposal and aggregates the learned point representations from the bottom-up 3D proposal generation stage.

The pooled points are transformed into a canonical coordinate system to enhance the capture of local spatial attributes. This transformation standardizes the spatial relationships within each proposal, thereby improving the network's ability to learn invariant features. The canonical coordinate system is defined with its origin at the center of the proposal. The X'- and Z'-axes are aligned parallel to the ground plane, while the Y'-axis remains consistent with the original LiDAR coordinate system.

The refining procedure leverages both the modified local spatial characteristics and the global semantic information obtained during the bottom-up 3D proposal generation phase. These characteristics are concatenated and input into a dedicated network head designed to refine the initial proposals. The network head comprises a sequence of fully connected layers that encode the local characteristics, which are subsequently integrated with global semantic data. This integration enables the network to retain depth awareness while focusing on the contextual details of each proposal. The refinement process involves predicting residuals for the bounding box parameters and refining the confidence scores associated with each proposal. Finally, candidate proposals that achieve a 3D intersection over union (IoU) score greater than 0.55 relative to their corresponding ground-truth boxes are classified as positive samples.

In conclusion, the PointRCNN algorithm can directly process raw point cloud data, distinguishing it from other 3D object detection algorithms. The direct extraction of spatial

information helps preserve the integrity of the data by avoiding information loss during conversion. The algorithm employs a bottom-up approach to generate high-quality 3D candidate proposals, which significantly narrows the search space. In the second stage, PointRCNN refines these proposals through precise bounding box adjustments. Owing to its capability for accurate spatial feature extraction and localization, PointRCNN is particularly well-suited for complex three-dimensional environments, such as construction sites.

3.3 Source domain

3.3.1 BIM project of MiC

A BIM of the MiC project was developed in Unity to generate synthetic point cloud data. Currently, there are no publicly available BIM models specifically tailored for MiC projects. To address this gap, it is essential to retrieve and adapt BIM models from traditional construction scenarios for integration into Unity. Recyclable elements within these BIM models include buildings, tower cranes, vehicles, workers, and stacked materials. Furthermore, standard-sized MiC module models must be designed and imported into Unity. The subsequent step involves designing and animating the hoisting process of MiC modules within the Unity environment. This animation incorporates both translational and rotational movements of the modules, ensuring a realistic representation of the construction process.

Notably, if a target MiC project possesses a site-specific BIM model from the early stages, this model can serve as a direct foundational source for synthetic data generation. The site-specific BIM will significantly bridge the gap between domains, since the geometric properties and construction environment in synthetic data can maintain high consistency with real-world scenarios. However, this study adopts a more general strategy to validate the effectiveness of the proposed framework even without site-specific BIMs. Future applications can readily integrate site-specific BIMs into the pipeline to enhance transfer capability in practice.

3.3.2 Synthetic data generation from the BIM project

Fig. 3 outlines a comprehensive framework for reconstructing 3D point clouds from the geometric data provided by a BIM project in Unity:

a) The process begins by configuring a camera in Unity to capture depth information, which is represented by normalized values between 0 and 1. Specifically, a value of 0 corresponds to the near plane, and 1 represents the far plane.

b) Using the depth map, sampling is performed to get pixel coordinates (x, y) and the corresponding depth value ($D_{(0\text{-}1)}$).

c) Pixel coordinates (x, y) are converted into normalized UV coordinates ($x_{(0\text{-}1)}$, $y_{(0\text{-}1)}$) by dividing the coordinates by the texture dimensions (width×height). The depth value ($D_{(0\text{-}1)}$) is also adjusted to match the new pixel coordinates (x, y).

d) These UV coordinates are further transformed into normalized device coordinates (NDC) by remapping them from the range [0,1] to [-1,1].

e) The NDC position is subsequently transformed into view (camera) space by applying the inverse projection matrix.

f) The view space position is further converted to world space by applying the camera-to-world transformation matrix and normalizing by the homogeneous component. Invalid depth values (above or below the acceptable range) are identified and marked as background or invalid pixels.

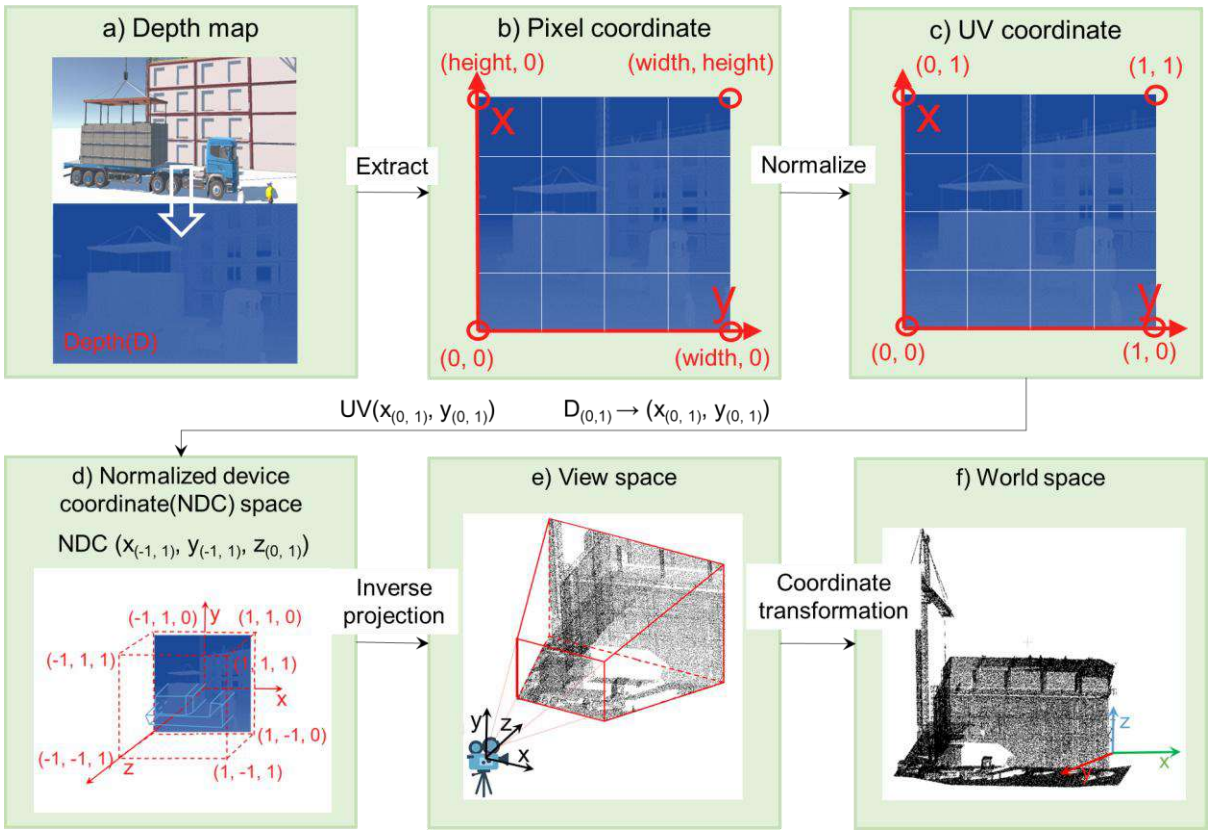


Fig. 3 Framework of obtaining 3D point cloud from Unity project

Table 1 presents the mathematical foundation for the coordinate transformations shown in Fig. 3 (steps a–f) (Hartley & Zisserman 2003; Xu & Pan 2023). The process begins with the Unity-

generated depth map (a). Next, pixel indices are defined in the image coordinate system (b). The pixel indices are then normalized into UV coordinates (c). Following this, the UV coordinates are remapped into NDC (d). NDC represents the standard [−1,1] cube used in computer graphics. Subsequently, the inverse projection matrix is applied to recover the NDC points in camera space (e). Finally, the camera's rotation matrix and translation vector transform the points from camera space into the world coordinate system (f).

Table 1 Coordinate transformations from depth map to world space

Step	Coordinate System	Point expression
a) Depth Map	Depth buffer output from Unity. Each pixel (x, y) has a normalized depth value $D \in [0,1]$ (D: Normalized depth value output by Unity; $D = 0$ corresponds to the near plane; $D = 1$ to the far plane)	
b) Pixel coordinate	Pixel indices in the image plane of resolution (w, h) (x_{pix}, y_{pix}: Pixel coordinates in the image plane (in pixels); w, h: Image width and height (in pixels))	$(x_{pix}, y_{pix}) \in [0, w] \times [0, h]$
c) UV coordinate	Pixel indices normalized to [0,1] (u, v: Normalized UV coordinates, ranging from 0 to 1)	$u = x_{pix}/w$ $v = y_{pix}/h$
d) NDC	UV coordinates mapped to [−1,1] (x_{ndc}, y_{ndc}, z_{ndc}: Normalized device coordinates (NDC), ranging from [−1,1] for x, y and [-1,1] for z)	$x_{ndc} = 2u - 1$ $y_{ndc} = 2v - 1$ $z_{ndc} = 2D - 1$
e) View (camera) space	Apply the inverse projection matrix P^{-1} to recover a 3D point in camera coordinates (p_{view}: Point in the camera (view) coordinate system; P: Camera projection matrix defined by the intrinsic parameters (field of view, aspect ratio, near and far clipping planes))	$p_{view}=P^{-1}\cdot(x_{ndc}, y_{ndc}, z_{ndc},1)^T$ $p_{view}=(X_v, Y_v, Z_v,)$
f) World space	Transform from camera space to global coordinates using extrinsics (p_{world} : Point in the global world coordinate system; R: Camera rotation matrix (3×3); t: Camera translation vector (3×1))	$p_{world}=R\cdot p_{view} + t$

Finally, continuous frames of construction video are captured by each camera, generating multiple depth map frames. These frames of depth maps are converted into point clouds within a unified world coordinate system. To ensure complete coverage of the construction scene, camera captures from different positions are combined during the experiment.

During the conversion process of point clouds, it is essential to record the geometric dimension and 3D position of the MiC module as corresponding labels. These labels are saved in a txt file, formatted as shown in Fig. 4. In Fig. 4, c_x, c_y, and c_z represent the 3D coordinates of the center of the MiC module. Meanwhile, d_x, d_y, and d_z denote the geometric dimensions of the object's 3D bounding box along the x-, y-, and z-axes, respectively, when the heading angle is 0 degrees. The heading refers to the orientation angle of the object in the top view, where a 0-degree angle aligns with the x-axis and increases counterclockwise from the x-axis toward the y-axis.

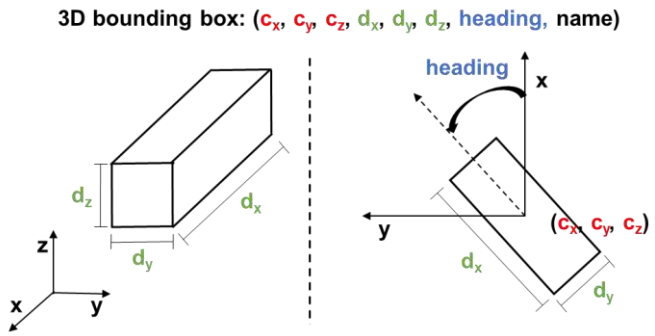


Fig. 4 Label format of 3D bounding box

3.4 Target domain

3.4.1 Real-world data curation

To obtain valid point cloud data for model training and validation, three key factors must be considered: the sensing range of the devices, area of interest, and prospective locations of the sensing devices (Liang et al. 2024). The fundamental principle is to strategically position an optimal number of sensors within the deployable equipment range for maximized region coverage. As a result, sensor location configurations will vary depending on the specific scenario.

The collected raw point cloud data are manually labeled with an open-source 3D point cloud annotation tool, i.e., SUSTechPOINTS (Li et al. 2020). The format is the same as that illustrated in Fig. 4. The primary labeling principle is to adhere to the original size of the MiC module, even when certain parts are missing due to occlusion. The labeling process required approximately 30–60 seconds per frame, depending on the complexity and degree of occlusion, and was conducted by two trained graduate research assistants. A third assistant is responsible for cross-checking the labeled frames.

