Synthetic-to-Real Domain Adaptation with Virtual Laser Scanning and Self-Training–Based Category-Aware Cuboid Mixing for Semantic Segmentation of Bridge Point Clouds

Xiaofei Yang, Ph.D.[1]; Yuguang Fu, Ph.D., M.ASCE[2]; and Jinwoo Kim, Ph.D., A.M.ASCE[3]

Abstract: A scarcity of real-world point clouds poses a considerable challenge in training a bridge semantic segmentation model. Although virtual point cloud synthetization offers a promising alternative, the persistent domain gap between synthetic and real-world data remains a critical obstacle. To address this, we present a synthetic-to-real domain adaptation method that integrates virtual laser scanning (VLS) and self-training-based category-aware cuboid mixing (ST-CACM). Our experimental evaluation demonstrates the method's effectiveness in bridge semantic segmentation through comparison with models trained on real-world point clouds and traditional synthetic point clouds. The proposed approach achieves an overall accuracy of 95.34%, a mean class accuracy of 94.31%, and a mean intersection over union of 89.83%, demonstrating performance comparable to that of the baseline models while significantly reducing the dependency on real-world training data. Notably, both core components, VLS and ST-CACM, effectively mitigated the domain gap between synthetic and real-world data, establishing a robust framework for synthetic-to-real domain adaptation in bridge segmentation tasks. These findings will advance the reconstruction of digital twins and the efficient operations and maintenance of in-service bridges. **DOI: 10.1061/JCCEE5.CPENG-6828.**

Author keywords: Synthetic point cloud generation; Virtual laser scanning; Self-training; Domain adaptation; Semantic segmentation.

Introduction

Bridges play a critical role in transportation networks, yet they are prone to accelerated deterioration and corrosion due to environmental and operational stresses (Moselhi et al. 2017). Timely bridge inspections and early warning systems are therefore essential for ensuring the structural integrity and functional safety of aging bridge infrastructure (Kong and Frangopol 2003; Yang et al. 2022b). With regard to this, digital twins (DTs) are revolutionizing infrastructure management by enabling dynamic, real-time representations of physical systems to support data-informed, optimal decision-making (Jiang et al. 2021). However, their adoption during the operation and maintenance phase—the longest and most critical stage of an asset's life cycle—remains limited (Pan et al. 2024). A critical barrier is the lack of high-fidelity geometric digital twins (gDTs), which can integrate component-level three-dimensional (3D) geometry with surface textures and defect annotations to create semantically rich virtual models, enabling automated bridge condition assessment and predictive maintenance.

The gDT construction workflow typically involves two sequential stages: (1) semantic segmentation of large-scale point clouds to classify bridge components and (2) generation of gDTs by fitting Industry Foundation Class (IFC) entities and spatial relationships with segmented bridge components (Lu and Brilakis 2019). Yet, manually constructing gDTs from point clouds is notoriously inefficient. This bottleneck has spurred research into automating bridge information modeling (BrIM) reconstruction processes (Mehranfar et al. 2024). Particularly, semantic segmentation of bridge point clouds persists as a technical bottleneck due to challenges in handling complex geometries (Yang et al. 2024).

In this context, deep learning technologies have emerged as promising solutions for 3D semantic segmentation across diverse bridge types (Yang et al. 2022b). However, the practical implementation of deep learning–based 3D semantic segmentation faces a significant challenge: the scarcity of annotated real-world training point clouds (Yang et al. 2023). This limitation is particularly critical (Kim and Chi 2021; Kim et al. 2020b) given that collecting and annotating real-world bridge point clouds is time-consuming, laborious, and costly (Lu et al. 2019). Moreover, obtaining real-world point clouds representing the full spectrum of bridge types with varying geometric design features is practically infeasible (Xia et al. 2022). To address these limitations, synthetic bridge point cloud generation has become an alternative, which can automatically simulate and label a wide spectrum of 3D virtual bridge models in a digital environment.

Earlier studies have demonstrated that augmenting limited real-world training data with synthetic samples enhances the accuracy of 3D semantic segmentation tasks (Ma et al. 2020; Yang et al. 2022b, 2023). However, semantic segmentation models trained on such hybrid data sets—composed of both synthetic and real-world

[1]Postdoctoral Researcher, School of Civil and Environmental Engineering, Nanyang Technological Univ., 50 Nanyang Ave., Singapore S639798. Email: xiaofei.yang@ntu.edu.sg

[2]Assistant Professor, School of Civil and Environmental Engineering, Nanyang Technological Univ., 50 Nanyang Ave., Singapore S639798. Email: yuguang.fu@ntu.edu.sg

[3]Assistant Professor, Dept. of Civil and Environmental Engineering, Hanyang Univ., 222 Wangsimni-ro, Seoul 04763, Republic of Korea (corresponding author). ORCID: https://orcid.org/0000-0002-2237-4965. Email: jinwookim@hanyang.ac.kr

Note. This manuscript was submitted on January 30, 2025; approved on August 7, 2025; published online on November 24, 2025. Discussion period open until April 24, 2026; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

data—consistently exhibited performance inferior to those trained on an equivalent amount of purely real-world data. This performance gap stems from domain gaps—discrepancies in visual and geometric characteristics between synthetic and real-world data (Kim et al. 2023). A major reason is that traditional sampling-based synthetic data often exhibit uniform density and completeness, contrasting sharply with real-world data that typically features occlusions, noise, and uneven point distributions. These inconsistencies degrade model generalizability because models trained on synthetic data struggle to interpret the irregular characteristic of real-world scans.

Domain adaptation can help reduce distribution discrepancies between synthetic and real-world data by transferring knowledge from a labeled source domain to an unlabeled target domain (Xu et al. 2024). A major class of domain adaptation methods for point clouds involves using generative adversarial network (GAN)–based domain adversarial learning to align synthetic and real-world data distributions, particularly in indoor building point clouds (Hu et al. 2024). However, GAN-based approaches for closing the domain gap are notoriously unstable, prone to model collapse, and tend to blur class boundaries—ultimately degrading segmentation accuracy. This limitation is particularly critical for the semantic segmentation of complex structures like bridges, where precise boundary segmentation is essential for accurate digital twin reconstruction (Yang et al. 2023). In contrast, self-training has proven effective for the semantic segmentation of complex outdoor point clouds (Yang et al. 2021, 2022a; Zou et al. 2021). Yet, its effectiveness relies heavily on the quality of pseudolabels for unlabeled real-world data. Without careful design, directly combining synthetic (source) and real-world (target) data may introduce bias toward dominant data sources and exacerbate distribution discrepancies.

In addition, several studies have highlighted the advantages of using dense 3D voxels over raw point clouds to learn domain-invariant features to reduce domain gaps (Rist et al. 2019), and others have shown that strategically mixing source and target data can help create intermediate domains with reduced domain gaps (Saltori et al. 2022; Xiao et al. 2022). These insights suggest that integrating 3D voxel representations and source–target data mixing strategies has strong potential to improve pseudolabel quality, thereby enhancing the robustness and stability of self-training. However, no prior work has unified self-training, 3D voxelization, and synthetic–real data mixing into a single framework to mitigate domain gaps. Therefore, developing a robust strategy that harmonizes these components remains a critical open challenge in bridge semantic segmentation.

To address the challenge, we propose a synthetic-to-real domain adaptation method integrating virtual laser scanning (VLS) simulation and self-training–based category-aware cuboid mixing (ST-CACM). Specifically, VLS simulates real-world laser scanner physics characteristics to generate virtual point clouds with realistic occlusions and visibility constraints. These VLS-generated synthetic data augment limited real-world data, forming a hybrid training data set. ST-CACM framework incorporates three synergistic components—self-training, cuboid mixing, and category-aware oversampling—to further close the domain gaps. In the self-training process, a semantic segmentation model pretrained on the hybrid data set generates initial pseudolabels for unlabeled real-world cuboids. High-confidence pseudolabels are iteratively refined and incorporated into training, progressively adapting the model to real-world variations.

CACM partitions both the hybrid training data set and real-world testing data set into 3D cuboids, which are strategically mixed during self-training iterations to construct an intermediate domain that more closely aligns with real-world data distributions. To address class imbalance in pseudolabeling, underrepresented classes (e.g., parapets) are oversampled during cuboid mixing. This hybrid approach will help model trained on synthetic point clouds to be more generalizable and scalable in real-world field conditions. To validate the proposed method and its components, we conduct three experiments to address the following questions:

- How does our domain adaptation method perform compared with models trained exclusively on real-world data sets?
- What is the effectiveness of ST-CACM in reducing the domain gap and improving segmentation performance?
- How significantly does VLS contribute to domain gap reduction and model enhancement?

This research makes several significant contributions to the field. First, we demonstrate that deep learning models trained with the proposed hybrid approach can achieve performance comparable to that of models trained exclusively on a large real-world data set. Second, we validate ST-CACM's effectiveness in bridging the domain gap between synthetic and real-world point clouds, resulting in enhanced model performance. Finally, we establish that our VLS-based synthetic point cloud generation method outperforms traditional sampling-based techniques for 3D semantic segmentation tasks.

Literature Review on 3D Semantic Segmentation for Civil Infrastructures

Deep Learning–Based 3D Semantic Segmentation Methods

There have been extensive studies to investigate deep learning methods for semantic segmentation of bridge point clouds. A pioneering work (Kim et al. 2020a), for example, delved into the use of the PointNet (Qi et al. 2017) algorithm for automated recognition of bridge components from 3D point clouds. Given that the PointNet algorithm was originally designed to process small-scale point clouds with a fixed input point number of 2,048, they initiated the processing by partitioning full-scale bridge point cloud data into a set of discrete subspaces at a fixed distance interval. Their study demonstrated that deep learning–based methods can be a promising solution to automatically segment diverse types of bridges, including curved and tilted ones.

Other researchers further explored different types of deep learning–based algorithms, such as Point Convolutional Neural Network (PointCNN) and Dynamic Graph Convolutional Neural Network (DGCNN). One study (Kim and Kim 2020) compared three algorithms of PointNet, PointCNN, and DGCNN and showed that DGCNN outperformed the other two algorithms for semantic segmentation. In a later study (Lee et al. 2021), a graph-based hierarchical DGCNN algorithm was presented for the semantic segmentation of railway bridges to improve model's performance, particularly for railway-specific components (e.g., electric poles). This method incorporated a *k*-nearest neighbors (kNN) concept (Zhang et al. 2017) to expand the scope of local feature acquisition, resulting in a 3% improvement of segmentation accuracy for electric poles.

Although the aforementioned methods have showed promising results, they were optimally designed only for small-scale point clouds with a limitation of the number of input points (e.g., 2,048), inappropriate to segment large-scale bridge point clouds. To respond to this challenge, a recent study (Jing et al. 2022) proposed a 3D deep learning neural network called BridgeNet to segment large-scale masonry arch bridge point clouds. The authors employed

an encoder-decoder structure to efficiently extract features from point clouds at distinct resolutions.

In addition to BridgeNet, a weighted superpoint graph (WSPG) network (Yang et al. 2022b) was proposed to better perform the semantic segmentation of bridge components from large-scale point clouds. The initial step of WSPG was to cluster full-scale bridge point clouds into numerous superpoints through the application of geometric partitioning method, while maintaining both semantic homogeneity and geometric simplicity in each superpoint. Subsequently, these superpoints were fed into PointNet and a graph neural network to group superpoints with identical semantics (e.g., superpoints in the same type of bridge components) and classify them into distinct component categories (e.g., girders). The proposed WSPG algorithm, in particular, was able to alleviate the data imbalance between minor (e.g., parapets) and major bridge components (e.g., girders).