3.4.2 Data augmentation

As shown in Fig. 5, three data augmentation strategies, i.e., rotation, downsampling, and cropping, are applied to the training sets to enhance the robustness and generalization ability of the model (Zhu et al. 2024). First, as shown in Fig. 5(a), random rotations are applied to the point cloud data around the vertical axis. This rotation helps the model develop rotational invariance, allowing it to recognize objects regardless of their orientation in the horizontal plane. Second, random subsampling of the original point cloud is employed to simulate varying

point densities. Downsampling, as a regularization technique, mitigates the risk of overfitting by preventing the model from becoming too reliant on specific point distributions. Third, as shown in Fig. 5(c), random cropping is performed within the 3D bounding box. This technique aims to replicate partial occlusions and incomplete object views, which are commonly encountered in real-world situations. These augmentation strategies are applied throughout the training phase with varying probabilities, contributing to a more diverse and comprehensive training dataset.

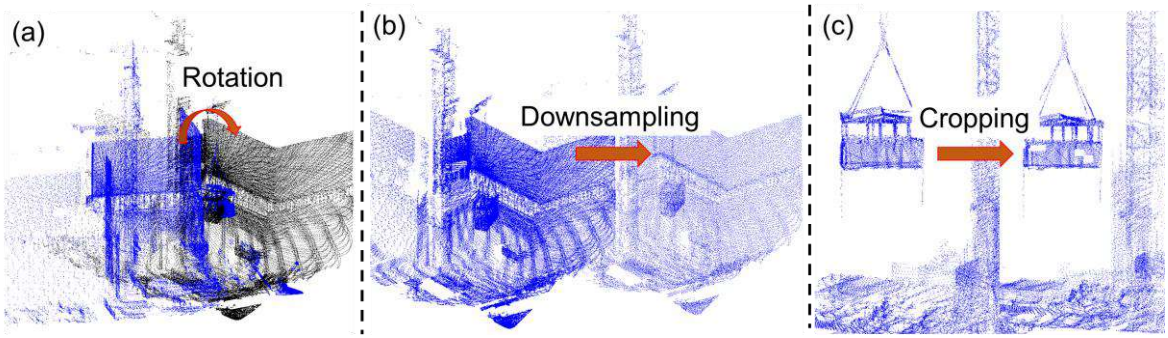


Fig. 5 Data augmentation strategies

3.5 Model evaluation

Two key metrics are selected to evaluate the performance of the TL model: recall and average precision (AP_{R40}), both of which are commonly used in related studies (Qian et al. 2022).

a) Recall

Recall quantifies the proportion of ground truth objects correctly detected by the model. It is defined as:

$$Recall = TP / (TP + FN), \quad (1)$$

where *TP* (True Positives) refers to the correctly detected objects that meet the 3D IoU threshold, and *FN* (False Negatives) refers to the ground truth objects that were not detected. The 3D IoU is denoted as:

$$IoU = Volume_{overlap} / (Volume_g + Volume_{pred} - Volume_{overlap}), \quad (2)$$

where $Volume_g$ indicates the volume of the 3D bounding box annotated by humans (ground truth), $Volume_{pred}$ indicates the volume of the predicted 3D bounding box, and $Volume_{overlap}$ indicates the overlapping volume between $Volume_g$ and $Volume_{pred}$. In existing 3D detection tasks, a 70% IoU threshold is typically used for large objects such as vehicles, while a 50% IoU threshold is applied to smaller dynamic objects such as pedestrians. This paper adopts a 70% IoU threshold for detecting MiC modules, aligning with established evaluation standards for vehicle-sized targets. This threshold ensures consistent benchmarking with traditional 3D

detection frameworks while maintaining robustness for object categories that require precise localization.

b) AP_{R40}

Fig. 6 illustrates the precision–recall curve and Average Precision (AP, or area under the curve, AUC) metrics. AP measures the area under the precision–recall curve and provides a comprehensive evaluation of a model's ability to balance precision and recall. AP is computed as:

$$AP = \Sigma_N (R_i - R_{i-1}) P_i \tag{3}$$

where N denotes the total number of sampling points on the precision-recall curve before the threshold, i ($N \geq i > 0$) is the index of sampling points before the threshold, R_i denotes the recall at the i-th sampling point, and P_i denotes the precision at recall level R_i. The summation in Eq. (3) integrates precision over different recall levels. AP is a measure of the average precision across the precision–recall curve, with sampling at 11 recall points (0.1 intervals). AP_{R40} refines this by using 40 recall points (0.025 intervals), offering denser sampling for a stricter evaluation. AP_{R40} is better suited to capture performance nuances, particularly at high recall levels. In current practice, AP_{R40} is standard in 3D detection tasks (e.g., KITTI, nuScenes), while AP is more commonly used in simpler 2D tasks. By measuring both AP_{R40} and Recall, we can effectively assess the accuracy and completeness of a 3D detection model's predictions.

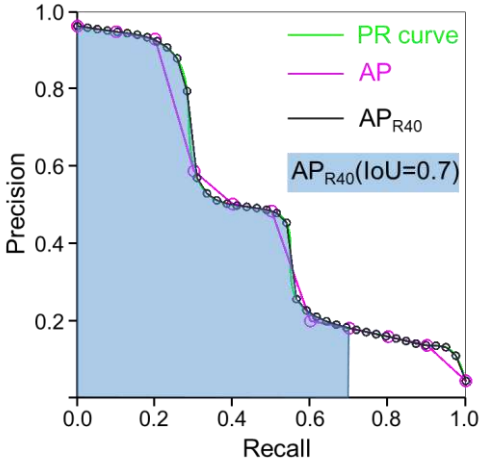


Fig. 6 Precision-Recall curve and AP metrics

4 Experiment

4.1 Experimental settings

4.1.1 Synthetic data preparation

As shown in Fig. 7, a single BIM-based MiC project is constructed in Unity to generate synthetic datasets for hoisting operations. To simulate a realistic construction site, the BIM environment incorporates several detailed elements: a) an MiC module with a top steel hoisting frame, b) a tower crane, c) a multi-story building structure, d) a truck, e) safety barriers, f) construction materials and equipment on the ground and working levels, and g) workers. The MiC module was created by the authors, and other related assets were sourced from an open-source Unity project for construction scenarios (Ding & Luo 2024). Two hoisting trajectories were animated with both translational and rotational movements. Compared with the case project in Shenzhen, the BIM project assumed a generic multi-story structure with simplified facade details and different floor heights.

Furthermore, nine cameras with various viewpoints were positioned to capture the movement of the MiC module throughout the scene. Specifically, three cameras were located on the ground level (G1-G2-G3), three at the mid-level of the building (M1-M2-M3), and three on the working level (R1-R2-R3). All cameras were oriented toward the center of the building to comprehensively capture the motion trajectory of the MiC module. Compared with the LiDAR scanning in a real-world project, the synthetic data generation from virtual depth maps shows differences in sampling density and occlusion patterns, which can influence the transfer effectiveness of the pre-trained model.

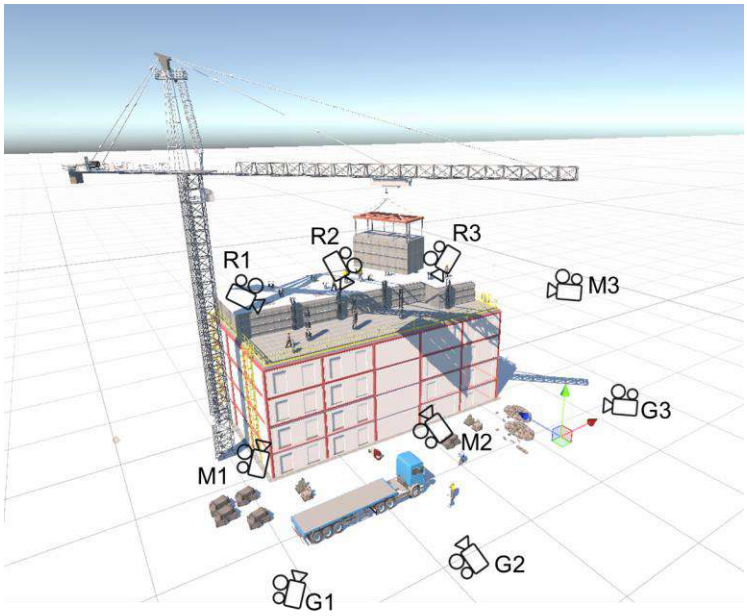


Fig. 7 Virtual MiC project in Unity

Fig. 8 shows two distinct hoisting processes designed and simulated within the BIM environment. These processes involve intricate operations, including both translational movement and rotation, from the starting (S) position to the ending (E) position on the building's working floor. Two dynamic scenarios are captured in accordance with the method outlined in Section 3.2. Each camera captures a total of 101 frames for scenario 1 and 113 frames for scenario 2. The point clouds from different sensors are merged to ensure that the MiC module is consistently represented in every frame. A total of fourteen combinations of point clouds from different sensors were designed, with the design principle ensuring that at least one camera is positioned on the ground, mid-level, and working level. Finally, 2996 point cloud frames were generated from the two scenarios.

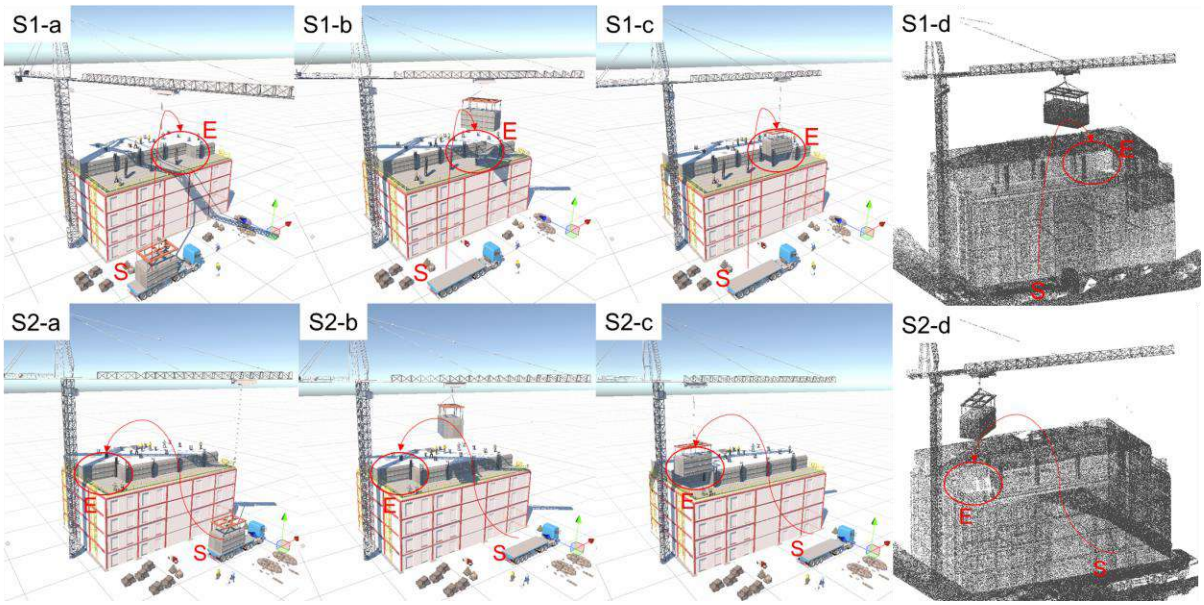


Fig. 8 Two scenarios of MiC project in Unity

4.1.2 Real-world Data Preparation

A real-world experiment was conducted to validate the proposed method. The test project was a thirty-three-story government sector building in Shenzhen, mainland China. The project employed a traditional cast-in-situ reinforced concrete structure below the fourth floor and a combination of MiC and cast-in-situ shear walls above the fourth floor. Data collection occurred on September 26, 2024, during the hoisting of the MiC module to the eighth floor. Fig. 9 shows the MiC module's status prior to hoisting, the hoisting process itself, and the subsequent placement process.