However, erroneous segmentation often occurred at the boundary between adjacent bridge components, despite great progress achieved. This motivated the authors to further explore the use of graph-structured deep metric learning to generate higher quality superpoints and mitigate such segmentation errors (Yang et al. 2023). Their method was designed to construct each superpoint composed only of point clouds associated with a single bridge component, thereby preserving homogeneous semantic information and ensuring the quality of superpoints.

Synthetic Point Cloud Generation Methods

Although deep learning–based methods hold promise for automating the semantic segmentation of various types of bridges, their widespread adoption within the infrastructure sector has been hampered. This limitation mainly arises from the lack of large-scale publicly available annotated point clouds. To overcome this limitation, synthetic point cloud data have recently attracted an increasing attention to remedy the scarcity of real-world point clouds, because synthetic point clouds can be an useful resource for deep neural networks to learn basic descriptors and geometric structures, such as planes, corners, and edges (Winiwarter et al. 2022).

An early example (Ma et al. 2020) introduced a method for generating synthetic building point clouds from building information models (BIMs) by randomly selecting points on the objects' surfaces of BIMs. The experiment results validated the effectiveness of using synthetic point clouds for augmenting a small real-world point cloud data set; a significant improvement of 7.1% in intersection over union was achieved for the semantic segmentation of building interior point clouds. Although they showed that synthetic point clouds can serve as an effective data set augmentation method (i.e., enlarging a given real-world data set with synthetically generated point clouds), synthetic data were not able to completely replace real-world point clouds.

Similarly, a user-friendly framework (Yang et al. 2022b) was proposed for the generation of synthetic bridge point clouds from BrIMs using Autodesk Revit, ifcConvert, and CloudCompare. Nevertheless, these generated synthetic data needed a time-consuming manual annotation step for humans to semantically label the synthetic point clouds. To address this technical limitation, another study leveraged FME Workbench to generate fully annotated point clouds from BIMs (Zhai et al. 2022). In this work, three main tools were adopted including a BIM model reader, a point cloud writer, and a series of geometry-guided and attribute-guided transformers to implement the geometry representation extraction and semantic labeling.

Notwithstanding, the aforementioned synthetic point cloud generation techniques uniformly sample points on the surface of geometric models, resulting in unrealistic synthetic data in terms of completeness, smooth surfaces, and a uniform point density with a fixed-point interval. They therefore lack the realistic properties of point clouds obtained from real-world laser scanners, particularly with respect to occlusions, which is a common occurrence in infrastructure scenes. This eventually leads to a significant domain gap between synthetic and real-world point clouds, holding deep learning models back from achieving their full performance.

To address this issue, some researchers investigated a VLS simulation technique that emulates real-world laser scanners through computer graphics rendering algorithms. VLS (either stationary or mobile) with adjustable characteristics employs ray casting simulators' operations, making the distribution of point clouds more akin to reality. Furthermore, VLS can also improve the realism of synthetic point clouds in terms of occlusions, density variations, and noises. With this, some early attempts explored the use of BlenSor (Gschwandtner et al. 2011), which is an add-in function in Blender (Blender Foundation 2022), to generate synthetic point clouds from BIMs (Han et al. 2021; Ma et al. 2021). Although their synthetic data were more realistic than ones generated by traditional samplined-based methods, the data annotation step relied on either other labeled point clouds using the kNN algorithm (Larose and Larose 2014) or manual labeling.

Heidelberg LiDAR Operations Simulator ++ (HELIOS++) (Winiwarter et al. 2022), in the meantime, was also investigated to generate realistic synthetic bridge point clouds, where each bridge component was scanned separately to automate the data annotation process (Mafipour et al. 2022). Although it successfully labeled synthetic point clouds automatically, this one-by-one scanning approach could not simulate occlusion scenarios, resulting in complete synthetic data. For this reason, another recent study (Korus et al. 2023) introduced DynamoPCSim performing ray tracing to generate well-annotated synthetic point clouds from BIMs. The simulator was able to transfer the BIM's semantic data (i.e., labels) to the generated point clouds for automated labeling.

As such, there have been a few research efforts and advancements related to synthetizing photorealistic bridge point clouds, particularly by applying VLS simulation. However, existing point cloud synthetization techniques still have an underlying limitation: synthetic point clouds generated from as-designed models (e.g., BIM) are inherently different from real-world as-is structures due to unmodeled physical degradation, such as material fatigue, structural deformations, and surface irregularities. For example, real-world bridges often display physical defects and deformations that lead to geometric discrepancies compared with their as-designed models. In addition, vegetation, such as grass and moss, commonly grows on the bridge surface, resulting in the acquisition of rugged point clouds that starkly contrast with the smooth and clean data generated from as-designed models. These discrepancies in visual and geometric properties between synthetic and real-world data introduce significant domain gaps that hinder the generalizability of deep learning models trained on synthetic data. As evidence, a recent study showed that models trained exclusively on synthetic point clouds achieve a mean intersection over union (IoU) of just 41.1% on real-world scans (Zhang and Zou 2023).

Domain Adaptation Methods for Point Clouds

Domain adaptation addresses the challenge of transferring knowledge from a labeled source domain to an unlabeled target domain by reducing distribution discrepancies between them (Xu et al. 2024). This technique enables models trained on source data to generalize effectively to the target domain, ensuring robust

performance (Ding et al. 2022). Existing domain adaptation methods for point cloud data generally fall into two primary categories: adversarial learning approaches and self-training frameworks (Zhang et al. 2024). The former aligns source and target distributions through adversarial training, whereas the latter leverages iterative pseudolabeling to bridge the domain gap without explicit target supervision.

Adversarial learning approaches generally employ gradient reversal layers (GRLs) to align source and target feature distributions. By adversarially training a domain classifier while reversing gradients during backpropagation, GRLs force the feature encoder to learn domain-invariant representations that minimize the discrepancy between source and target domains, thereby improving cross-domain generalization (Bousmalis et al. 2017). An early example pioneered domain adaptation in 3D object detection by combining fine-grained local adaptation with adversarial global alignment, significantly improving LiDAR-based far-range detection (Wang et al. 2019).

Moreover, SegTrans was proposed as a domain adaptation framework for point cloud semantic segmentation, aiming to bridge the gap between labeled source and unlabeled target data sets (Shen et al. 2023). SegTrans achieves this through two key components: a data selection module (DSM), which mitigates data set discrepancies at the input level, and an adversarial learning module (ALM), which iteratively aligns domain-specific features through adversarial loss. Although adversarial learning has demonstrated success in aligning source and target domains, it faces a critical limitation: the absence of labeled target domain data. Without target labels, the training process becomes biased toward the source domain, forcing the model to overfit to source-specific patterns. This imbalance results in unreliable predictions and degraded performance when applied to the target domain.

Beyond adversarial learning, self-training has emerged as a highly effective alternative for domain adaptation (Saito et al. 2017). Initially popularized for image semantic segmentation (Zou et al. 2018), self-training uses labeled source domain data to train an initial model and then iteratively improves pseudolabels for the unlabeled target domain. Recent work has extended this framework to domain adaptation for 3D point clouds (Zou et al. 2021). For example, ST3D improved adaptation to object size variations in the target domain through data augmentation and introduced a quality-aware triplet memory bank to filter reliable pseudolabels (Yang et al. 2021). ST3D++ further addressed noise in pseudolabels by redesigning the entire self-training pipeline, from label generation to model updates, to reduce errors (Yang et al. 2022a). Density-insensitive teacher–student framework tackled density distribution mismatches between domains (e.g., sparse versus dense point clouds) with a density-insensitive training strategy, minimizing domain gaps caused by sampling differences (Hu et al. 2023). Previous work also highlights the benefits of dense 3D voxels over raw point clouds for learning domain-invariant features of point clouds (Rist et al. 2019) and mixing source–target data to create an intermediate domain with a smaller domain gap (Saltori et al. 2022; Xiao et al. 2022).

Despite significant progress, domain adaptation for civil engineering point clouds remains understudied. Although a pioneering work has explored the use of domain adversarial learning to bridge the gap between synthetic and real-world building point clouds (Hu et al. 2024), no prior studies have tailored to bridge semantic segmentation tasks in this field. Consequently, mitigating domain gaps to maximum the performance of semantic segmentation models trained on synthetic data in real-world applications remains an open challenge. Inspired by the success of self-training, 3D voxels, and data mixing in 3D outdoor domains, this study specifically designed a synthetic-to-real domain adaptation method integrating VLS simulation and ST-CACM for bridge semantic segmentation tasks.

Synthetic-to-Real Domain Adaptation and Semantic Segmentation

To bridge the gaps, we devise a synthetic-to-real domain adaptation method integrated with VLS simulation and ST-CACM, to alleviate the domain gap between synthetic and real-world bridge point clouds. Fig. 1 illustrates the overall pipeline of the proposed method. In the first stage, synthetic bridge point clouds are generated from BrIMs using a VLS tool called HELIOS++ (Winiwarter et al. 2022). In the second stage, ST-CACM is employed to perform self-training, during which CACM is integrated to merge the original labeled training data with the filtered pseudolabeled real-world data at each self-training iteration. Through this process, CACM constructs a cuboid-based intermediate domain between synthetic and real-world bridge point clouds, further reducing their domain gaps. After completing the two-stage domain adaptation, the point clouds are finally partitioned using a semantic segmentation model. Technical details of VLS, ST-CACM, and semantic segmentation are described in the following subsections.

Virtual Laser Scanning–Based Synthetic Point Cloud Generation

We first generate synthetic point clouds from a given BrIM using a VLS tool, called HELIOS++ (Winiwarter et al. 2022). This tool was selected because it has been proven effective at replicating the functionality of real laser scanners through the simulation of laser beams and the measurement of the corresponding reflection signals (Hämmerle et al. 2017; Kukko and Hyyppä 2009). It can thus generate synthetic point cloud data having a similar quality as real-world data, providing a precise approximation to physical realism and flexible computational requirements. It also offers extensive customization options, allowing users to tailor various aspects of the laser scanning simulation according to field requirements.

The HELIOS++ for VLS simulation encompasses the preparation of four important files, namely, scanner, scene parts, scenes, and surveys. The scanner file specifies a scanner's properties, such as scanning range, resolution, precision, and field of view, which are defined by users within an Extensible Markup Language (XML) file. A virtual laser scanner is configured as per the terrestrial laser scanner model FARO Focus X 330 (FARO Technologies Inc., Lake Mary, Florida), which is commonly used for the capture of real-world bridge point clouds. The scene parts files are constructed from a collection of geometric bridge components stored in separate Wavefront Object (OBJ) files. The scenes XML file references the scene parts files and can apply various parameters to specify the preprocessing procedures for these scene parts files, including translation, rotation, and scaling. The survey XML file defines the scanning positions, and scanner and platform settings are configured.

To generate synthetic data that closely resembles the real-world bridge point cloud data set collected with an authentic laser scanner, it is crucial to accurately replicate the scanner's hardware characteristics and measurement behaviors. The simulation parameters, carefully configured to align with the authentic laser scanner used for the real-world bridge point cloud data set collection (Lu et al. 2019), are detailed in Table 1. Here, the files of scene parts, scenes, and surveys can be automatically generated with the assist of the Blender2Helios (Neumann et al. 2022) plugin.

The specific workflow of generating well-annotated VLS-based synthetic bridge point clouds from prebuilt BrIMs is shown in

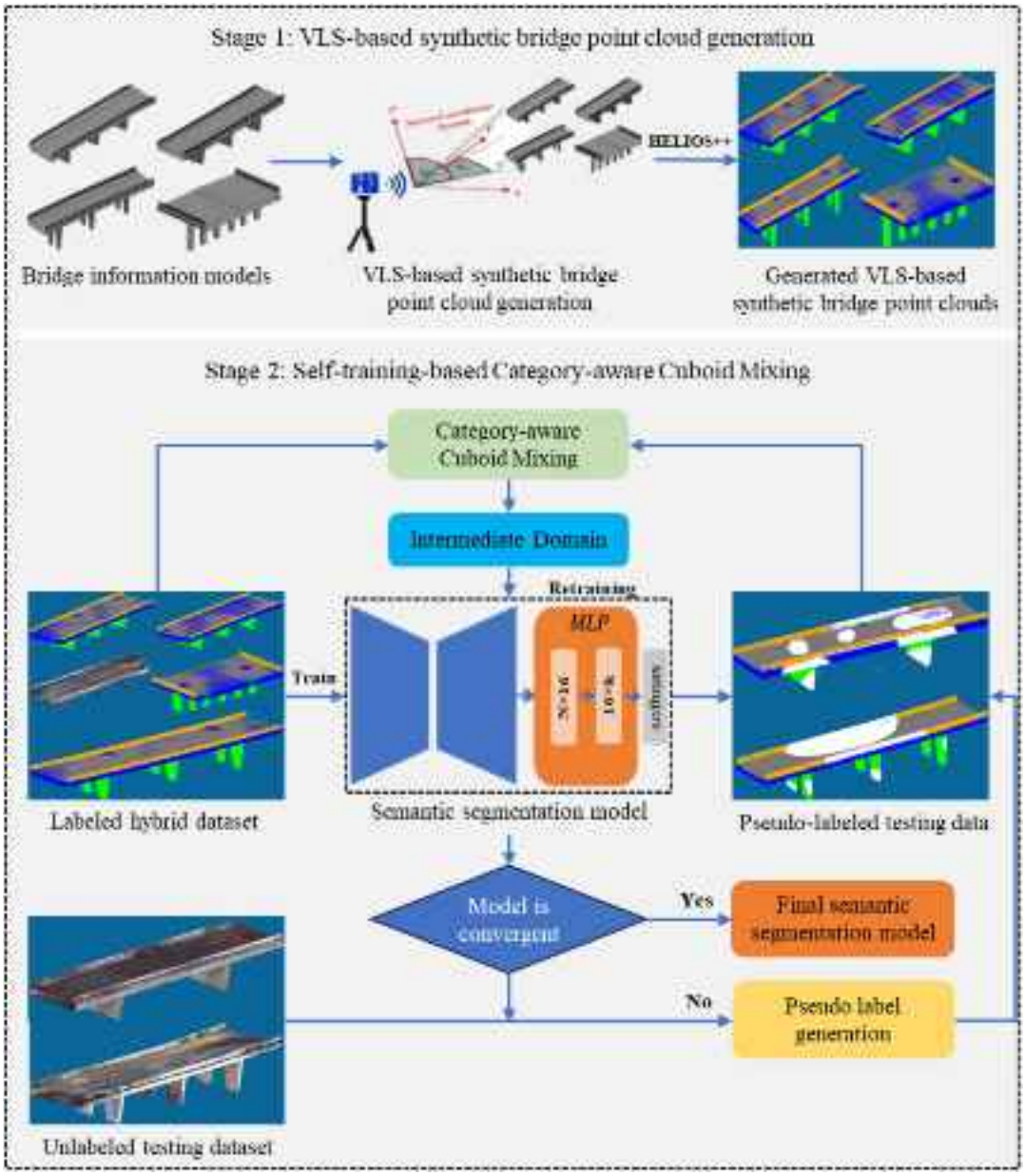

Fig. 1. Overall pipeline of the proposed synthetic-to-real domain adaptation.

Table 1. HELIOS++ simulation parameter settings

Parameter settings	Value
Pulse frequency	122,000 HZ
Scan frequency	60 HZ
Ranging error	±2 mm
Beam divergence	0.19 mrad
Wavelength	1,550 nm
Vertical field of view	300°
Horizontal field of view	360°
Head rotation	10°/s

Fig. 2(a). Initially, the BrIMs are manually reconstructed from a publicly available real-world bridge point cloud benchmark data set (Lu et al. 2019) using Autodesk Revit 2024 (Autodesk 2024) as the authoring tool. This reconstruction process ensures that subsequent synthetic point clouds, generated via virtual laser scanning, maintain geometric consistency with the real-world data while isolating domain-specific variations (e.g., noise). The reconstructed BrIMs are then exported into intermediate files with common 3D exchange format (*.fbx or *.ifc format).

For bridge scene preparation, these intermediate files with *.fbx or *.ifc format are imported into Blender 3.2 (Blender Foundation 2022) either directly for *.fbx files or through specific plug-in BlenderBIM for *.ifc files. Once imported into Blender, the bridge components are organized into a structured scene and exported into separate folders using the Blender2Helios (Neumann et al. 2022) plug-in, with each folder representing a specific bridge component class. Within these folders, individual bridge components are saved as OBJ files. An XML scene file is also automatically generated, referencing the corresponding bridge components in the scene parts.

Next, the scanning positions are set to replicate the scanner configurations of the real-world benchmark data set collection workflow. During VLS simulation, the virtual scanner emits rays that mimic real-world laser scanner configurations. These rays are projected from predefined scanner positions and trace trajectories until

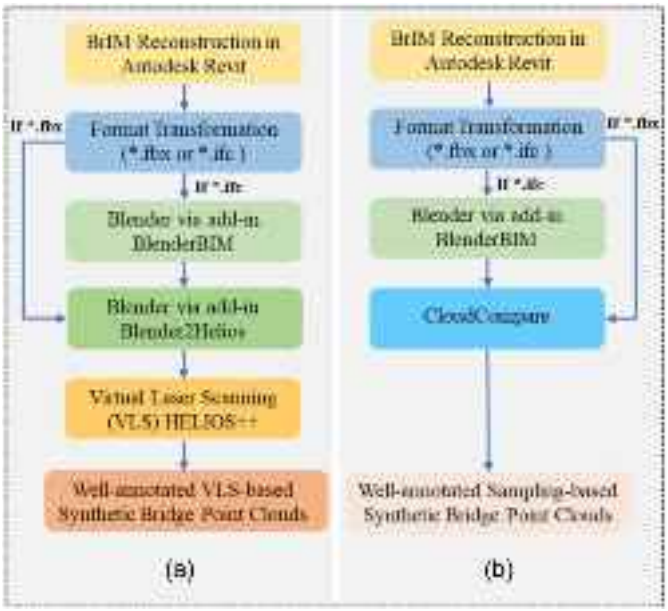

Fig. 2. Workflows of VLS-based (ours) and sampling-based (existing) synthetic bridge point cloud generation: (a) VLS-based synthetic data generation; and (b) sampling-based synthetic data generation.

intersecting with bridge components. Each intersection is recorded as a 3D coordinate, forming the point in the synthetic point cloud. The generated point inherits the semantic class of its corresponding bridge component automatically that are preassigned semantic labels (e.g., decks, girders, and piers) during the 3D scene preparation phase. For example, points intersecting with a bridge deck receive the Deck label, and those hitting the pier surface are labeled Pier. This ensures an error-free annotation process that is reproducible and scalable. The output of the virtual laser scanner consists of a single point cloud per individual scan, akin to the process of conducting a real laser scan. These can be seamlessly merged to form a complete point cloud model. Given that each point cloud is provided in a consistent global coordinate system, registration is unnecessary. The resulting point clouds are conventionally stored in an XYZ file format, an open ASCII format designed for the storage of point cloud data.

Self-Training–Based Category-Aware Cuboid Mixing

Despite the enhanced realism of VLS-generated synthetic point clouds compared with existing sampling-based generation method [Fig. 2(b)], the geometric and visual discrepancies with real-world bridges can still exist. To this end, we build upon a ST-CACM algorithm—an integrated system structured as a closed-loop feedback cycle that combines three synergistic components: ST, cuboid mixing (CM), and category-aware (CA) oversampling—to further bridge the domain gaps between the two different data sources. ST is a semisupervised learning technique that can better capture and learn the distributional characteristics of a given data set by leveraging both labeled and unlabeled data, improving a model's performance. Technically, given training point clouds can be considered as the source data, and real-world testing point clouds can be regarded as the target data. In this scheme, ST generates pseudolabels for the target (testing) scenes and utilizes both source (training) data and pseudolabeled target (testing) data for model training. However, the direct use of source data with strong labels and target data with weak (pseudo) labels may introduce domain bias, which can rather pose an adverse threat to semantic segmentation.

To mitigate this problem, a CACM method, mixing strongly labeled source data with weakly labeled target data in the self-training process, is applied to establish an intermediate domain D_m and avoid the domain bias. Particularly, we devise a cuboid-level source and target data mixing algorithm, which augments and rectifies source data features by incorporating features from the target domain. This enables the generated intermediate domain to become closer to the target domain compared with the source domain. Additionally, category-aware oversampling ensures balanced pseudolabel distributions by prioritizing underrepresented classes (e.g., parapets) during cuboid mixing. Thus, a semantic segmentation model trained on the newly established, intermediate domain can perform better on target data. Details on the ST strategy and CACM are given next.

Self-Training Strategy

The core idea of ST strategy is to iteratively train a semantic segmentation model using both (1) an accurately labeled hybrid training data set (a small portion of real-world point clouds and a larger portion of VLS-based synthetic point clouds), and (2) pseudolabeled testing data set. A step-by-step breakdown of how self-training works is as follows:

1. Initial model training: Start with training a base semantic segmentation model with the labelled hybrid training data set to learn patterns from the labeled data.
2. Pseudolabel generation: Use the trained model in the previous step to predict pseudolabels for an unlabeled testing real-world data set. Following the previous approaches (Rosenberg et al. 2005), pseudolabels are acquired with Eq. (1):

$$\tilde{Y}_{i,j}^{t} = \begin{cases} 1, & \text{if } \max(S_i^t) > T, j = \arg\max S_i^t \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$S_i^t = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}} \quad \text{for } i = 1, 2, \ldots, c \tag{2}$$

where $\tilde{Y}_{i,j}^{t} = [\tilde{Y}_{i,1}^{t}, \ldots, \tilde{Y}_{i,c}^{t}]$ stands for the generated pseudolabels; c = number of classes; S_i^t = confidence score, which indicates the model's predicted probability of a sample belonging to each class and is derived by applying the softmax function to the raw logits z_i [Eq. (2)], which are the prenormalized outputs for each class; and T = confidence threshold that defines the minimum score required to retain a prediction, filtering out uncertain results.
3. High-confidence prediction selection: Filter predictions using a confidence threshold (e.g., softmax probability >0.9) to retain only high-quality pseudolabels. Points with lower confidence are excluded and visually represented in Fig. 1.
4. Training data expanding: Combine the original labeled hybrid training data set with the filtered pseudolabeled real-world testing data set.
5. Model retraining: Train a new semantic segmentation model on the expanded data set.
6. Iterative refinement: Repeat Steps 2 and 3 for multiple iterations until convergence or a predefined stopping criterion is met.