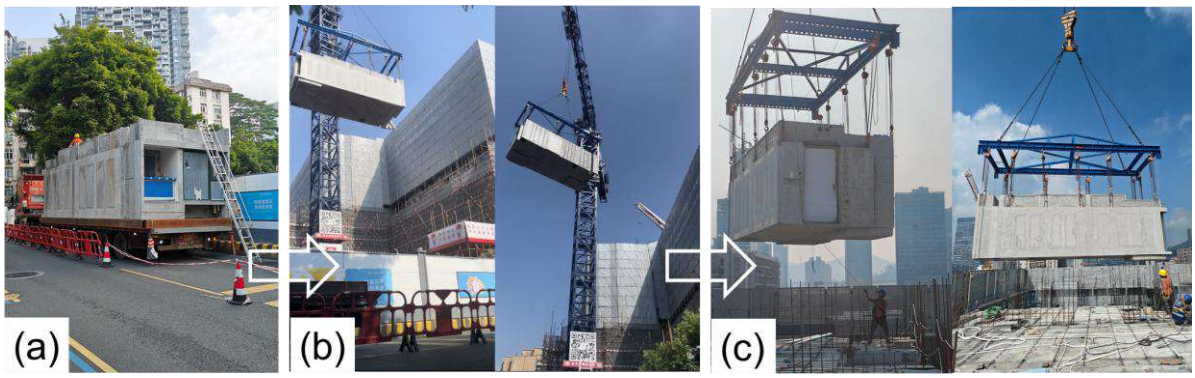


Fig. 9 On-site MiC module building process. (a) The MiC module before hoisting; (b) hoisting process, and (c) placing process

Fig. 10 illustrates the integrated point-cloud collection device designed for this paper. The device comprises four key components: a LiDAR sensor, a NUC mini computer, a battery, and

a touchscreen interface. Two models of LiDAR sensors were selected for use: the Livox Avia and the Livox Mid-70. The Livox Mid-70 offers a 70.4° circular field of view, a minimum detection range of 5 cm, a maximum detection range of 260 m, and a range precision of 2 cm. The Livox Avia, on the other hand, has a 70.4° by 77.2° circular field of view, a maximum detection range of 450 m, and a range precision of 2 cm. The NUC mini computer is connected to the LiDAR sensor for the acquisition, pre-processing, and storage of point cloud data. The battery was specifically designed to ensure compatibility with the on-site equipment, taking into consideration the differences in voltage and power stability between the site's electrical system and typical residential systems.

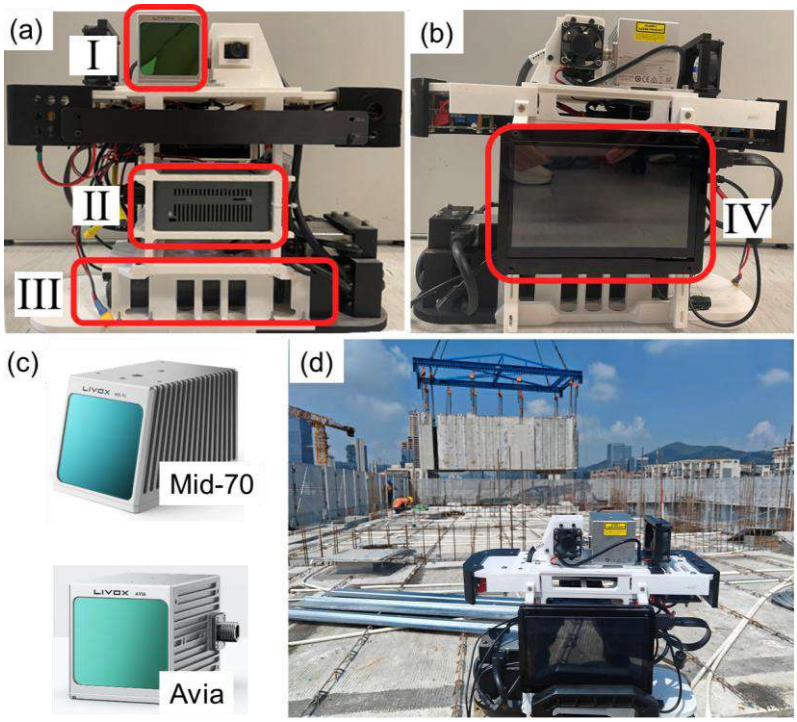


Fig. 10 Point cloud collection device in this paper. (a) Front of the device: (I) LiDAR sensor, (II) NUC mini computer, (III) Battery; (b) Back of device: (IV) Touchscreen; (c) Livox Mid-70 and Avia; (d) Installation illustration of point cloud collection device on site

In accordance with the sensor placement principles outlined in Section 3.4.1, Fig. 11 illustrates the strategic positioning of one Livox Mid-70 and two Livox Avia sensors to ensure comprehensive coverage of the potential hoisting area for the MiC module. The Livox Mid-70 (LiDAR 1) is positioned on the ground floor to monitor the area where the MiC modules are initially hoisted from the truck. One Livox Avia 70 (LiDAR 2) is placed on a terrace on the eighth floor of a neighboring building (Building A in Fig. 11) to capture the intermediate stage of the hoisting process. The second Livox Avia (LiDAR 3) is located on the working level of

the building under construction (Building B in Fig. 11) to cover the area above the working level.

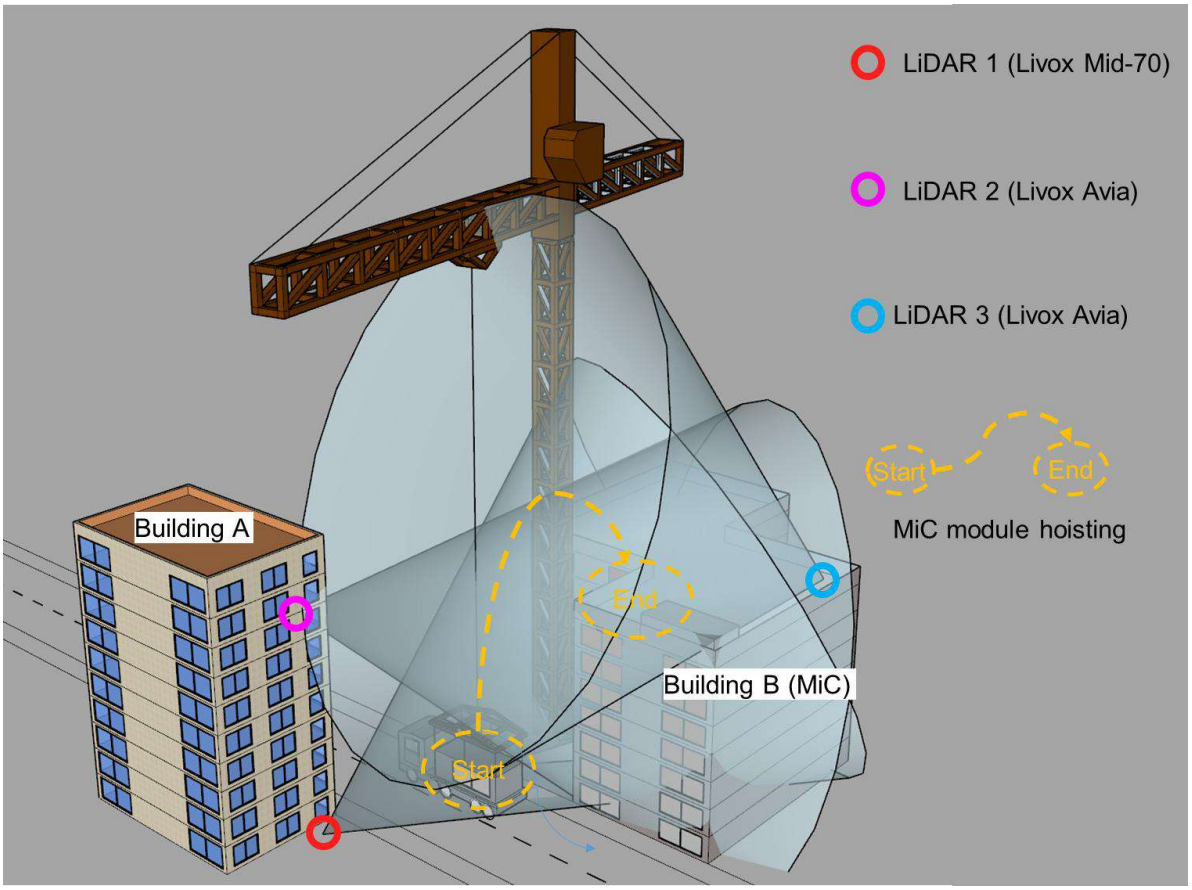
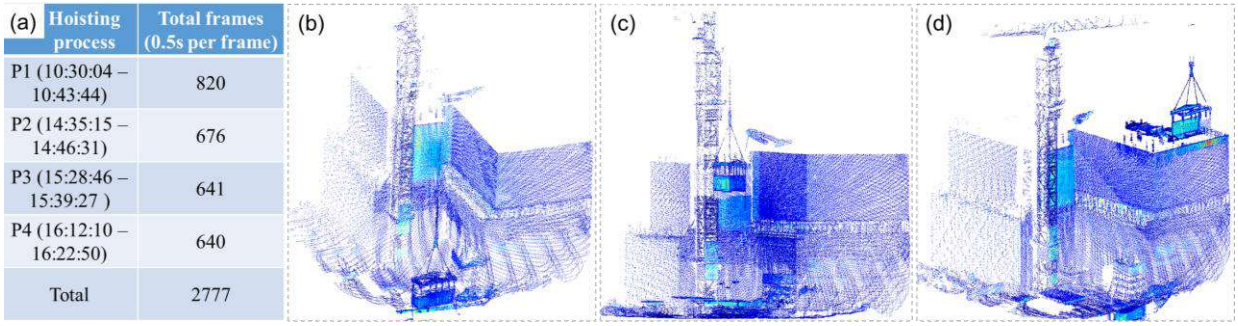


Fig. 11 Sensor placement scheme illustration

In this paper, four real-world hoisting processes were recorded. Fig. 12(a) displays the time intervals and corresponding frame numbers for each of four hoisting processes (P1–P4). The interval between each frame was 0.5 s, resulting in a total of 2777 frames. Figs. 12(b), (c), and (d) illustrate the example scenes at three module installation stages, i.e., hooking, hoisting, and positioning. The hooking stage primarily captures the initial connection of the hoisting equipment to the module. The hoisting stage refers to the vertical and horizontal movement of the module in the workspace. The final positioning stage involves the fine alignment and placement of the module in its designated location.



(a) Hoisting process	Total frames (0.5s per frame)
P1 (10:30:04 – 10:43:44)	820
P2 (14:35:15 – 14:46:31)	676
P3 (15:28:46 – 15:39:27)	641
P4 (16:12:10 – 16:22:50)	640
Total	2777

Fig. 12 Point cloud data of MiC hoisting process: a) Summarization; b) Initial stage of hoisting; c) Mid process of hoisting; d) Final process of hoisting

4.1.3 Test cases

The proposed TL model was trained and tested on a cloud server equipped with an Nvidia L40S GPU. The experiments were conducted using the *OpenPCDet* library (ver. 0.6.0 (2020)), which includes various 3D detection algorithms. The training process for the PointRCNN algorithm employed the Adam OneCycle optimizer for learning rate adjustment. The learning rate, weight decay, and momentum were set to 0.01, 0.01, and 0.9, respectively. A batch size of 1 was used, and the maximum number of epochs was set to 300. The minimum confidence threshold for detections was set to 0.95. Additionally, the mean size of the bounding box was defined as (9.2 m, 3.7 m, 3.4 m).

.

The pre-trained model was developed using the 2,996 frames of synthetic data described in Section 4.1.1. The training and test sets were randomly divided in an 8:2 ratio, resulting in 2397 training frames and 599 test frames.

As shown in Table 2, three baseline models without any processing procedures were first designed to establish reference performance. In addition, five standard test cases were designed to assess the effectiveness of TL from BIM-based synthetic data for 3D module detection in point clouds of MiC hoisting.

For the baseline models' tests, the complete dataset from the P1 process (820 frames) was used for training, while the remaining frames from the three other processes (P2/P3/P4, 1957 frames) were reserved for testing. Meanwhile, no pre-training or data augmentation was used. Notably, three different 3D detection algorithms, i.e., PointRCNN, PartA2, and Second, were used for three baseline model tests (including PointR.-P1-Full(Ref.), PartA2.-P1-Ful, and Second.-P1-

Full), respectively. It is worth mentioning that the PointRCNN algorithm achieved the best performance after preliminary computation, as shown in Table 2. Therefore, PointR.-P1-Full was set as a reference case to facilitate the design of subsequent cases.

Based on the reference case, the subsequent five standard test cases explored various strategies to minimize the training data requirements. The progression of the test cases follows a logical sequence:

- PointR.-P1-7: Training data was minimized to only seven frames;
- PointR.-P1-7-Aug: Data augmentation was applied to compensate for the limited data;
- PointR.-P0-Pre: Augmentation techniques were implemented to mitigate data scarcity;
- PointR.-P1-7-Pre: A pre-trained model was fine-tuned using just seven frames of data from the P1 process;
- PointR.-P1-7-Aug-Pre: Integrating all methodologies (pre-trained model, limited P1 data, and data augmentation) was used.

This experimental design effectively evaluates the trade-offs between the volume of training data and model performance while investigating the potential of TL and augmentation techniques to reduce reliance on labeled training data.

Table 2 Details of eight test cases

Type	Dataset	Data augmentation	Pretrained model	Code
Baseline test	Training: P1(820 frames) - Test: P2/P3/P4(1,957 frames)	False	False	PointR.-P1-Full(Ref.)
	Training: P1(820 frames) - Test: P2/P3/P4(1,957 frames)	False	False	PartA2.-P1-Full
	Training: P1(820 frames) - Test: P2/P3/P4(1,957 frames)	False	False	Second.-P1-Full
Standard test	Training: P1(7 frames) - Test: P1/P2/P2/P3(2,770 frames)	False	False	PointR.-P1-7
	Training: P1(7 frames) - Test: P1/P2/P2/P3(2,770 frames)	True	False	PointR.-P1-7-Aug
	No training - Test: P1/P2/P3/P4(2,777 frames)	False	True	PointR.-P0-Pre
	Training: P1(7 frames) - Test: P1/P2/P2/P3(2,770 frames)	False	True	PointR.-P1-7-Pre
	Training: P1(7 frames) - Test: P1/P2/P2/P3(2,770 frames)	True	True	PointR.-P1-7-Aug-Pre

Table 3 presents four groups of test cases designed to evaluate key parameters influencing TL and data augmentation strategies. For the first three groups, the reference case PointR.-P1-7-Pre used the PointRCNN algorithm, with seven frames from P1 for training, without data augmentation.

Table 3 Details of eleven sensitivity test cases

Category	Dataset splitting	Data augmentation	Algorithm	Code
	Training: P1(7 frames) – Test: P2/P3/P4(2770 frames)	False	PointRCNN	PointR.-P1-7-Pre (Ref.)

3D object detection algorithm	Training: P1(7 frames) – Test: P2/P3/P4(2770 frames)	False	PartA2	PartA-P1-7-Pre
	Training: P1(7 frames) – Test: P2/P3/P4(2770 frames)	False	Second	Second-P1-7-Pre
Training data number	Training: P1(4 frames) – Test: P2/P3/P4(2773 frames)	False	PointRCNN	PointR.-P1-4-Pre
	Training: P1(13 frames) – Test: P2/P3/P4(2764 frames)	False	PointRCNN	PointR.-P1-13-Pre
Dataset selection	Training: P2(7 frames) – Test: P1/P3/P4(2770 frames)	False	PointRCNN	PointR.-P2-7-Pre
	Training: P3(7 frames) – Test: P1/P2/P4(2770 frames)	False	PointRCNN	PointR.-P3-7-Pre
	Training: P4(7 frames) – Test: P1/P2/P3(2770 frames)	False	PointRCNN	PointR.-P4-7-Pre
Data augmentation strategy	Training: P1(7 frames) – Test: P1/P2/P3/P4(2,700 frames)	False	PointRCNN	PointR.-P1-7
	Training: P1(7 frames) – Test: P1/P2/P3/P4(2,700 frames)	Cropping	PointRCNN	PointR.-P1-7-Crop
	Training: P1(7 frames) – Test: P1/P2/P3/P4(2,700 frames)	Rotation	PointRCNN	PointR.-P1-7-Rot
	Training: P1(7 frames) – Test: P1/P2/P3/P4(2,700 frames)	Sampling	PointRCNN	PointR.-P1-7-Samp
	Training: P1(7 frames) – Test: P1/P2/P3/P4(2,700 frames)	Combination	PointRCNN	PointR.-P1-7-Aug

The first group examines the impact of different 3D object detection algorithms; the comparisons are between PointRCNN and PartA2 or Second, while the other parameters remain unchanged. According to the classification principle of point cloud processing algorithms, PointRCNN is a point-based algorithm, while the Second and partA2 are both voxel-based types. The most widely adopted algorithms are selected from each of the two algorithm types. Meanwhile, the comparison of the three algorithms can not only show the performance differences between the algorithms, but also compare the performance of different point cloud processing algorithm types in this task.

The second group evaluates the effect of training data size by varying the number of frames used for training (4, 7, and 13 frames, corresponding to intervals of 64, 128, and 256, respectively, in P1). The third group assesses dataset selection by training models on different datasets (P1, P2, P3, and P4). The final group focuses on the influence of data augmentation strategies, including cropping, rotation, sampling, and various combinations thereof.

4.2 Experiment results

The experimental results demonstrate the effectiveness of the pre-trained model, which was developed using 2996 frames of synthetic data. The model achieved optimal performance at epoch 188, with an AP_{R40} (IoU = 0.7) of 75.1% and a recall of 59%. The AP_{R40} (IoU = 0.7) score of 75.1% reflects strong precision, while the recall value of 59% indicates that the model

successfully identifies approximately 59% of all relevant instances in the test set. These results provide a solid foundation for further fine-tuning and adaptation to real-world applications. Fig. 13 displays three sample prediction results from the pre-trained model.

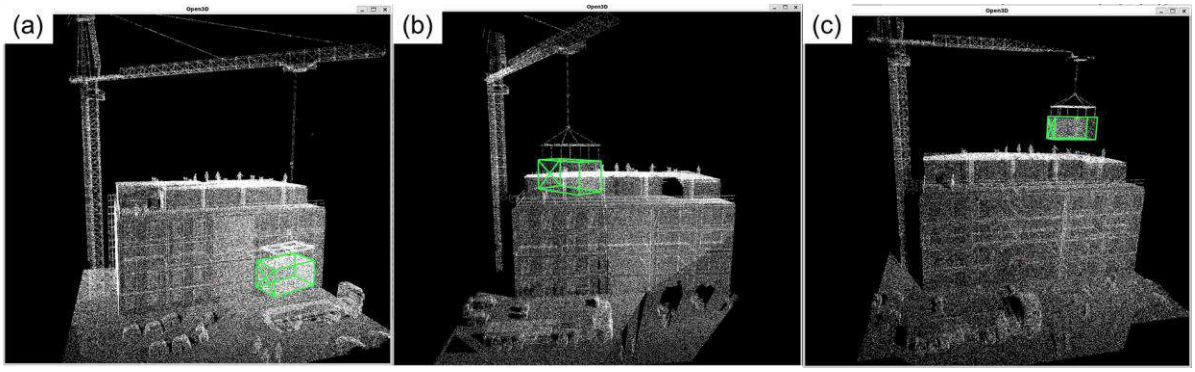


Fig. 13 Example prediction result of pre-trained model. (a) initial-stage hoisting, (b) mid-stage hoisting, and (c) final-stage hoisting.

The module detection results of the three baseline tests are listed in Table 4. The first row of PointR.-P1-Full case achieved the highest performance with a 98.1% recall and 94.9% AP_{R40} (IoU = 0.7). The result demonstrated that comprehensive training data (820 frames) yielded very high performance, albeit at the cost of significant training time cost (21 h and 28 min). In contrast, the PartA2.-P1-Full case achieved the next best performance, with 74.5% recall and 74.5% AP_{R40}, with a shorter training time of 15 h and 14 min and a faster inference at 0.082 s per frame. On the other hand, the Second.-P1-Full case exhibited the weakest detection capability, with recall and AP_{R40} values at 78.8% and 46.8%, respectively. It predicted an unrealistically high number of objects on average (6.421), which reflected a tendency of generating excessive false positives. Nevertheless, it demonstrated the fastest training (6 h and 28 min) and inference (0.021 s per frame).

In addition, the results of five standard tests are presented in Table 4. First of all, when the training data was reduced to 7 frames, the performance of PointR.-P1-7 declined significantly, with 22.7% recall and 2.3% AP_{R40} (IoU=0.7). The outcome suggests that effective module detection is not achievable with such limited training data without additional measures. In contrast, data augmentation applied to the 7-frame training data (PointR.-P1-7-Aug) significantly improved performance, achieving 70.4% recall and 53.6% AP_{R40} (IoU=0.7), confirming that augmentation can partially compensate for data scarcity. In addition, the pre-trained without any training data (PointR.-P0-Pre) failed to detect objects (0% recall and AP),

highlighting the necessity of fine-tuning with domain-specific data. In the PointR.-P1-7-Pre case with only 7-frame data for fine-tuning a pre-trained model, high performance of 90.5% recall and 75.6% AP_{R40} (IoU = 0.7) was achieved, requiring only 12 min of training time. The result demonstrates that introducing TL can significantly improve detection performance while drastically reducing training time (over 94% reduction compared to the reference case). Finally, the case combining pre-training and data augmentation (PointR.-P1-7-Aug-Pre) using 7-frame data achieved near-reference performance, with 94.6% recall and 88.2% AP_{R40} (IoU = 0.7), requiring only 1 h and 15 min of training time. The prediction time consistently remained below 0.22 s per frame, demonstrating that efficiency improvements were achieved without sacrificing prediction speed.

These results confirm that the combination of TL and data augmentation offers an effective strategy for substantially reducing the need for labeled data while maintaining high detection performance in point cloud-based MiC module detection. Specifically, utilizing only seven training frames yielded near-reference performance levels comparable to training with 820 frames.

Table 4 Performance comparison of eight test cases

Type	Code	Recall	AP_{R40}@0.7	Average predicted number of objects	Total training time (hh:mm)	Inference time per frame(s)
Baseline test	PointR.-P1-Full(Ref.)	98.1%	94.9%	0.960	21:28	0.219
	PartA2.-P1-Full	74.5%	74.5%	0.933	15:14	0.082
	Second.-P1-Full	78.8%	46.8%	6.421	6:28	0.021
Standard test	PointR.-P1-7	22.7%	2.3%	1.688	00:11	0.223
	PointR.-P1-7-Aug	70.4%	53.6%	0.462	01:15	0.218
	PointR.-P0-Pre	0.0%	0.0%	0.000	/	0.226
	PointR.-P1-7-Pre	90.5%	75.6%	1.073	00:12	0.222
	PointR.-P1-7-Aug-Pre	94.6%	88.2%	1.023	01:15	0.222

Fig. 14 illustrates the detection performance trends for various model configurations at different IoU thresholds. For the baseline tests, the AP_{R40} of the PointR.-P1-Full reached a stable performance plateau up to an IoU of 0.8 until a sharp decline. For the PartA2.-P1-Full case in Fig. 14(b), the curve shows a relatively smooth decline before dropping sharply after the IoU reached 0.7 (AP_{R40} of 74.5%). In contrast, the Second.-P1-Full case exhibited the earliest performance deterioration among the three baselines, with a noticeable downward trend starting from lower IoU thresholds; this case only achieved 46.8% AP_{R40} value when IoU was 0.7. For the other five standard cases, the AP_{R40} of PointR.-P1-7 (Fig. 14(d)) remains consistently below 15% across all IoU levels, eventually dropping to 2.3% at an IoU=0.7. The AP_{R40} of both PointR.-P1-7-Aug (Fig. 14(e)) and PointR.-P1-7-Pre (Fig. 14(f)) show significant

improvements compared to that of PointR.-P1-7 but still fall short of the performance of the reference case. Meanwhile, the AP_{R40} of PointR.-P1-7-Pre maintains a higher plateau level than PointR.-P1-7-Aug (approximately 80% versus 60%), indicating that the pre-training strategy is more effective than data augmentation alone. PointR.-P1-7-Aug-Pre (Fig. 14(g)) sustains high performance with an AP_{R40} (above 88.2%) up to IoU = 0.7, which is comparable to the reference case. This trend suggests that the combination of TL and data augmentation can significantly enhance performance, effectively mirroring the features of the reference case across various IoU thresholds.