We set the self-training epochs to 100 based on a previous experiment (Ding et al. 2022). The reasons for fusing a hybrid data set are (1) training a bridge semantic segmentation model with the hybrid data set can alleviate the real-world data scarcity issue; and (2) the model trained on the hybrid data set is supposed to outperform the one trained on a purely synthetic data set because the domain of the hybrid data set is closer to that of the real-world testing

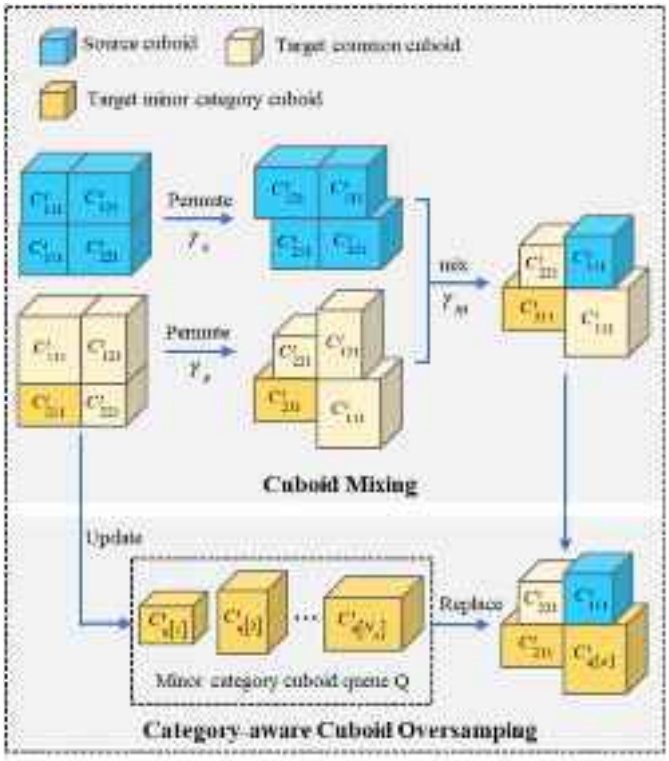

Fig. 3. CACM method using $(d_x, d_y, d_z) = (2, 2, 1)$ as an example.

data. This proximity boosts the generation of higher quality pseudolabels for the unlabeled testing data set.

Category-Aware Cuboid Mixing

CACM strategically merges the original labeled training data with filtered pseudolabeled real-world data during self-training iterations to construct a cuboid-based intermediate domain between the training and testing data sets for alleviating their domain gap. The intermediate domain is then leveraged to retrain a sparse-convolution-based U-Net algorithm, enabling us to improve the performance of bridge semantic segmentation. A detailed explanation of the CACM implementation process is provided next, as illustrated in Fig. 3.

Cuboid Mixing

Cuboid mixing is implemented to create a new intermediate domain D_m by strategically blending the VLS-based hybrid data set with real-world point cloud data. Once a target bridge scene (testing data in this study) is given, a source bridge scene (training data) is randomly selected. Both of the target and source scenes are then segmented by varying dimensions [Eq. (3)] to generate a set of cuboids, which serve as the smallest indivisible unit for cuboid mixing. Their segmentation positions are first evenly distributed, and then randomness is introduced to increase diversities [Eq. (4)]

$$P = \{\text{cuboid}_{ijk}\}, \quad i \in \{1, \ldots, d_x\}, j \in \{1, \ldots, d_y\}, k \in \{1, \ldots, d_z\}$$
$$\text{cuboid}_{ijk} = \{p | p\text{in}[x_{i-1}, y_{j-1}, z_{k-1}, x_i, y_j, z_k]\} \tag{3}$$

where P = bridge scene, which is a collection of multiple cuboids of different sizes; d_x, d_y, and d_z = number of segments along the x-, y-, and z-axis, respectively; Each cuboid cuboid_{ijk} is defined by a six-tuple bounding box $[x_{i-1}, y_{j-1}, z_{k-1}, x_i, y_j, z_k]$, where x_{i-1}, y_{j-1}, and z_{k-1}, respectively, denote the coordinates of the center of the bounding box, and x_i, y_j, and z_k represent the width, height, and depth of the bounding box, respectively. Here, x_i, y_j, and z_k also stand for the segmentation positions along the corresponding dimensions

$$x_i = \begin{cases} \dfrac{i}{d_x}\max p_x + \left(1 - \dfrac{i}{d_x}\right)\min p_x, & \text{if } i \in \{0, d_x\} \\ \dfrac{i}{d_x}\max p_x + \left(1 - \dfrac{i}{d_x}\right)\min p_x + \Delta\delta, & \text{otherwise} \end{cases} \tag{4}$$

where $\Delta\delta$ = value of random perturbation with a uniform distribution ranging from $-\varepsilon_\delta$ to ε_δ. The same equation is also applied to y_j and z_k.

After segmenting the selected source and target bridge scenes into cuboids with different dimensions, these cuboids first undergo spatial permutation with a probability γ_s. They, are then randomly swapped between the source and target scenes with one another (probability γ_m) to create a hybrid representation, i.e., intermediate domain. This mixing-based domain adaptation would be useful in reducing the geometric and visual discrepancies between training and testing point clouds, improving segmentation performance.

Category-Aware Oversampling

Following the aforementioned cuboid mixing, a category-aware oversampling strategy is applied to adjust category distributions in the mixed point cloud data. This can moderate the category imbalance of pseudolabels during the self-training process, wherein minor categories of bridge components (e.g., parapets) are underrepresented within the pseudolabeled data. To ensure that minor categories of cuboids are adequately represented, we prioritize sampling cuboids on minor categories more frequently, by guaranteeing at least v minor cuboids contained in the mixed bridge scenes. Here, v is set to 2 as per the previous study (Ding et al. 2022). The details are described as next.

The per-category pseudolabel ratio $r \in [0, 1]^c$ is initially computed as per Eq. (5)

$$r_c = \frac{n_l^c}{N_l} \tag{5}$$

where c = specific bridge component category; n_l^c = number of pseudolabels for a specific bridge component category; and N_l = total number of pseudolabels for all categories of bridge components. The least common categories are designated as the minor categories. Subsequently, the minor cuboids are identified as those with a pseudolabel ratio higher than the average value r_m on at least one of the minor categories. A queue Q of minor cuboids is then constructed with a capacity of N_q to store these cuboids. Specifically, $\text{cuboid}^q[w]$ stands for the wth minor cuboid in Q. It is worth noting that Q undergoes dynamic updates using a First In, First Out rule because cuboids are randomly split in each iteration according to Eq. (3). During each training iteration, the cuboids in Q are sampled and replace existing cuboids if necessary to ensure at least v minor cuboids contained in the mixed bridge scenes. Through this straightforward oversampling strategy, category-awareness is incorporated into the cuboid mixing process, thereby alleviating the data imbalance issue in the self-training phase.

Bridge Point Cloud Semantic Segmentation

The aforementioned ST-CACM is built upon a bridge point cloud semantic segmentation method. Bridge semantic segmentation aims to classify components and assign a semantic label to each point within the whole point cloud model. In this study, a

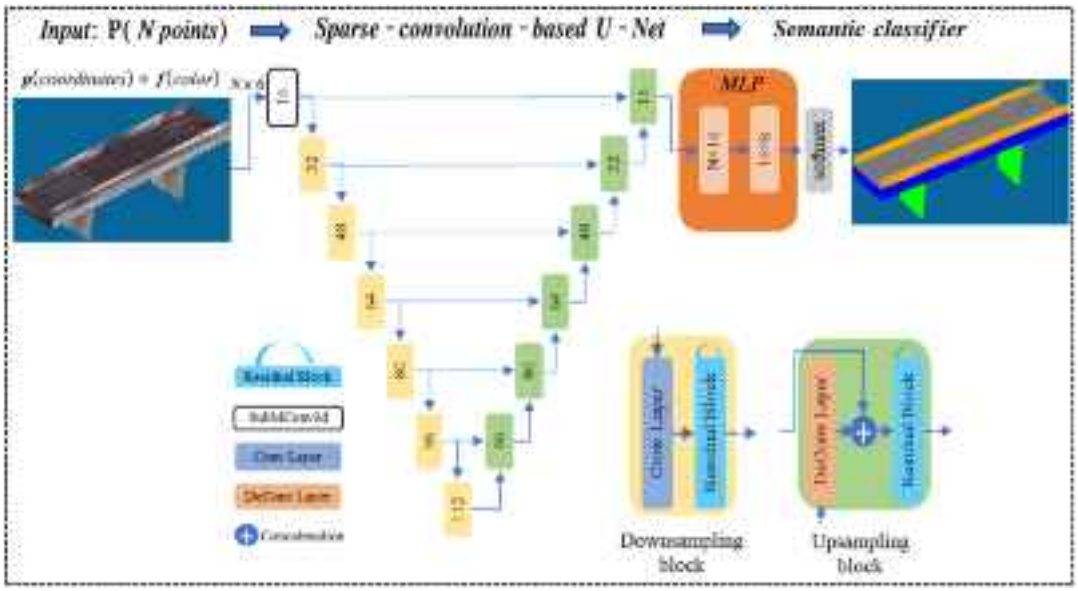

Fig. 4. Framework of the sparse-convolution-based U-Net method for bridge semantic segmentation task.

sparse-convolution-based U-Net method (Choy et al. 2019; Jiang et al. 2020) is selected to perform a bridge point cloud semantic segmentation task owing to its high performance and capacity of processing large-scale point clouds. The framework of the sparse-convolution-based U-Net method is presented in Fig. 4.

The sparse-convolution-based U-Net takes a point cloud, P, as input, which is represented as N points consisting of 3D coordinates and red, green, blue (RGB) color information. The input point cloud is unordered, wherein each point exists independently without any explicit connection or order. To address this, the input point cloud is first voxelized by assigning each point to a specific voxel in the voxel grid, converting the unordered points into ordered volumetric grids. These grids are then transferred to a 3D U-Net style backbone architecture with a depth of 7. Within the 3D U-Net backbone, a submanifold sparse convolution and sparse convolution is firstly employed to extract voxel-based features. Subsequently, these extracted features are processed in downsampling and upsampling blocks, as shown in Fig. 4. The downsampling block includes a convolution (Conv) layer and a residual block, and the upsampling block is composed of a deconvolution (DeConv) layer and a residual block, with a residual connection and concatenation operation. The extracted voxel-based features are then mapped into point-based features. The output of the backbone is then passed through a semantic classifier, designed with a multilayer perceptron (MLP) to produce the semantic predictions.

Experiment and Results

We conducted three experiments to answer the research questions. In each experiment, we assembled the training data sets differently depending on the research questions, whereas the same real testing point clouds were adopted in all cases for a fair comparison. Specifically, we basically adopted a publicly available real-world bridge point cloud data set (Lu et al. 2019) and extracted seven concrete slab bridges to secure sufficient training samples and the representativeness of training and testing data sets; these seven bridges contained four types of components, i.e., piers, girders, decks, and parapets. Among the seven bridges, two of them were allocated into the testing data set, and the other five were assigned to the real-world training data set. We then selected one bridge's point clouds from the real-world training data set and hybridized it with (1) VLS-generated synthetic data set (ours), and (2) traditional sampling-based synthetic data set (existing). Both synthetic data sets were built to replicate the other four bridges that were included in the real-world training data set (Fig. 2).

This hybrid data set setting enabled us to rigorously evaluate the supplementary effects of synthetic point clouds in real-world application scenarios, wherein some real-world training data are available, but their quantity and diversity are limited—an increasingly realistic condition in infrastructure inspection tasks (Yang et al. 2022b). By pushing the extreme case where 80% of training examples (four out of five bridges) are synthetic, we can directly measure how well our method performs when synthetic data dominate, yet are still anchored by a minimal amount of real-world input. Fig. 5 shows the visualization examples of point cloud data sets.

The data sets were used to train and test semantic segmentation models in each experiment, and a total of four different evaluation metrics were adopted for comprehensive and rigorous evaluation. These metrics included overall accuracy (OA), mean class accuracy (mAcc), and mean intersection over union (mIoU) for overall evaluation, and IoU for component-level evaluation. Technically, OA provides a general indication of how well a model performs across all classes, but it can be misleading in imbalanced data sets by favoring the majority class. In contrast, mAcc offers a balanced view of the model's performance by averaging accuracy across all classes, which is beneficial in addressing class imbalance. However, mAcc can be skewed by classes with few samples and does not assess prediction quality (i.e., how precise or specific they are).