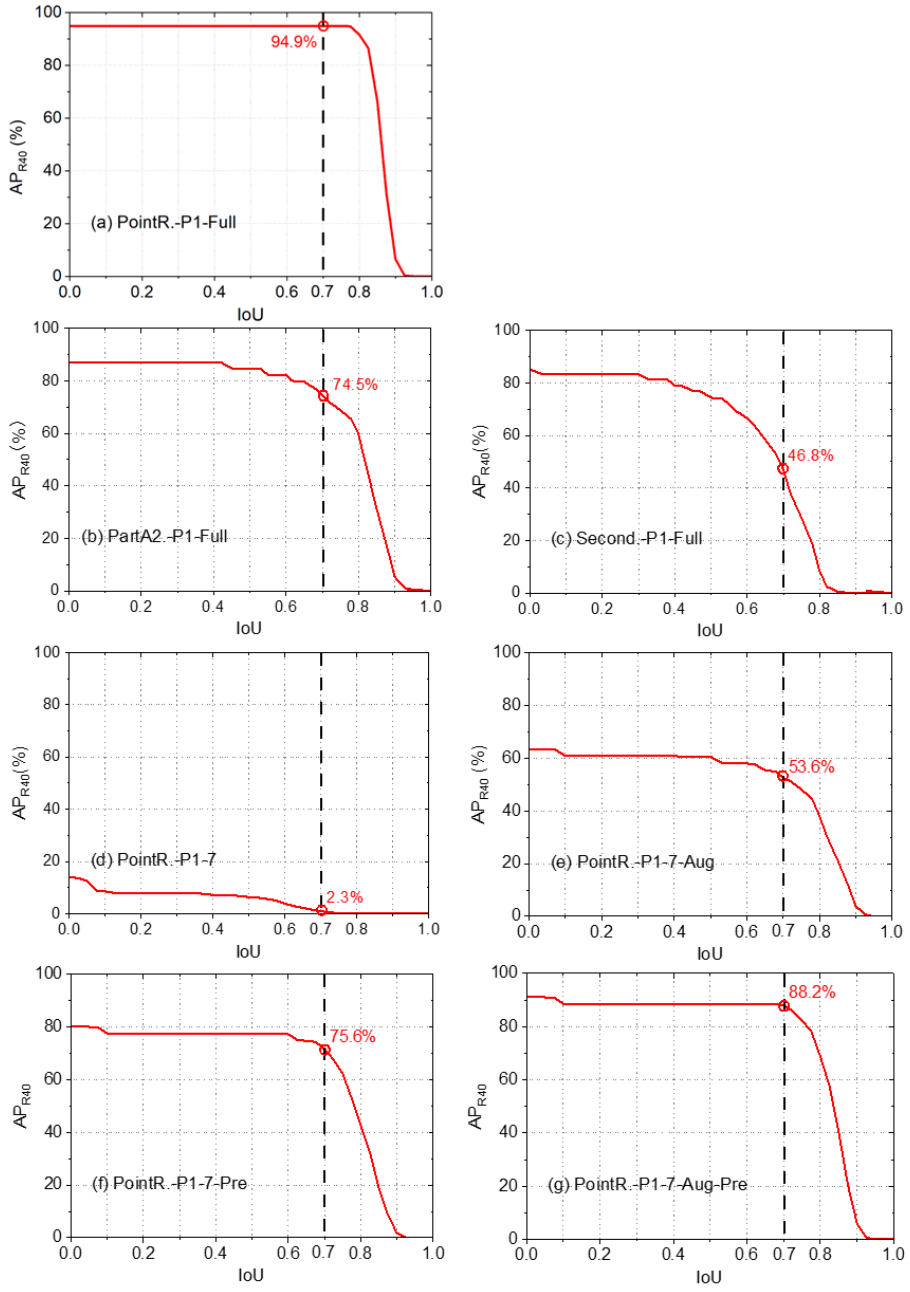


Fig. 14 Detection metrics, AP_{R40} versus IoU curve

Fig. 15 shows the training loss curves for all test cases, excluding PointR.-P0-Pre. All three baseline tests exhibit a smooth decrease in the training loss curve, which typically indicates good convergence behavior and training stability. Among the five standard test cases, PointR.-P1-Aug-Pre shows the more stable loss decline, whereas the PointR.-P1-7 case is the only one that lacks a discernible downward trend. This observation aligns with the earlier results regarding detection accuracy. Specifically, the training loss for the PointR.-P1-Full case quickly stabilizes after approximately 50 iterations, ultimately converging to a minimal loss value near 0.5. In contrast, both PartA2.-P1-Full and Second.-P1-full cases converged before 25 iterations and finally stayed at a level of about 0.6. In addition, the loss curve for the PointR.-P1-7 case displays significant fluctuations throughout all 300 iterations, failing to decrease below 2.0. The training losses of the other three cases, i.e., PointR.-P1-7-Aug, PointR.-P1-7-Pre, and PointR.-P1-7-Aug-Pre, demonstrate a slow decline, stabilizing after about 200 iterations. Notably, the PointR.-P1-7-Aug-Pre case is the only one that eventually stabilizes below a loss of 1, resembling the behavior of the reference case, PointR.-P1-Full.

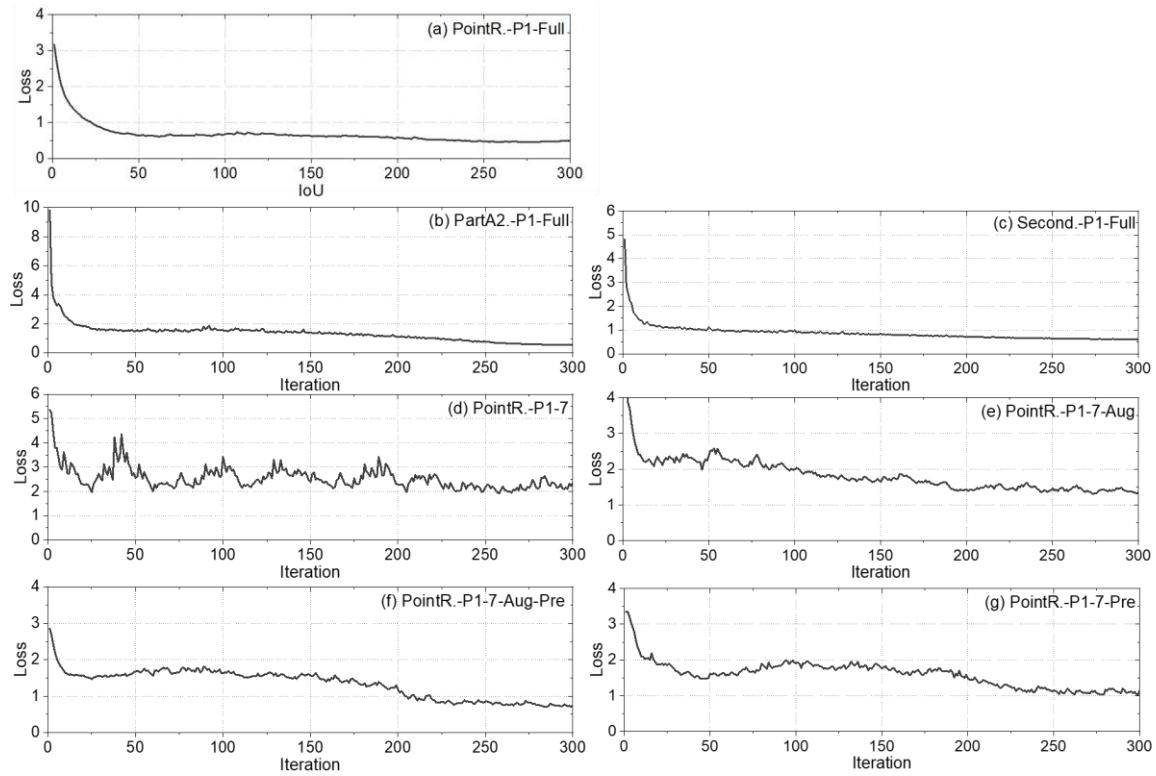


Fig. 15 Training loss curve

Fig. 16 displays typical prediction results for the PointR.-P1-7-Aug-Pre case. Figs. 16(a)–16(c) depict TP results from the initial hoisting phase to the final placing, where the model

successfully predicted 3D bounding boxes that accurately aligned with the point cloud data of the MiC module. In contrast, Figs. 16(d)–16(f) illustrate a false positive (FP) result, where the predicted 3D bounding box significantly deviates from the actual point clouds of the MiC modules.

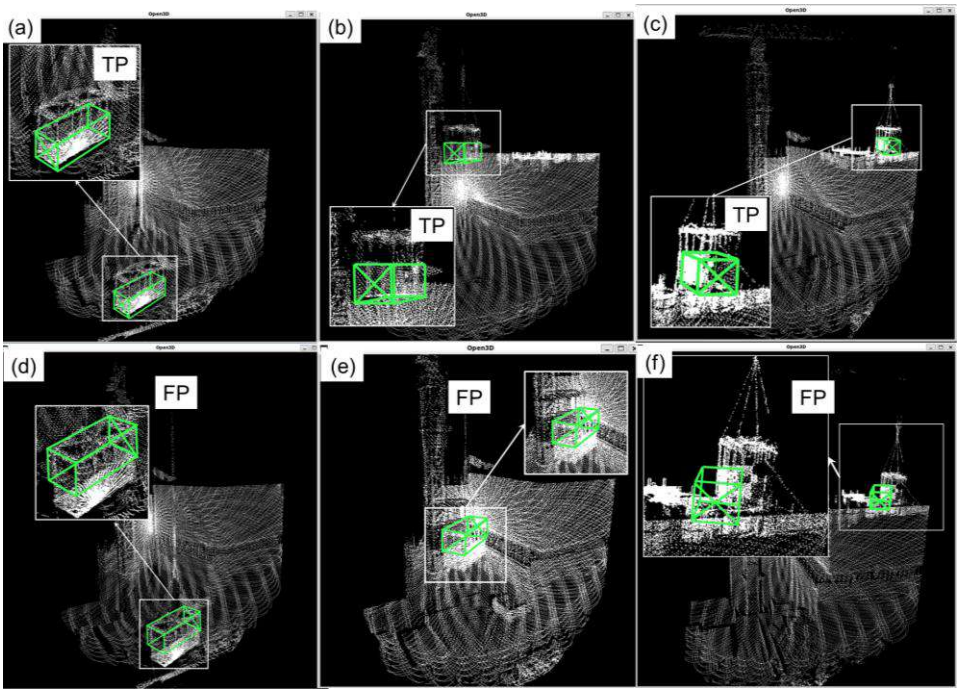


Fig. 16 Example prediction result of developed TL model

4.3 Sensitivity analysis

The performance results of the four-group sensitivity tests are listed in Table 5. Notably, the cases within the group of 3D detection algorithms exhibit significant performance variations. When compared to the reference case (PointR.-P1-7-Pre), the PartA2 algorithm (PartA-P1-7-Pre) demonstrated moderate performance, achieving 58.1% recall and 57.9% AP_{R40} (IoU = 0.7), while also reducing the training time to 10 min. The Second-P1-7-Pre achieved the lowest performance (44.2%) in AP_{R40} (IoU = 0.7), with a recall of 74.7%. However, it exhibited the fastest prediction speed (0.026 s per frame) and substantial efficiency advantages, requiring only 8 min of training time. Notably, compared to both PointRCNN and PartA, the Second algorithm detected more objects per frame (7.592) but with low detection accuracy, suggesting potential over-detection issues that compromise overall accuracy. These results underscore the trade-off between accuracy and efficiency that must be considered when selecting an appropriate detection algorithm.

For the training data group, the results demonstrate a progressive improvement in PointRCNN's performance as the training data increases. Specifically, performance improved from 78.6% recall and 64.5% AP_{R40} with four frames to 90.5% recall and 75.6% AP_{R40} with seven frames and further to 94.5% recall and 81.2% AP_{R40} with 13 frames. The PointR.-P1-4-Pre case shows that pre-trained models can still achieve moderate results, when fine-tuned with four frames

For the dataset selection group, models trained on the P1 and P3 datasets outperformed those trained on the P2 and P4 datasets. In particular, the PointR.-P3-7-Pre case achieved 87.4% recall and 75.1% AP_{R40}, closely approaching the performance of the reference case. On the other hand, the model trained on the P4 dataset exhibited the lowest performance (75.0% recall, 62.0% AP_{R40}). These results suggest that datasets such as P1 and P3, which offer more diverse and representative data, contribute to better performance.

Finally, data augmentation strategies notably enhanced the performance of the model. Among the individual augmentation techniques, the case with rotation augmentation achieved the highest performance among those with other individual augmentation techniques (54.9% recall and 51.3% AP_{R40}). The case employing a combination of augmentation techniques (PointR.-P1-7-Aug) demonstrated the most significant improvement, achieving a recall of 70.4% and an AP_{R40} (IoU = 0.7) of 53.6%.

Table 5 Result of sensitivity analysis

Category	Code	Recall	AP_{R40}(IoU=0.7)	Average predicted number of objects	Total training time (hh:mm)	Inference time per frame(s)
3D object detection algorithm	PointR.-P1-7-Pre(Ref.1)	**90.5%**	**75.6%**	1.073	00:12	0.222
	PartA-P1-7-Pre	58.1%	57.9%	0.901	00:10	0.069
	Second-P1-7-Pre	74.7%	44.2%	7.592	**00:08**	**0.026**
Training data number	PointR.-P1-4-Pre	78.6%	64.5%	0.853	00:08	0.229
	PointR.-P1-13-Pre	**94.5%**	**81.2%**	1.035	00:19	0.226
Dataset selection	PointR.-P2-7-Pre	78.1%	68.1%	1.011	00:13	0.224
	PointR.-P3-7-Pre	**87.4%**	**75.1%**	1.029	00:13	0.222
	PointR.-P4-7-Pre	77.0%	67.0%	0.919	00:12	0.224
Data augmentation	PointR.-P1-7(Ref.2)	22.7%	2.3%	1.688	00:11	0.223
	PointR.-P1-7-Crop	51.2%	45.2%	0.491	01:23	0.218
	PointR.-P1-7-Rot	54.9%	51.3%	0.705	01:30	0.224
	PointR.-P1-7-Samp	52.2%	44.8%	0.491	01:19	0.218
	PointR.-P1-7-Aug	**70.4%**	**53.6%**	0.462	01:15	0.218

Fig. 17 further compares AP_{R40} and IoU metrics across different groups in the sensitivity analysis. For the 3D object detection algorithm group, both the PointR.-P1-7-Pre and Second-P1-7-pre cases exhibit a similar trend in AP_{R40}, i.e., stabilizing until an IoU threshold of 0.7,

after which they decline sharply to zero at IoU=0.9. Although the AP_{R40} of the PartA-P1-7-Pre case remains consistent with that of the PointR.-P1-7-Pre case within the IoU=0.4, it steadily decreases and shows a more pronounced downward slope as the IoU value increases. For the training data group, the case with a larger amount of training data (PointR.-P1-13-Pre) shows a higher initial value compared to the case with less training data (PointR.-P1-4-Pre), although both cases follow the same decreasing trend as the IoU threshold increases. In the dataset selection group, different cases show a similar trend in $AP_{R40,}$ with slight variations in the initial values. For the data augmentation group, the AP_{R40} of cases incorporating various data augmentation techniques consistently starts at approximately 60%. Specifically, the PointR.-P1-7-Crop and PointR.-P1-7-Samp cases exhibit a more pronounced downward trend, resulting in lower AP_{R40} for PointR.-P1-7-Crop and PointR.-P1-7-Samp, which are smaller than those for PointR.-P1-7-Rot and PointR.-P1-7-Aug at IOU=0.7.

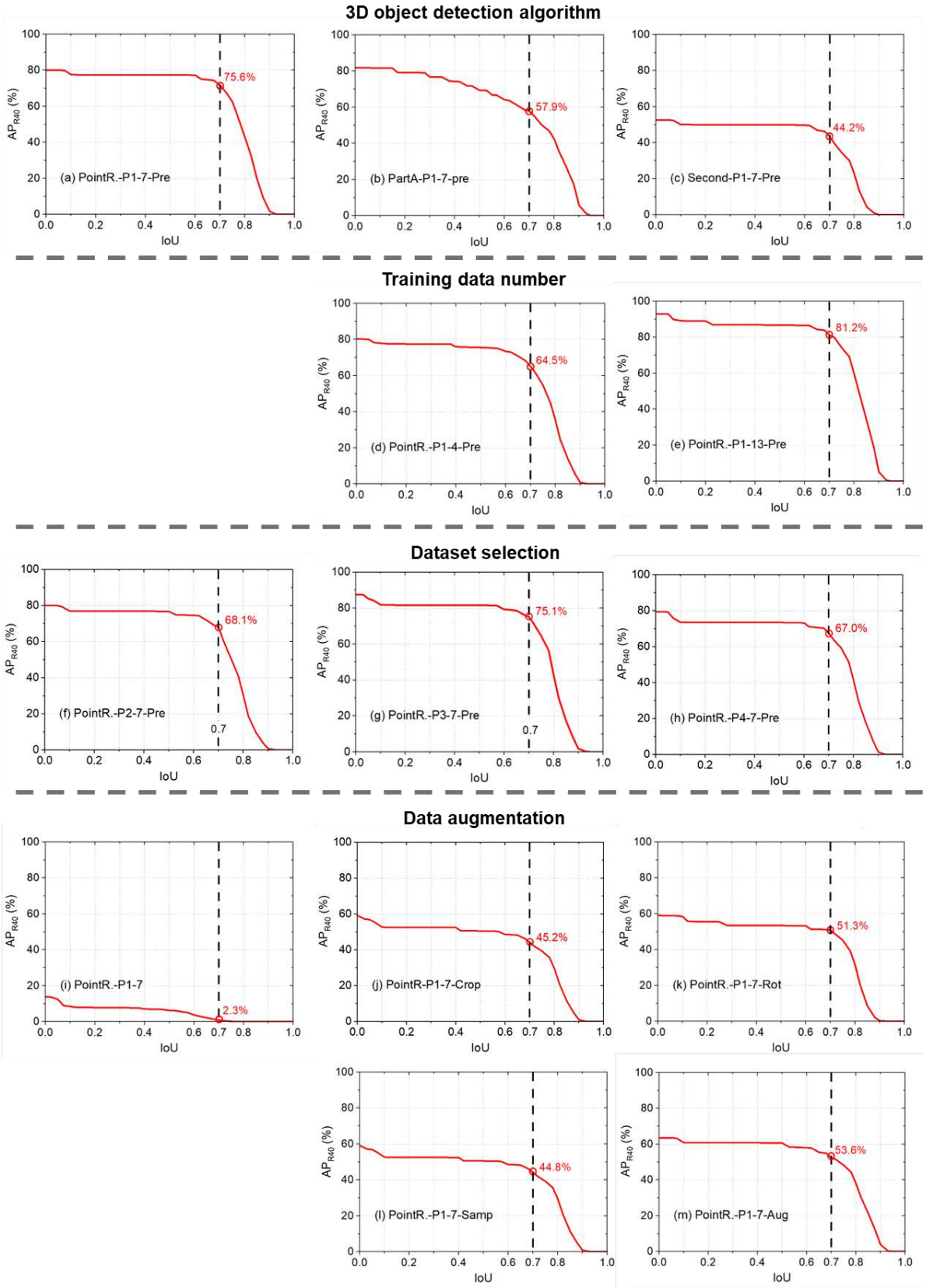


Fig. 17 Detection metrics, AP_{R40} versus IoU curve, for sensitivity analysis

In summary, the selection of a 3D object detection algorithm has a substantial impact on both accuracy and efficiency. The PointRCNN algorithm demonstrated the highest performance in terms of detection accuracy, while the Second algorithm offered faster processing times, albeit with lower accuracy. Additionally, the amount of training data directly influences the performance of the pre-trained model, with even limited fine-tuning using only four frames yielding moderate results. Dataset selection also plays a crucial role in detection accuracy, with more diverse and representative data contributing to improved model performance. Lastly, data augmentation techniques, particularly rotation augmentation, significantly enhance the model effectiveness, with the combined augmentation strategy yielding the best overall performance.

5 Discussion

5.1 Potential applications

The results from the MiC module detection can be extended to multiple potential applications on construction sites:

- **Risks alert system**: The 3D object detection results can be leveraged to identify and analyze potential risks on the construction site. For example, the system can notify on-site workers when they enter a hazardous zone beneath a hoisted module. In addition, crane operators can be alerted if the distance between the hoisted module and a worker or between the module and nearby obstacles falls below a predefined safety threshold, thus enhancing site safety and preventing accidents.
- **Productivity and progress monitoring**: A system for productivity and progress monitoring can be developed by integrating the detection results with BIM models. This system can analyze the progress of the construction process through changes in the position of the modules. Time and location data associated with module hoisting will be recorded and integrated into the project management system. By combining the project management system with AI tools, the system can facilitate automated and intelligent productivity evaluations and provide real-time progress updates (Chen et al. 2025).
- **Quality control system**: A quality control system for the installation of MiC modules can be established by comparing the detection results with the as-designed BIM. This system can accurately identify deviations in the module position and orientation, verifying whether they fall within acceptable tolerance ranges. Based on the deviation

analysis provided by the system, crane operators can take corrective actions to ensure precise module placement.

5.2 Applicability beyond MiC module hoisting

The "BIM-to-Reality" 3D detection paradigm can be extended beyond MiC module hoisting scenarios. The source domain of BIM technology is the most important digital project data source and largely mandated in the modern projects, and the network architecture effectively learns universal spatial features, not limited to hoisting. For example, the proposed paradigm can first be used in detecting other lifting objects, such as precast concrete beams, exterior wall panels, large equipment components, or bridge segments. Furthermore, the same workflow can be applied to construction ground task detection, alerts for workers near a 3D dangerous zone, human-robot collaboration, task-aware point cloud compression, and so on, using the proposed synthetic data generation and model pre-training techniques. All these applications should inherit high-level accuracy and low-level dependency on real-world data in this paper. The proposed approach can be extended to other industrial scenarios, such as infrastructure construction, industrial plant installation, and marine engineering, in which lifting operations are crucial for safety and productivity. Although the same "BIM-to-Reality" workflow can be followed, it should be noted that there are different characteristics for specific tasks and objects in the extension scenarios. For instance, on-site workers have small, distant, and fast-moving characteristics, which are different from the characteristics of usual lifting objects. These characteristics may lead to implementation challenges, which need further exploration in the future.

5.3 Strengths and Limitations

The proposed BIM-based TL paradigm for 3D module detection in point clouds of MiC hoisting shows strength in the following two aspects compared to the conventional image-based detection method:

- **Accurate three-dimensional sensing.** The proposed method inherently retains comprehensive three-dimensional geometric information in the hoisting operation scene. The 3D detection result supports localization at centimeter-level precision and allows accurate calculation of the distances between, the module, crane hook, and adjacent structures. However, conventional image-based monitoring methods can only

recognize module presence in installation stage but lack direct depth information, which limits reliable three-dimensional positioning and collision risk assessment.