In this sense, mIoU provides a detailed evaluation of segmentation quality by averaging the spatial overlap between predicted and ground truth segments for all classes. Despite its strengths, mIoU's effectiveness can be heavily influenced by point density and distribution variations, especially in sparse or unevenly sampled areas. These factors can lead to biased evaluations in different regions of the point cloud. Similarly, IoU assesses how well the predicted segmentation masks align with the ground-truth labels for a specific class, particularly accurately measuring boundary segmentation quality. Small segmentation errors on the boundaries can have drastic consequences in the later digital twin construction

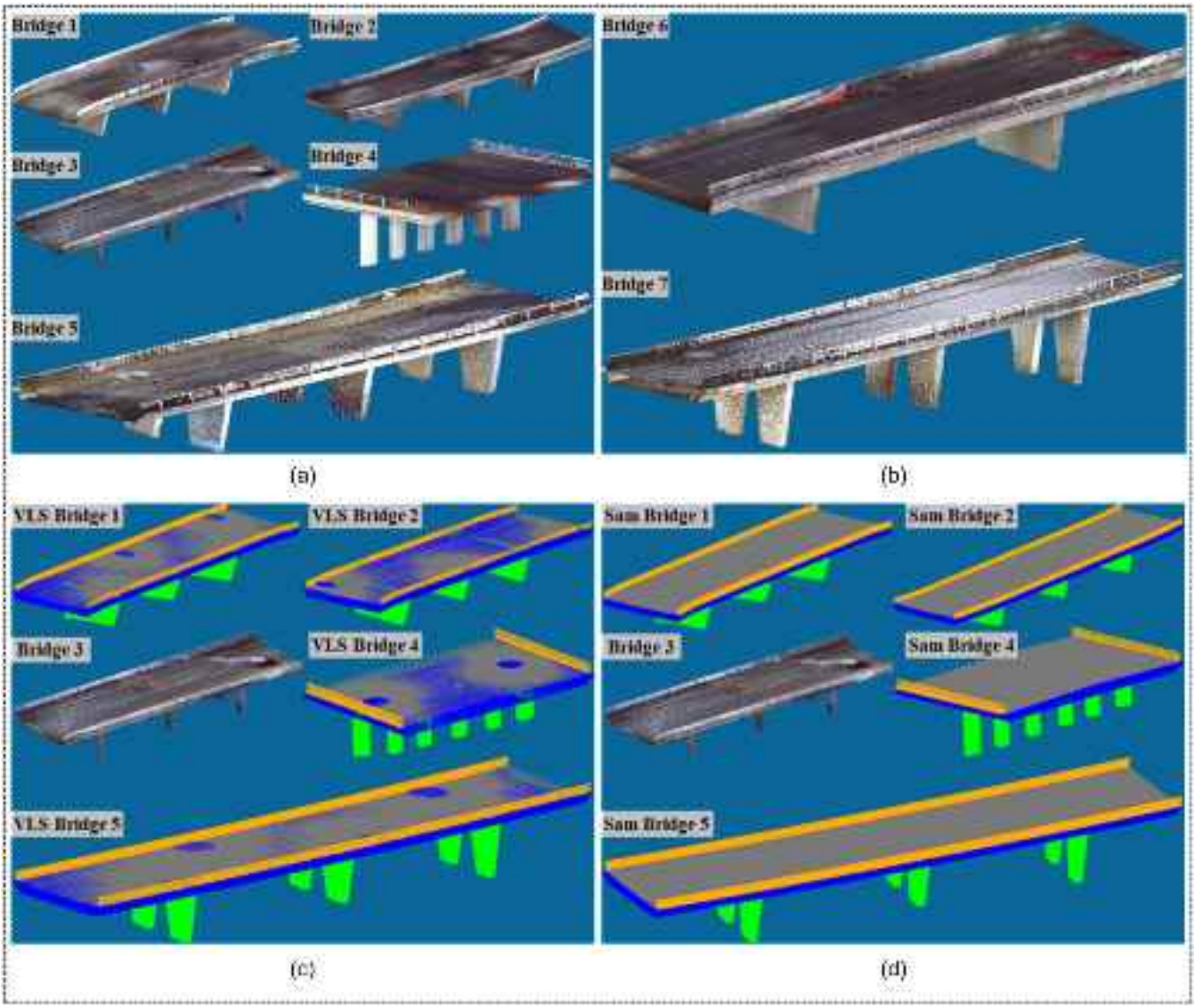

Fig. 5. Visualization of the established data sets: (a) real-world training data set; (b) real-world testing data set; (c) VLS-based hybrid training data set; and (d) sampling-based hybrid training data set.

process. Therefore, IoU is suitable for evaluating component-level segmentation performance. These evaluation metrics can be computed with the following equations:

$$\mathrm{OA} = \frac{\sum_{i=1}^{n} p_{ii}}{\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}} \tag{6}$$

$$\mathrm{mAcc} = \frac{1}{n}\sum_{i=1}^{n}\frac{p_{ii}}{\sum_{j=1}^{n} p_{ij}} \tag{7}$$

$$\mathrm{mIoU} = \frac{1}{n}\sum_{i=1}^{n}\frac{p_{ii}}{\sum_{j=1}^{n} p_{ij} + \sum_{j=1}^{n} p_{ji} - p_{ii}} \tag{8}$$

$$\mathrm{IoU} = \frac{p_{ii}}{\sum_{j=1}^{n} p_{ij} + \sum_{j=1}^{n} p_{ji} - p_{ii}} \tag{9}$$

where n = number of bridge component categories; and p_{ij} = points of category i that are misclassified as category j.

In this setting, the sparse-convolution-based U-Net, described in the section "Bridge Point Cloud Semantic Segmentation," was trained and evaluated on an Ubuntu 20.04 system equipped with an Intel Xeon CPU and two NVIDIA 2080 Ti GPUs. Pytorch (Paszke et al. 2019) was chosen as the deep learning framework. In this computer environment, the voxel size was set to 0.1 m to balance the segmentation accuracy with computational complexity and training speed. Stochastic gradient descent was applied as the optimizer to minimize the overall training loss, with a batch size of four was selected. An initial learning rate was set to 0.01 with a stepwise learning rate decay. Lastly, the initial segmentation model was trained for 300 epochs, and the self-training was conducted for an additional 100 epochs. More detailed settings and results of each experiment are as follows.

Experiment 1: Evaluating the Effectiveness of the Proposed Method

Experiment 1 aims to evaluate the effectiveness of the proposed method (VLS + ST-CACM) in mitigating domain gaps and

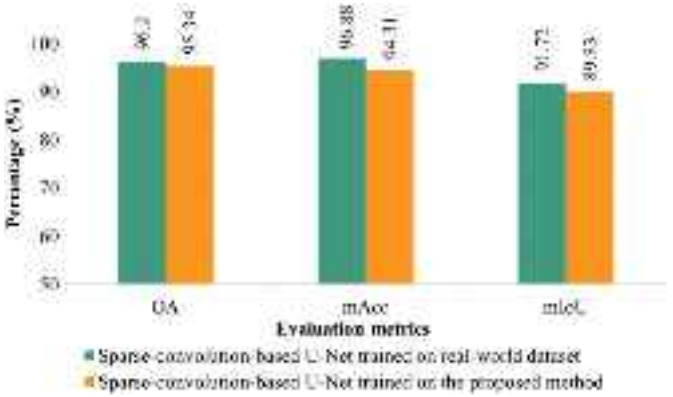

Fig. 6. Overall evaluation results between the model trained with a real-world data set (real-only) and the one trained with the proposed method (VLS + ST-CACM).

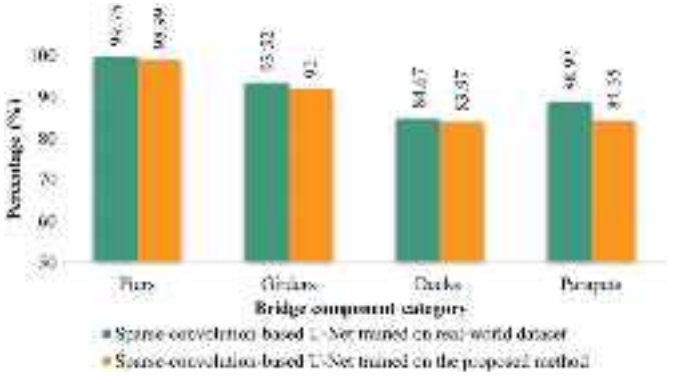

Fig. 7. Component-level evaluation results between the two models trained with a real-world data set (real-only) and the proposed method (VLS + ST-CACM).

enhancing generalization to real-world data in bridge segmentation tasks. To this end, we trained and compared two different sparse-convolution-based U-Net models: (1) real-only model, trained on an entirely real-world data set comprising point clouds from five concrete slab bridges [Fig. 5(a)]; and (2) proposed method (VLS + ST-CACM model), trained on a hybrid data set combining synthetic and real-world data [Fig. 5(c)].

To ensure a controlled comparison, we replaced four real-world bridge point clouds in the training data set with VLS-generated synthetic data while retaining one real-world bridge to establish the hybrid data set. Notably, the VLS-generated synthetic bridge point clouds preserved geometric consistency with real-world data. Additionally, ST-CACM was applied to this hybrid data set to build an intermediate domain between hybrid training and real-world testing data sets, effectively mitigating domain gaps from real-world data. Both models were then evaluated on a separate real-world test data set comprising two bridges [Fig. 5(b)]. This setting ensured that both models learned from an equivalent number of training point clouds (five real-world bridges versus one real-world bridge + four synthetic bridges) while maintaining geometric consistency, allowing us to isolate the domain-specific variations introduced by synthetic point clouds.

Fig. 6 presents the overall performance of the two models. When training a semantic segmentation model with the proposed method, its performance was comparable to that of the real-only model. Quantitatively, the VLS + ST-CACM model was able to achieve 95.34% for OA, 94.31% for mAcc, and 89.83% for mIoU, with only slight decreases of 0.86% for OA, 2.57% for mAcc, and 1.89% for mIoU. These results can be a strong signal that our domain adaptation method was able to create realistic synthetic point clouds and close the domain gap between synthetic and real data. This comparable performance is more noteworthy considering that the number of real-world bridges included in the training data sets was reduced from five to one; collecting and labeling real-world point clouds is extremely time-consuming and labor-intensive.

We also compared the component-level performances between the model trained with the real-world data set and that trained with the proposed method. As displayed in Fig. 7, the proposed method achieved a similar level of performance for piers, girders, and decks when compared with the model trained on the real-world data set. However, there was a relatively noticeable drop in performance for parapets, with a decrease of 4.57% in the IoU metric. This performance decline could be attributed to more incomplete areas caused by partial obstruction by other structures and nonoptimal scanning angles, which result from the parapet's elevated and edge positions.

Additionally, although the other components, such as piers, girders, and decks are flat and smooth, the irregular shapes of real-world parapets lead to higher noise levels in real-world parapet point clouds, compared with those of synthetically generated bridge point clouds. This result implies that, although our VLS-generated synthetic point clouds can mimic the noises and incompleteness caused by occlusion, the domain gap with real-world bridges could inevitably exist to some extent.

Experiment 2: Testing the Impacts of ST-CACM

In this experiment, we answer the second research question about the impacts of ST-CACM, one of our technical novelties in our domain adaptation method. To evaluate its overall impact, we conducted a comparative study between two semantic segmentation models trained with and without ST-CACM. To further assess the influence of ST-CACM, we examined how varying proportions of VLS-based synthetic data affected the performance of ST-CACM. In addition, an ablation study was performed to evaluate the contribution of each individual component within the ST-CACM framework. All models were evaluated on the real-world testing data set comprising two bridges, as used in Experiment 1 [Fig. 5(b)].