- **Robust against external lighting variation.** The proposed 3D detection method in point clouds is less sensitive to lighting variations, shadows, or occlusions that commonly degrade image-based detection in outdoor hoisting environments.

Despite the strength and promising results, the proposed TL paradigm for MiC module detection still has several limitations that warrant further exploration in future research.

- **Difference between synthetic and real-world data**: The synthetic data is generated from depth maps in a virtual environment, whereas the real-world data is collected using LiDAR sensors in this study. The synthetic data generation method based on the depth map is the most widely adopted in the related studies (Xu & Pan 2023). Part of the reason is that the point cloud acquisition principle of LiDAR also varies in different products. It is difficult to create a general method to create synthetic data. However, it can not be ignored that there are differences in sensing mechanisms between these two methods of point cloud generation. Therefore, the influence of the difference on transfer learning needs to be explored in the future.
- **Reliability in complex environments**: The reliability of the proposed method needs to be tested in more complex environments, particularly when dealing with heavily occluded modules. In urban construction sites, occlusions caused by temporary buildings, construction equipment, or other modules may hinder LiDAR sensors from collecting sufficient point cloud data, which could impact the model's accuracy. In addition, environmental conditions such as weather, fog, and strong lighting can significantly impact the quality of point clouds at outdoor construction sites. Current research has not conducted extensive testing to assess system performance under these adverse conditions.
- **LiDAR sensor deployment**: The current approach requires strategic placement of multiple LiDAR sensors to ensure comprehensive point-cloud coverage across the construction site. Replicating this configuration across various projects may pose challenges due to site-specific constraints and the high cost of deploying multiple high-quality sensors. Future research should explore more flexible sensor deployment techniques, such as simulation-based optimization of sensor deployment (Zhou & Xue 2025; Sun et al. 2026).

- **Noise concern in synthetic data generation**: The synthetic point clouds generated from BIM are inherently noise-free, whereas real-world LiDAR scans inevitably contain random measurement variance. To reduce this gap, random downsampling was applied to the dense depth-map-derived point clouds, which both balanced point density and introduced irregularities comparable to LiDAR noise. In addition, the BIM environment incorporated cluttered site elements such as stacked materials, workers, and exposed reinforcement cages, increasing occlusion and background complexity that resemble noise-like effects in practice. Furthermore, compared with the module dimensions (9.2 m × 3.7 m × 3.4 m), the potential noise can be negligible since the LiDAR sensor keeps 2 cm precision. Nevertheless, future research should explore explicit noise modeling in synthetic point cloud generation to improve domain consistency, although the experimental result showed that the noise level in real LiDAR scans did not substantially degrade detection performance.
- **IoU threshold for precision**: The IoU threshold for positive samples was set at 0.7, based on standards used for large objects in autonomous driving. However, this threshold may not be sufficient for the precise positioning of MiC modules. Our experiments indicated that varying the threshold did not yield optimal detection results, suggesting that further refinements are required to improve spatial matching precision.

6 Conclusion

This paper introduces a TL paradigm for module detection in real-world MiC hoisting processes, utilizing BIM-based synthetic data. The proposed paradigm effectively addresses the challenge of limited labeled datasets commonly encountered in construction applications by integrating minimal real-world data with synthetic data generated from a virtual environment. Experimental results indicate that combining TL with data augmentation and using only seven training frames resulted in a recall of 94.6% and an AP_{R40} (IoU = 0.7) score of 88.2%. This performance is comparable to a model trained on 820 frames. In the sensitivity analysis, the PointRCNN algorithm demonstrated superior accuracy compared to the other two competing approaches (e.g., PartA2 and Second) despite its higher computational cost.

Furthermore, pre-trained models demonstrated the ability to achieve satisfactory results with limited fine-tuning data. Notably, increasing the number of training data from 4 to 13 resulted in a significant improvement in detection accuracy. Dataset selection was also found to be a

critical factor influencing model performance. Training with diverse and representative datasets (from P1) consistently produced superior results. In addition, data augmentation methods significantly enhanced model performance, with integrated augmentation strategies outperforming individual augmentation methods by a considerable margin.

The proposed TL paradigm effectively reduces the need for large amounts of labeled point cloud data for 3D object detection in disordered construction scenarios. However, further research is needed to address current limitations, including the synthetic data creation mechanism, performance under complex environments, sensor deployment constraints, and higher IoU thresholds. Moreover, future studies could also explore the integration of multi-modal data and advanced sensor technologies to improve detection accuracy and robustness in real-world construction environments.

Acknowledgment

The work in this paper was supported in part by the Innovation and Technology Commission (ITC) (No. ITP/004/23LP) and The University of Hong Kong (A/C No. 109001305). We appreciate the China State Construction Hailong Technology Company Limited. providing case project access. The training of the TL model was performed using high performance computing (HPC) facilities offered by Information Technology Services, The University of Hong Kong. The authors would also like to thank Xi Wei for assisting with the data curation.

CRediT Author Statement

Dong Liang: Writing – original draft, Formal analysis. **Longyong Wu**: Methodology. **Meng Sun**: Data curation. **Ruibo Hu**: Data curation. **Lingming Kong**: Data curation. **Yipeng Pan**: Software. **Fan Xue**: Writing - Review & Editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Data availability

The authors have shared the GitHub link for the Unity project, code, and synthetic dataset (https://github.com/leodongdong/Unity_to_pointcloud). In addition, real-world project data is available upon request from the authors.

References

Arshad, H. & Zayed, T. (2024). A multi-sensing IoT system for MiC module monitoring during logistics and operation phases. *Sensors, 24*(15), 4900. doi:10.3390/s24154900

Chen, J., Fang, Y. & Cho, Y. K. (2017). Real-Time 3D crane workspace update using a hybrid visualization approach. *Journal of Computing in Civil Engineering, 31*(5), 04017049. doi:10.1061/(ASCE)CP.1943-5487.0000698

Chen, J., Fang, Y. & Cho, Y. K. (2018). Performance evaluation of 3D descriptors for object recognition in construction applications. *Automation in Construction, 86*, 44-52. doi:10.1016/j.autcon.2017.10.033

Chen, L., Wang, Y. & Siu, M.-F. F. (2020). Detecting semantic regions of construction site images by transfer learning and saliency computation. *Automation in Construction, 114*, 103185. doi:10.1016/j.autcon.2020.103185

Chen, Y., Zhang, H., Li, C., Chi, B., Chen, X. & Wu, J. (2025). Large language model empowered smart city mobility. *Frontiers of Engineering Management, 12*, 201-207. doi:10.1007/s42524-025-4213-0

Chua, W. P. & Cheah, C. C. (2024). Deep-learning-based automated building construction progress monitoring for prefabricated prefinished volumetric construction. *Sensors, 24*(21), 7074. doi:10.3390/s24217074

Czerniawski, T. & Leite, F. (2019). Semantic segmentation of building point clouds using deep learning: A method for creating training data using BIM to point cloud label transfer. *ASCE International Conference on Computing in Civil Engineering 2019* (pp. 410-416). American Society of Civil Engineers. doi:10.1061/9780784482421.052

Ding, Y. & Luo, X. (2024). A virtual construction vehicles and workers dataset with three-dimensional annotations. *Engineering Applications of Artificial Intelligence, 133*, 107964. doi:10.1016/j.engappai.2024.107964

Dong, Z., Lu, W. & Chen, J. (2024). Neural rendering-based semantic point cloud retrieval for indoor construction progress monitoring. *Automation in Construction, 164*, 105448. doi:10.1016/j.autcon.2024.105448

Fang, W., Love, P. E., Luo, H. & Ding, L. (2020). Computer vision for behaviour-based safety in construction: A review and future directions. *Advanced Engineering Informatics, 43*, 100980. doi:10.1016/i.aei.2019.100980

Frías, E., Pinto, J., Sousa, R., Lorenzo, H. & Díaz-Vilariño, L. (2022). Exploiting BIM objects for synthetic data generation toward indoor point cloud classification using deep learning. *Journal of Computing in Civil Engineering, 36*(6), 04022032. doi:10.1061/(ASCE)CP.1943-5487.0001039

Fu, Y., Lu, W., Dong, Z. & Fang, Y. (2025). A synthetic data-enhanced method for automated 3D pose recognition of construction workers. *Expert Systems With Applications, 294*, 128768. doi:10.1016/j.eswa.2025.128768

Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research, 32*(11), 1231-1237. doi:10.1177/0278364913491297

Gong, Y., Shao, H., Luo, J. & Li, Z. (2020). A deep transfer learning model for inclusion defect detection of aeronautics composite materials. *Composite Structures, 252*, 112681. doi:10.1016/j.compstruct.2020.112681

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L. & Bennamoun, M. (2021). Deep learning for 3D point clouds: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(12), 4338-4364. doi:10.1109/TPAMI.2020.3005434

Hartley, R. & Zisserman, A. (2003). *Multiple view geometry in computer vision.* Cambridge university press. doi:10.1017/CBO9780511811685

Hong, Y., Park, S. & Kim, H. (2020). Synthetic data generation for indoor scene understanding using BIM. *Proceedings of the International Symposium on Automation and Robotics in Construction. 37*, pp. 334-338. IAARC Publications.

Jaquier, N., Welle, M. C., Gams, A., Yao, K., Fichera, B., Billard, A., Ude, A., Asfour, T. & Kragic, D. (2023). Transfer learning in robotics: An upcoming breakthrough? A review of promises and challenges. *The International Journal of Robotics Research, 44*(3), 465-485. doi:10.1177/02783649241273565

Jiang, W. & Ding, L. (2024). Unsafe hoisting behavior recognition for tower crane based on transfer learning. *Automation in Construction, 160*, 105299. doi:10.1016/j.autcon.2024.105299

Jiang, Y., Li, M., Guo, D., Wua, W., Zhonga, R. Y. & Huang, G. Q. (2022). Digital twin-enabled smart modular integrated construction system for on-site assembly. *Computers in Industry, 136*, 103594. doi:10.1016/j.compind.2021.103594

Kolar, Z., Chen, H. & Luo, X. (2018). Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images. *Automation in Construction, 89*, 58-70. doi:10.1016/j.autcon.2018.01.003

Kong, L., Zhao, R., Anumba, C. J., Lu, W. & Xue, F. (2025). Open BIM exchange on Blockchain 3.0 virtual disk: A traceable semantic differential transaction approach. *Frontiers of Engineering Management, in press*. doi:10.1007/s42524-024-4006-x

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J. & Beijbom, O. (2019). PointPillars: fast encoders for object detection from point clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 12697-12705). doi:10.1109/CVPR.2019.01298

Li, E., Wang, S., Li, C., Li, D., Wu, X. & Hao, Q. (2020). SUSTech POINTS: A portable 3D point cloud interactive annotation platform system. *IEEE Intelligent Vehicles Symposium (IV)* (pp. 1108-1115). IEEE. doi:10.1109/IV47402.2020.9304562

Li, J., Yang, L., Shi, Z., Chen, Y., Jin, Y., Akiyama, K. & Xu, A. (2024). SparseDet: Towards efficient multi-view 3D object detection via sparse scene representation. *Advanced Engineering Informatics, 62*, 102955. doi:10.1016/j.aei.2024.102955

Li, J., Yuan, C., Wang, X., Chen, G. & Ma, G. (2025a). Semi-supervised crack detection using segment anything model and deep transfer learning. *Automation in Construction, 170*, 105899. doi:10.1016/j.autcon.2024.105899

Li, M., Xue, F. & Yeh, A. G. (2025). Efficient and accurate assessment of window view distance using City Information Models and 3D Computer Vision. *Landscape and Urban Planning, 260*, 105389. doi:10.1016/j.landurbplan.2025.105389

Li, M., Xue, F., Wu, Y. & Yeh, A. G. (2022). A room with a view: Automatic assessment of window views for high-rise high-density areas using City Information Models and deep transfer learning. *Landscape and Urban Planning, 226*, 104505. doi:10.1016/j.landurbplan.2022.104505

Li, Z., Yu, Y., Tian, F., Chen, X., Xiahou, X. & Li, Q. (2025b). Vigilance recognition for construction workers using EEG and transfer learning. *Advanced Engineering Informatics, 64*, 103052. doi:10.1016/j.aei.2024.103052

Liang, D., Chen, S.-H., Chen, Z., Wu, Y., Chu, L. Y. & Xue, F. (2024). 4D point cloud-based spatial-temporal semantic registration for monitoring mobile crane construction activities. *Automation in Construction, 165*, 105576. doi:10.1016/j.autcon.2024.105576

Liang, D., Chen, S.-H., Wu, Y., Chen, Z. & Xue, F. (2025). Digital twinning hoisting process of high-rise MiC components based on 4D point cloud and BIM models. *Proceedings of the 29th International Symposium on Advancement of Construction Management and Real Estate (CRIOCM2024).* Springer (in press).

Liang, M., Yang, B., Chen, Y., Hu, R. & Urtasun, R. (2019). Multi-task multi-sensor fusion for 3D object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 7337-7346). doi:10.1109/CVPR.2019.00752

Liu, Z., Meng, X., Xing, Z. & Jiang, A. (2021). Digital twin-based safety risk coupling of prefabricated building hoisting. *Sensors, 21*, 3583. doi:10.3390/s21113583

Ma, J. W., Czerniawski, T. & Leite, F. (2020). Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds. *Automation in Construction, 113*, 103144. doi:10.1016/j.autcon.2020.103144

Meng, S., Su, X., Sun, G., Li, M. & Xue, F. (2025). From 3D pedestrian networks to wheelable networks: An automatic wheelability assessment method for high-density urban areas using contrastive deep learning of smartphone point clouds. *Computers, Environment and Urban Systems, 117*, 102255. doi:10.1016/j.compenvurbsys.2025.102255

Oh, J.-H., Kim, H.-G. & Lee, K. M. (2023). Developing and evaluating deep learning algorithms for object detection: key points for achieving superior model performance. *Korean Journal of Radiology, 24*(7), 698. doi:10.3348/kjr.2022.0765

OpenPCDet Development Team. (2020). *OpenPCDet: An open-source toolbox for 3D object detection from point clouds*. Retrieved from https://github.com/open-mmlab/OpenPCDet

Pan, S. J. & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345-1359. doi:10.1109/TKDE.2009.191

Pan, W., Ng, T., Huang, G., Chan, S., Au, F., Tam, K., Chu, L., Yang, Y., Zheng, Z. & Pan, M. (2021). *Modular integrated construction for high-rise buildings in Hong Kong: supply chain identification, analyses and establishment.* Construction Industry Council. Retrieved from https://www.cic.hk/files/page/10389/CIC_Reference%20Material_MiC%20Market%20Analysis%20Report.pdf

Panahi, R., Louis, J., Podder, A., Swanson, C. & Pless, S. (2023). Automated assembly progress monitoring in modular construction factories using computer vision-based

instance segmentation. *Computing in Civil Engineering 2023*, (pp. 290-297). doi:10.1061/9780784485224.036

Pham, H. T. & Han, S. (2024). Generating realistic training images from synthetic data for excavator pose estimation. *Automation in Construction, 167*, 105718. doi:10.1016/j.autcon.2024.105718

Qi, C. R., Su, H., Mo, K. & Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR)*, (pp. 652-660). doi:10.1109/CVPR.2017.16

Qi, C. R., Yi, L., Su, H. & Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, (p. 30). doi:10.5555/3295222.3295263

Qian, R., Lai, X. & Li, X. (2022). 3D Object detection for autonomous driving: a survey. *Pattern Recognition, 130*, 108796. doi:10.1016/j.patcog.2022.108796

Schreier, T., Renz, K., Geiger, A. & Chitta, K. (2023). On offline evaluation of 3d object detection for autonomous driving. *IEEE/CVF International Conference on Computer Vision(CVPR)*, (pp. 4084-4089). doi:10.1109/iccvw60793.2023.00441

Shao, Y., Sun, Z., Tan, A. & Yan, T. (2023). Efficient three-dimensional point cloud object detection based on improved Complex-YOLO. *Frontiers in Neurorobotics, 17*, 1092564. doi:10.3389/fnbot.2023.1092564

Sharif, M.-M., Nahangi, M., Haas, C. & West, J. (2017). Automated model-based finding of 3D objects in cluttered construction point cloud models. *Computer-Aided Civil and Infrastructure Engineering, 32*, 893-908. doi:10.1111/mice.12306

Shi, S., Wang, X. & Li, H. (2019). Pointrcnn: 3d object proposal generation and detection from point cloud. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition(CVPR)*, (pp. 770-779). doi:10.1109/cvpr.2019.00086

Son, H., Kim, C. & Choi, K. (2010). Rapid 3D object detection and modeling using range data from 3D range imaging camera for heavy equipment operation. *Automation in Construction, 19*, 898-906. doi:10.1016/j.autcon.2010.06.003

Sun, A., An, X., Li, P., Lv, M. & Liu, W. (2025). Near real-time 3D reconstruction of construction sites based on surveillance cameras. *Buildings, 15*(4), 567. doi:10.3390/buildings15040567

Sun, M., Liang, D., Wang, J., Bate, B. & Xue, F. (2026). Clarity in DEM cementation predictions: Integrating automated machine learning and interpretability analysis of

shear wave velocity. *Advanced Engineering Informatics, 69*(Part D), 104060. doi:10.1016/j.aei.2025.104060

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M. & Gao, A. (2020). Scalability in perception for autonomous driving: Waymo open dataset. *IEEE/CVF conference on computer vision and pattern recognition(CVPR)*, (pp. 2446-2454). doi:10.1109/cvpr42600.2020.00252

Teizer, J. (2008). 3D range imaging camera sensing for active safety in construction. *Journal of Information Technology in Construction (ITcon), 13*(8), 103-117. Retrieved from https://www.itcon.org/2008/8

Teizer, J., Bosche, F., Caldas, C. H., Haas, C. T. & Liapi, K. A. (2005). Real-Time, three-dimensional object detection and modeling in construction . *22nd International Symposium on Automation and Robotics in Construction (ISARC)*, (pp. 1-5). doi:10.22260/isarc2005/0026

Wang, T. & Gan, V. J. (2023). Automated joint 3D reconstruction and visual inspection for buildings using computer vision and transfer learning. *Automation in Construction, 149*, 104810. doi:10.1016/j.autcon.2023.104810

Wang, Y., Zhao, N. & Lee, G. H. (2024). Syn-to-real unsupervised domain adaptation for indoor 3d object detection. *arXiv preprint arXiv:2406.11311*.

Weiss, K., Khoshgoftaar, T. M. & Wang, D. (2016). A survey of transfer learning. *Journal of Big data, 3*, 1-40. doi:10.1186/s40537-016-0043-6

Wu, L., Zhang, Q., Hou, J. & Xu, Y. (2023). Leveraging single-view Images for unsupervised 3D point cloud completion. *IEEE Transactions on Multimedia, 27*, 940-953. doi:10.1109/TMM.2023.3340892

Wu, Y., Xue, F., Li, M. & Chen, S. (2024). A novel Building Section Skeleton for compact 3D reconstruction from point clouds: A study of high-density urban scenes. *ISPRS Journal of Photogrammetry and Remote Sensing, 209*, 85-100. doi:10.1016/j.isprsjprs.2024.01.020

Wuni, I. Y., Shen, G. Q. & Hwang, B.-G. (2020). Risks of modular integrated construction: A review and future research directions. *Frontiers of Engineering Management, 7*(1), 63-80. doi:10.1007/s42524-019-0059-7

Xiao, K., Li, T., Li, J., Huang, D. & Peng, Y. (2024). Equal emphasis on data and network: a two-stage 3D point cloud object detection algorithm with feature alignment. *Remote sensing, 16*(2), 249. doi:10.3390/rs16020249

Xu, J. & Pan, W. (2023). Virtual prototyping-enabled pseudo-LiDAR point cloud dataset for 3D module detection in modular integrated construction. *The 30th EG-ICE: International Conference on Intelligent Computing in Engineering.*

Yang, B., Luo, W. & Urtasun, R. (2018). Pixor: Real-time 3d object detection from point clouds. *IEEE conference on Computer Vision and Pattern Recognition(CVPR)*, (pp. 7652-7660). doi:10.1109/cvpr.2018.00798

Yang, L., Lin, Y.-C., Cai, H. & Habib, A. (2024). From scans to parametric BIM: An enhanced framework using synthetic data augmentation and parametric modeling for highway bridges. *Journal of Computing in Civil Engineering, 38*(3), 04024008. doi:10.1061/JCCEE5.CPENG-5640

Yin, J., Aidi, A. H., Nuzul, H. A. & Nabilah, B. A. (2024). Intelligent construction risk management through transfer learning: trends, challenges, and future strategies. *Artificial Intelligence Evolution, 6*(1), 1-16. doi:10.37256/aie.6120255255

Zhai, Y., Chen, K., Zhou, J. X., Cao, J., Lyu, Z., Jin, X., Shen, G. Q., Lu, W. & Huang, G. Q. (2019). An Internet of Things-enabled BIM platform for modular integrated construction: A case study in Hong Kong. *Advanced Engineering Informatics, 42*, 100997. doi:10.1016/j.aei.2019.100997

Zhang, Z. & Pan, W. (2021). Multi-criteria decision analysis for tower crane layout planning in high-rise modular integrated construction. *Automation in Construction, 127*, 103709. doi:10.1016/j.autcon.2021.103709

Zhang, Z., Pan, W. & Zheng, Z. (2019). Transfer learning enabled process recognition for module installation of high-rise modular buildings. *Modular and Offsite Construction (MOC) Summit Proceedings*, (pp. 268-275). doi:10.29173/mocs103

Zhao, H., Meng, M., Li, X., Xu, J., Li, L. & Galland, S. (2024b). A survey of autonomous driving frameworks and simulators. *Advanced Engineering Informatics, 62*, 102850. doi:10.1016/j.aei.2024.102850

Zhao, S., Wang, J., Shi, T. & Huang, K. (2024a). Contrastive and transfer learning-based visual small component inspection in assembly. *Advanced Engineering Informatics, 59*, 102308. doi:10.1016/j.aei.2023.102308

Zheng, Z., Zhang, Z. & Pan, W. (2020). Virtual prototyping- and transfer learning-enabled module detection for modular integrated construction. *Automation in Construction, 120*, 103387. doi:10.1016/j.autcon.2020.103387

Zhou, J. X., Shen, G. Q., Yoon, S. H. & Jin, X. (2021). Customization of on-site assembly services by integrating the internet of things and BIM technologies in modular integrated construction. *Automation in Construction, 126*, 103663. doi:10.1016/j.autcon.2021.103663

Zhou, Q. & Xue, F. (2025). Automatic Information Gain-guided Convergence for refining building design parameters: Enhancing effectiveness and interpretability in simulation-based optimization. *Building and Environment, 275*, 112788. doi:10.1016/j.buildenv.2025.112788

Zhou, Y. & Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR)*, (pp. 4490-4499). doi:10.1109/CVPR.2018.00472

Zhu, A., Zhang, Z. & Pan, W. (2022). Crane-lift path planning for high-rise modular integrated construction through metaheuristic optimization and virtual prototyping. *Automation in Construction, 141*, 104434. doi:10.1016/j.autcon.2022.104434

Zhu, A., Zhang, Z. & Pan, W. (2024). Developing a fast and accurate collision detection strategy for crane-lift path planning in high-rise modular integrated construction. *Advanced Engineering Informatics, 61*, 102509. doi:10.1016/j.aei.2024.102509

Zhu, Q., Fan, L. & Weng, N. (2024). Advancements in point cloud data augmentation for deep learning: A survey . *Pattern Recognition, 153*, 110532. doi:10.1016/j.patcog.2024.110532

Zoubir, H., Rguig, M., Aroussi, M. E., Chehri, A., Saadane, R. & Jeon, G. (2022). Concrete bridge defects Identification and localization based on classification deep convolutional neural networks and transfer learning. *Remote Sensing, 14*(19), 4882. doi:10.3390/rs14194882