Fig. 8 exhibits the overall evaluation metrics of the two models with and without ST-CACM: (1) a VLS + ST-CACM model, and (2) another VLS-only model. Both models were trained with the same hybrid training data set [Fig. 5(c)], but ST-CACM was applied only for the VLS + ST-CACM model training. It can easily be seen that integrating ST-CACM improved all the evaluation metrics, with increases of 2.33% in OA, 0.92% in mAcc, and 2.47% in mIoU, compared with the case without ST-CACM. These results show the beneficial impacts of ST-CACM on bridge segmentation performance. Although the overall improvements are relatively modest, they are significant within the context of domain adaptation tasks, where even small gains can be critical for practical applications.

Meanwhile, we examined in detail the impacts of ST-CACM on component-level segmentation performance. As shown in Fig. 9, when training a semantic segmentation model using the proposed method with ST-CACM, superior performance was achieved compared with the one trained without ST-CACM across most bridge component categories. Notably, while maintaining the comparable performance for piers and parapets, the performance for girders and decks was increased by 4.02% and 6.80%, respectively, resulting in a significant increase of 2.98% across all categories

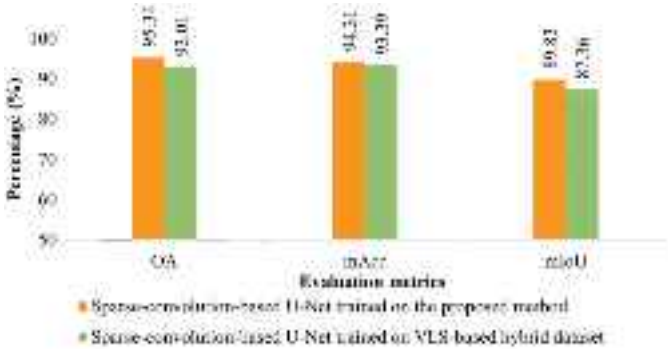

Fig. 8. Overall evaluation results for the two models trained with and without ST-CACM.

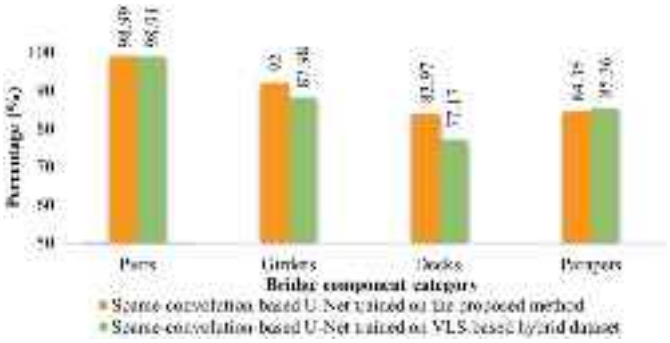

Fig. 9. Component-level evaluation results between the two models trained with and without ST-CACM.

on average. However, a marginal decline in parapet performance was observed when training with ST-CACM. This is attributed to real-world parapet point clouds—which exhibit high noise levels (e.g., environmental interference) and irregular geometric incompleteness (e.g., occlusions)—compromising pseudolabel quality during self-training. Consequently, integrating low-confidence pseudolabeled real-world parapet data with labeled training data may reduce model robustness for this class. These results indicated that ST-CACM was able to establish a meaningful intermediate domain between hybrid training and real-world test data sets, mitigate the domain gaps between different data sets, and ultimately enhance the performance of semantic segmentation for most components.

Table 2 presents the effect of different proportions of VLS-generated synthetic data on the proposed ST-CACM method. Evidently, when the ST-CACM method was applied solely on VLS-based synthetic data, it achieved the lowest performance across all overall evaluation metrics and the IoU values for all bridge components. This result indicates the inadequacy of relying exclusively on synthetic data for model training. As the proportion of synthetic data was reduced from 100% to 20% within the training data set, a consistent upward trend was observed in all evaluation metrics. Such a pattern implies that the combination of an appropriate quantity of real-world data can enhance the performance of the ST-CACM method.

Furthermore, when real-world data formed the majority in the hybrid training data set (i.e., synthetic data accounts for 20%–40%), increasing the proportion of synthetic data resulted in only a marginal decline in performance, with the mIoU dropping by less than 1%. Conversely, when synthetic data became the dominant part of the hybrid training data set (i.e., synthetic data proportion is 60% or higher), a notable and significant decrease in performance was observed. This degradation was particularly pronounced when the proportion of synthetic data increased from 80% to 100%, highlighting the ST-CACM method's sensitivity to the balance of synthetic and real-world data used during training.

In light of this, and considering a practical scenario where real-world training data are available but limited in quantity, this study adopted a hybrid training configuration comprising 80% synthetic data. Purely synthetic training (i.e., 100%) was avoided because it significantly degrades performance, primarily due to the poor quality of pseudolabels generated during self-training. The selected 80% synthetic configuration effectively minimizes dependence on precious and scarce real-world data while retaining sufficient real input to preserve model reliability.

Table 3 describes the ablation study of the proposed ST-CACM framework. Starting from a baseline model trained without any ST-CACM components, we observed consistent improvements across all overall evaluation metrics with the progressive integration of ST, CM, and CA. The addition of ST alone improved the metric of mIoU to 88.47%, and incorporating CM further raised the mIoU to 89.33%. The full ST-CACM pipeline achieved the highest mIoU of 89.83%, representing a total improvement of 2.47% over the baseline. These results demonstrate that each component in ST-CACM contributes complementary benefits to the overall segmentation performance. Among the components, ST showed the largest individual gain, with a 1.03% improvement in mIoU, highlighting its effectiveness in closing the domain gap between synthetic and real-world data.

Additionally, Table 3 reveals varying degrees of improvement across different bridge components. For classes that are initially more challenging—such as decks—the improvements were more substantial, indicating the ST-CACM method's ability to enhance performance on difficult segmentation tasks. For classes that were already well-segmented—such as piers—the method still provided marginal gains, underscoring its robustness and general applicability. It is also worth noting that the performance on parapets was significantly affected by ST. This is due to the high noise levels in real-world parapet point clouds (e.g., environmental interference) and irregular geometric incompleteness (e.g., occlusions), which compromised pseudolabel quality during self-training.

Table 2. Effect of varying proportions of VLS-based synthetic data on the proposed ST-CACM Method

Methods	OA (%)	mAcc (%)	mIoU (%)	IoU for each class (%)			
				Piers	Girders	Decks	Parapets
100% synthetic data	94.21	93.28	86.97	98.12	89.86	78.33	81.55
80% synthetic data	95.34	94.31	89.83	98.99	92.00	83.97	84.35
60% synthetic data	95.63	94.76	90.95	99.12	92.73	84.32	86.41
40% synthetic data	95.88	95.83	91.19	99.26	93.12	84.50	87.85
20% synthetic data	96.00	96.75	91.45	99.36	93.48	84.66	88.28

Table 3. Ablation study of the proposed ST-CACM

Methods	OA (%)	mAcc (%)	mIoU (%)	IoU for each class (%)			
				Piers	Girders	Decks	Parapets
VLS-only model	93.01	93.39	87.36	98.91	87.98	77.17	85.36
VLS + ST model	94.13	93.80	88.39	98.94	89.56	81.24	83.82
VLS + ST-CM model	94.88	94.12	89.33	98.97	91.32	82.78	84.26
VLS + ST-CACM model	95.34	94.31	89.83	98.99	92	83.97	84.35

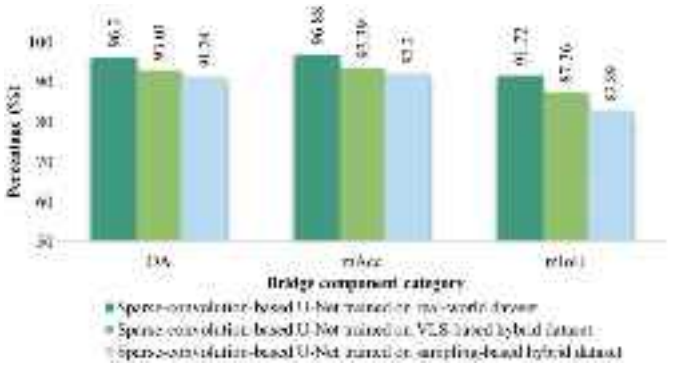

Fig. 10. Overall evaluation results for the three models trained with real-world-only data set, VLS-based hybrid data set, and traditional sampling-based hybrid data set.

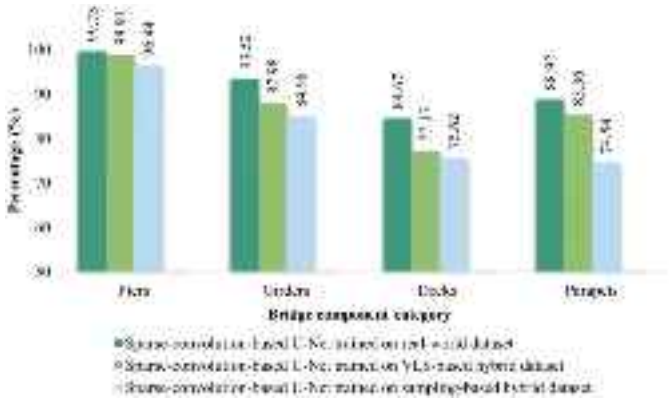

Fig. 11. Component-level evaluation results for three models trained with real-world-only data set, VLS-based hybrid data set, and traditional sampling-based hybrid data set.

However, the inclusion of CM and CA led to modest improvements in parapet segmentation, suggesting that these modules help mitigate the limitations of ST in such challenging cases.

Experiment 3: Characterizing the Effects of VLS on Virtual Point Cloud Synthetization

To answer the third question, we characterized the effects of VLS on the quality of synthetic point clouds and the performance of semantic segmentation. For this purpose, we trained and compared three different models trained with (1) a VLS-based hybrid data set (VLS-generated synthetic + real-world), (2) a traditional sampling-based hybrid data set (sampling-generated synthetic + real-world), and (3) a real-world-only data set. The total quantity of training point cloud data remained the same in all three cases.

Fig. 10 present the three models' performance in terms of OA, mAcc, and mIoU. It is evident that the VLS-based hybrid data set outperformed the traditional sampling-based hybrid data set across all overall evaluation metrics, with an increase of 1.77% in OA, 1.19% in mAcc, and 4.47% in mIoU. This improvement can be an indication that VLS-generated synthetic point clouds have visual and geometric features more closely aligned with the real-world data, reducing the domain gaps and enhancing the segmentation performance. Nonetheless, the model trained on the VLS-based hybrid data set fell behind the one trained on the real-world-only data set by 3.19% in OA, 3.49% in mAcc, and 4.36% in mIoU. This comparison result implies that although our VLS tool was able to create more realistic synthetic point clouds than the existing sampling-based synthetization, the domain gap with real-world point clouds still exists to some extent.

The component-level evaluation results are also presented in Fig. 11. As can easily be seen in the figure, the model trained on the VLS-based hybrid data set was not able to perform better than the model trained with the real-world data set in all categories, as similarly observed in Experiment 1. However, it is promising that the VLS-based hybrid data set can generate a more powerful model than the traditional sampling-based hybrid data set; it increased the IoU metric by 2.47% for piers, 3% for girders, 1.55% for decks, and 10.82% for parapets. Particularly, the performance was significantly improved for a specific type of bridge component, parapets (10.82% increase). This observation can be explained by the differences in geometric features between VLS-based and sampling-based synthetic point clouds, against real-world point clouds.

To better demonstrate the discrepancies among real-world data, VLS-based synthetic data, and sampling-generated data, Fig. 12 provides a visual comparison of these data sets. As displayed in Fig. 12, the real-world point clouds of parapets appear incomplete and noisy, likely due to occlusions, scanning errors, and environmental factors during data collection. These incompleteness and noises were reflected in VLS simulation because it mimics the real scanning mechanics, whereas such realism could not be imitated in sampling-based synthetization. Putting all together, it is thought that VLS can be a valid technique to generate more realistic, synthetic bridge point clouds, improving the performance of semantic segmentation.

Discussion

This study presented a synthetic-to-real domain adaptation method to close the visual gap between synthetic and real-world bridge point clouds. Our experimental results demonstrated the effectiveness of the proposed method. Specifically, our method was able to achieve a bridge semantic segmentation accuracy of 89.83% mIoU, which is comparable to the performance of the model trained only with a real-world data set (91.72%). Given that this comparability was obtained with much less quantity of real-world bridge point clouds (five versus one), this technical capability would be more significant. In addition, it was observed that ST-CACM—a

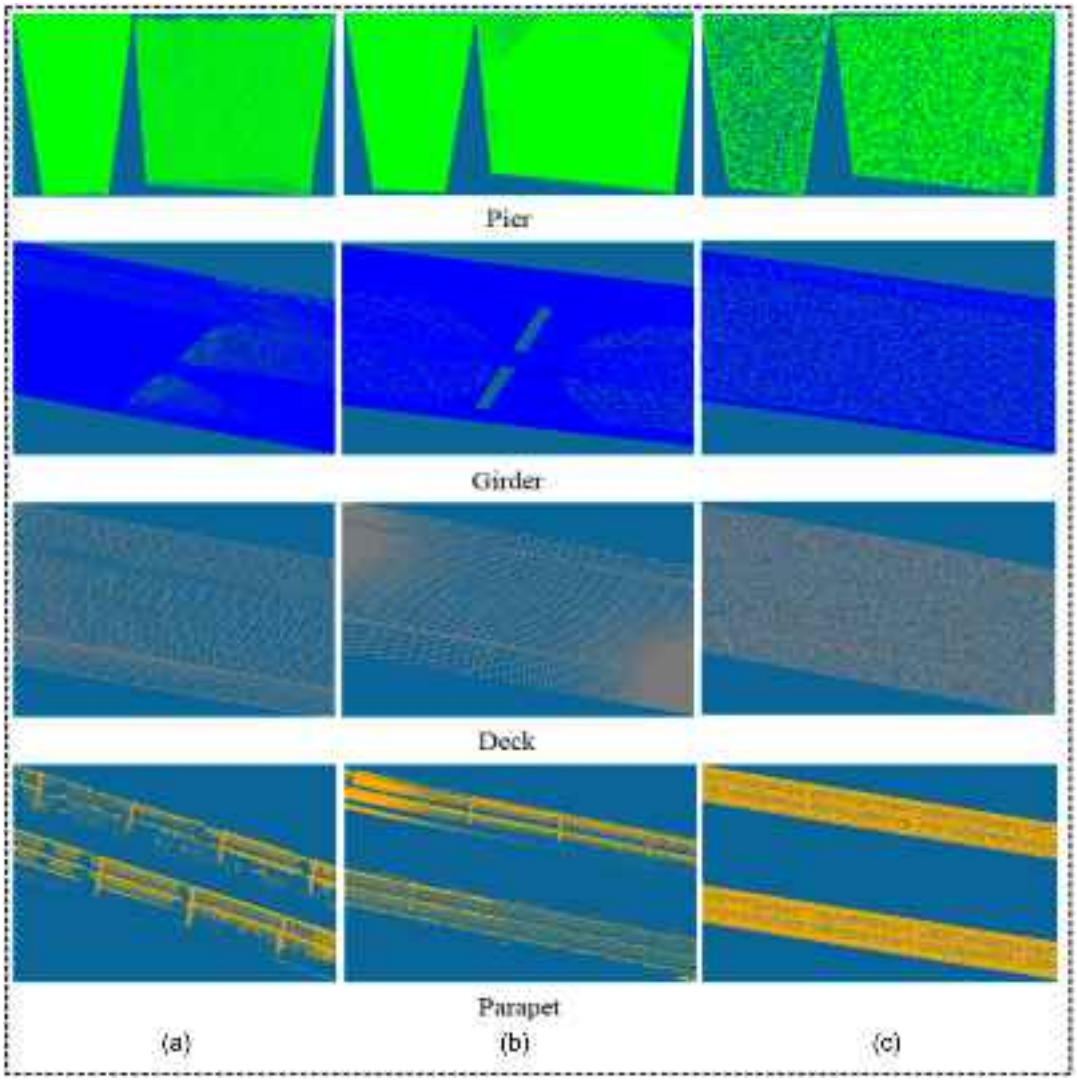

Fig. 12. Visualization examples of different types of bridge components: (a) real-world data; (b) VLS-based synthetic data; and (c) sampling-based synthetic data.

key module of our domain adaptation method—was proven to be beneficial to the segmentation performance through comparison with the VLS-based hybrid data set–trained model (89.83% versus 87.36%). The other key module, VLS, outperformed a traditional sampling-based point cloud synthetization approach in terms of mIoU (87.36% versus 82.89%).

These results show that the proposed method can alleviate the domain gap between real-world and synthetic bridge point clouds and eventually train a more scalable and comparable segmentation model to a real-world data set-trained model. This would contribute to addressing the scarcity of real-world point clouds and facilitating the wide application of synthetic data for semantic segmentation tasks in construction.

We performed a detailed analysis of the component-level performance differences between the model trained on the real-world data set and the one trained on the VLS-based hybrid data set. Looking at piers first, the two models exhibited a comparable performance, with less than a 1% difference in the IoU metric, which is far lower than other bridge components (Fig. 11). This result may be explained by the fact that piers are relatively easy accurately simulate in a synthetic environment because they typically have simpler and more uniform geometric shapes compared with other bridge components like parapets (Fig. 12). Another reason can be that piers are usually vertical structures with fewer obstructions around them, allowing for more optimal and consistent scanning angles. This nature can lead to higher-quality real-world point clouds that can be closely replicated in our virtual environment, contributing to the high degree of similarity between synthetic and real-world pier point clouds.

Unlike piers, we observed a significant performance difference in girders between the real-world data set and the VLS-hybrid data set (5.54% IoU). It may be attributed to the fact that girders are typically located directly above active roadways, posing challenges for close-range scanning and restricting optimal scanning angles. These can result in incomplete data capture and increased noise, especially for the connection points of the girders. Also, the presence of moving vehicles beneath the girders during scanning can significantly affect data quality due to occlusions that temporarily block the scanner's line of sight. Heavy traffic, especially from trucks or buses, can be another influencing factor that induces geometric deformation in the bridge structure. Given the factors, it can be thought that VLS has a great potential to create realistic and effective-to-learn synthetic point clouds for some types of bridge components (e.g., piers), but the domain gap with real-world point

clouds still exists for some other components (e.g., girders). However, the domain gap in girders can be further alleviated by ST-CACM (92.00% versus 87.98%).

Another type of bridge component, decks, also showed a significant performance difference of 7.50% IoU between the real-world data set and VLS-based hybrid data set. This disparity may owe to the influence of moving vehicles, the presence of physical defects (e.g., potholes) or vegetation (e.g., grass and moss, on real bridge decks, and geometric deformation caused by heavy traffic. These environmental factors could be absent in the idealized BrIM models that were used to create synthetic point clouds. However, similar to piers and girders, ST-CACM also exhibited a good effect in bridging the domain gap between VLS-based synthetic and real-world decks' point clouds (83.97% versus 77.17%).

Lastly, parapets also exhibited a slightly considerable difference of 3.56% IoU between the two models trained with a real-world data set and with a VLS-based hybrid data set. It may be because parapets are often located at the edges of bridges, elevated above the main structure, making them susceptible to partial obstruction by other structures. Moreover, the scanning angle for parapets is often less optimal compared with the horizontal surfaces of the bridge. These nonoptimal angles not only result in incomplete point cloud capture but also reduce the accuracy of point cloud data, introducing more noise. Another reason behind this observation can be that parapets usually have irregular shapes (not a uniform square shape like girders), causing scattering and refraction of the laser used in scanning, which further leads to inaccurate data points and increases noise in the collected point clouds. For this reason, synthetic point clouds generated from BrIMs can have geometric discrepancies compared with real-world data.

These discrepancies particularly affected the performance of parapet segmentation, but another domain adaptation technique, i.e., ST-CACM, was able to mitigate such performance decline by 1.01%. Putting all together, it can be concluded that VLS simulation has a promising potential to better replicate geometric characteristics of real-world point clouds than an existing sampling-based method. However, it is also revealed that integrating VLS and ST-CACM can further close the domain gap between synthetic and real-world data and improve the segmentation performance for all types of bridge components.

Training a bridge semantic segmentation model on a hybrid data set, comprising a small portion of real-world data and a larger portion of synthetic data, was found to be effective. As can be seen in Fig. 10, the model trained on the VLS-based hybrid data set achieved a satisfactory performance with OA of 93.01%, mAcc of 93.39%, and mIoU of 87.36%. It also secured an admissible performance of the IoU metric for each component, with 98.91% for piers, 87.98% for girders, 77.17% for decks, and 85.36% for parapets. Therefore, it is suggested for bridge professionals to create a small real-world bridge point cloud data set and augment it with a larger synthetic data set, rather than investing additional resources and time into collecting and labeling a large amount of real-world bridge point clouds. This hybrid approach can leverage the benefits of both data sources while mitigating adverse threats.

Particularly, real-world point clouds are realistic and representative but require numerous resources for data collection and labeling. In contrast, synthetic point clouds can be automatically generated and labeled, but inevitably exhibit domain gaps in capturing fine-grained real-world characteristics. Although synthetic point clouds have shown promising potential in training data-hungry models, they still cannot entirely replace real-world data (Xiao et al. 2024). Our study assumes a practical scenario in which some real-world training data are available, but their quantity is limited—an increasingly realistic condition in infrastructure inspection tasks. Based on this assumption, our experiments replaced 80% of the original real-world data with synthetic data (retaining only one of five real-world bridges). This setting represents an extreme case, where the proportion of synthetic point clouds is maximized (four out of five; 80%). This 80% synthetic composition demonstrates our method's robustness under high reliance on synthetic data. However, increasing the synthetic data proportion further (e.g., to 100%) may risk degrading pseudolabel quality during self-training, undermining the effectiveness of the domain adaptation framework. Thus, retaining minimal real-world data is essential for preserving the model reliability.

From a practical and commercial perspective, synthetic point clouds would be more attractive and profitable than real-world data. Collecting real-world point cloud data often involves expensive equipment, skilled labor, and time-consuming fieldwork. However, our VLS simulation tool can effortlessly generate diverse bridge point clouds closely resembling real-world scenarios, containing various geometric characteristics (e.g., occlusions, noises, and environmental conditions). Additionally, it automates the annotation of synthetic bridge point clouds without the need for manual and subjective labeling, which has also been an underlying limitation of existing techniques for synthetic point cloud generation (Ma et al. 2020; Yang et al. 2022b). The accurate and consistent annotation in synthetic data enables deep learning networks to learn precise geometric features of each bridge component, avoiding the issues caused by incorrect human annotations, especially at component boundaries. The clear component boundary segmentation is crucial for the reconstruction accuracy of digital twinning for existing bridges. Even small segmentation errors at the boundaries can have significant consequences during the digital twinning process. In addition, precise boundary segmentation can enhance the segmentation performance for minority component categories with fewer points like parapets, thereby improving the overall semantic segmentation performance.

Despite its success, the proposed domain adaptation method is not intended to be a panacea. Some challenges still remain:

- VLS-based synthetic data generation relies on the prebuilt BrIMs. However, most bridges do not have as-built BrIMs, making it challenging to establish a large synthetic data set that contains various bridge types.
- Although VLS-based synthetic bridge point clouds closely resemble reality, they lack realistic color information, which limits the deep learning–based bridge semantic segmentation model achieving the maximum performance.
- The generation process of VLS-based synthetic data involves complex and time-consuming data format conversions across different software.
- Although VLS-based synthetic data more closely mirror real-world scans and outperform sampling-based synthetic point clouds, no formal statistical significance test has yet been applied to compare the distributions of real-world, VLS-based synthetic, and sampling-based synthetic data sets.
- Despite the demonstrated success of the proposed domain adaptation method, its performance on a purely synthetic data set is suboptimal, primarily due to the lower quality of pseudolabels generated during self-training.

Consequently, synthetic data cannot fully substitute real-world bridge point clouds. To address these limitations, future work should focus on the following areas:

- Investigating the cutting-edge generative AI technologies to automatically generate various types of bridges with only text input.
- Generating realistic bridge color information from BrIMs and mapping it to VLS-based synthetic bridge point clouds.

- Developing a plugin for HELIOS++ that enables the direct generation of well-annotated point clouds from BrIMs.
- Applying analysis of variance to statistically assess and validate the effectiveness of VLS-based synthetic data in comparison with real-world and sampling-based synthetic data sets.
- Exploring the integration of self-training with adversarial training to enhance domain adaptation by compelling the model to learn domain-invariant features, aligning the source (e.g., synthetic data) and target domain (e.g., real-world data) within a shared feature space to minimize domain discrepancies and improve generalization to real-world scenarios.

In addition, in future work, our proposed framework will be designed with a modular architecture, enabling seamless integration with other state-of-the-art models through standardized data preprocessing and domain alignment steps, thereby further validating its versatility.

Conclusion

The bridge semantic segmentation task is a crucial intermediate step for reconstructing BrIMs of existing bridges. However, the scarcity of real-world bridge point clouds has been a significant obstacle in applying deep learning–based bridge semantic segmentation methods. Well-annotated synthetic data can be a promising solution, but there exists a substantial domain gap between synthetic and real-world point clouds. In this sense, this study presented a domain adaptation method composed of VLS simulation and ST-CACM. To validate its effectiveness, extensive comparison experiments were performed. Overall, the proposed method achieved 95.34% for OA, 94.31% for mAcc, and 89.83% for mIoU, exhibiting a comparable level of semantic segmentation accuracy to the model trained on a large size of real-world data set.

The efficacy of VLS and ST-CACM was also proven, respectively. Specifically, the model performance was improved in integration of VLS (IoU of 4.47% compared with the model trained on traditional sampling-based hybrid data set) and ST-CACM (IoU of 2.47% compared with the model trained on only VLS-based hybrid data set). These results demonstrate that the proposed method can bridge the domain gap between synthetic and real-world bridge point clouds and generate a more promising segmentation model. This finding is more encouraging granted that the proposed method was able to reduce the quantity of real-world bridge point cloud data by 80%. This added-on knowledge can help to alleviate the data scarcity challenge of real-world bridge point clouds for semantic segmentation tasks in construction. We also expect that this advancement will eventually contribute to digital twinning of existing infrastructures and facilitating smart, efficient, and cost-saving operations and maintenance.

Appendix. Summary of Point Distribution across Categories in the Real-World Bridge Point Cloud Data Set

Bridge number	Total number of points	Number of points per component			
		Pier	Girder	Deck	Parapet
Bridge 1	3,474,764	731,687	1,740,870	697,020	305,187
Bridge 2	4,438,498	771,740	1,956,234	1,200,568	509,956
Bridge 3	33,400,434	2,470,372	27,941,101	1,926,654	1,062,307
Bridge 4	1,562,540	438,033	968,470	124,568	31,469
Bridge 5	32,944,903	3,947,322	21,441,886	6,280,225	1,275,470
Bridge 6	3,587,275	604,872	1,641,460	1,057,432	283,511
Bridge 7	30,851,968	4,050,184	20,789,567	4,938,701	1,073,516

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-23963349). The authors would also like to acknowledge the support by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No. AISG2-TC-2021-001), and the Ministry of Education Tier 1 Grants, Singapore (Nos. RS04/21 and RG146/23).

Author Contributions

Xiaofei Yang: Conceptualization; Formal analysis; Investigation; Methodology; Validation; Visualization; Writing – original draft. Yuguang Fu: Resources; Supervision; Writing – review and editing. Jinwoo Kim: Conceptualization; Funding acquisition; Investigation; Methodology; Supervision; Writing – review and editing.

References

Autodesk. 2024 "Autodesk Revit: BIM software to design and make anything." Accessed October 14, 2025. https://www.autodesk.com.sg/products/revit/overview.

Blender Foundation. 2022. "Blender 3.2: Open source 3D creation suite." Accessed October 14, 2025. https://www.blender.org/.

Bousmalis, K., N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. 2017. "Unsupervised pixel-level domain adaptation with generative adversarial networks." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 3722–3731. New York: IEEE. https://doi.org/10.48550/arXiv.1612.05424.

Choy, C., J. Gwak, and S. Savarese. 2019. "4D spatio-temporal convnets: Minkowski convolutional neural networks." In *Proc., IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 3075–3084. New York: IEEE. https://doi.org/10.48550/arXiv.1904.08755.

Ding, R., J. Yang, L. Jiang, and X. Qi. 2022. "DODA: Data-oriented sim-to-real domain adaptation for 3D semantic segmentation." In *Proc., European Conf. on Computer Vision*, 284–303. New York: Springer. https://doi.org/10.1007/978-3-031-19812-0_17.

Gschwandtner, M., R. Kwitt, A. Uhl, and W. Pree. 2011. "BlenSor: Blender sensor simulation toolbox." In *Proc., Advances in Visual Computing: 7th Int. Symp., ISVC 2011*, 199–208. New York: Springer. https://doi.org/10.1007/978-3-642-24031-7_20.

Hämmerle, M., N. Lukač, K.-C. Chen, Z. Koma, C.-K. Wang, K. Anders, and B. Höfle. 2017. "Simulating various terrestrial and UAV LiDAR scanning configurations for understory forest structure modelling." *ISPRS Ann. Photogramm. Rem. Sens. Spat. Inf. Sci.* IV-2/W4 (Sep): 59–65. https://doi.org/10.5194/isprs-annals-IV-2-W4-59-2017.

Han, B., J. W. Ma, and F. Leite. 2021. "A framework for semi-automatically identifying fully occluded objects in 3D models: Towards comprehensive construction design review in virtual reality." *Adv. Eng. Inf.* 50 (Oct): 101398. https://doi.org/10.1016/j.aei.2021.101398.

Hu, D., V. J. Gan, and R. Zhai. 2024. "Automated BIM-to-scan point cloud semantic segmentation using a domain adaptation network with hybrid attention and whitening (DawNet)." *Autom. Constr.* 164 (Aug): 105473. https://doi.org/10.1016/j.autcon.2024.105473.

Hu, Q., D. Liu, and W. Hu. 2023. "Density-insensitive unsupervised domain adaption on 3d object detection." In *Proc., IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 17556–17566. New York: IEEE. https://doi.org/10.48550/arXiv.2304.09446.

Jiang, F., L. Ma, T. Broyd, and K. Chen. 2021. "Digital twin and its implementations in the civil engineering sector." *Autom. Constr.* 130 (Oct): 103838. https://doi.org/10.1016/j.autcon.2021.103838.

Jiang, L., H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia. 2020. "Pointgroup: Dual-set point grouping for 3D instance segmentation." In *Proc., IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 4867–4876. New York: IEEE. https://doi.org/10.48550/arXiv.2004.01658.

Jing, Y., B. Sheil, and S. Acikgoz. 2022. "Segmentation of large-scale masonry arch bridge point clouds with a synthetic simulator and the BridgeNet neural network." *Autom. Constr.* 142 (Oct): 104459. https://doi.org/10.1016/j.autcon.2022.104459.

Kim, H., and C. Kim. 2020. "Deep-learning-based classification of point clouds for bridge inspection." *Rem. Sens.* 12 (22): 3757. https://doi.org/10.3390/rs12223757.

Kim, H., J. Yoon, and S. H. Sim. 2020a. "Automated bridge component recognition from point clouds using deep learning." *Struct. Control Health Monit.* 27 (9): e2591. https://doi.org/10.1002/stc.2591.

Kim, J., and S. Chi. 2021. "A few-shot learning approach for database-free vision-based monitoring on construction sites." *Autom. Constr.* 124 (Apr): 103566. https://doi.org/10.1016/j.autcon.2021.103566.

Kim, J., J. Hwang, S. Chi, and J. Seo. 2020b. "Towards database-free vision-based monitoring on construction sites: A deep active learning approach." *Autom. Constr.* 120 (Dec): 103376. https://doi.org/10.1016/j.autcon.2020.103376.

Kim, J., D. Kim, S. Lee, and S. Chi. 2023. "Hybrid DNN training using both synthetic and real construction images to overcome training data shortage." *Autom. Constr.* 149 (May): 104771. https://doi.org/10.1016/j.autcon.2023.104771.

Kong, J. S., and D. M. Frangopol. 2003. "Life-cycle reliability-based maintenance cost optimization of deteriorating structures with emphasis on bridges." *J. Struct. Eng.* 129 (6): 818–828. https://doi.org/10.1061/(ASCE)0733-9445(2003)129:6(818).

Korus, K., T. Czerniawski, and M. Salamak. 2023. "Visual programming simulator for producing realistic labeled point clouds from digital infrastructure models." *Autom. Constr.* 156 (Dec): 105126. https://doi.org/10.1016/j.autcon.2023.105126.

Kukko, A., and J. Hyyppä. 2009. "Small-footprint laser scanning simulator for system validation, error assessment, and algorithm development." *Photogramm. Eng. Remote Sens.* 75 (10): 1177–1189. https://doi.org/10.14358/PERS.75.10.1177.

Larose, D. T., and C. D. Larose. 2014. "k-nearest neighbor algorithm." In *Discovering knowledge in data: An introduction to data mining*, 149–164. Hoboken, NJ: Wiley. https://doi.org/10.1002/9781118874059.ch7.

Lee, J. S., J. Park, and Y.-M. Ryu. 2021. "Semantic segmentation of bridge components based on hierarchical point cloud model." *Autom. Constr.* 130 (Oct): 103847. https://doi.org/10.1016/j.autcon.2021.103847.

Lu, R., and I. Brilakis. 2019. "Digital twinning of existing reinforced concrete bridges from labelled point clusters." *Autom. Constr.* 105 (Sep): 102837. https://doi.org/10.1016/j.autcon.2019.102837.

Lu, R., I. Brilakis, and C. R. Middleton. 2019. "Detection of structural components in point clouds of existing RC bridges." *Comput.-Aided Civ. Infrastruct. Eng.* 34 (3): 191–212. https://doi.org/10.1111/mice.12407.

Ma, J. W., T. Czerniawski, and F. Leite. 2020. "Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds." *Autom. Constr.* 113 (May): 103144. https://doi.org/10.1016/j.autcon.2020.103144.

Ma, J. W., B. Han, and F. Leite. 2021. "An automated framework for generating synthetic point clouds from as-built BIM with semantic annotation for scan-to-BIM." In *Proc., 2021 Winter Simulation Conf. (WSC)*, 1–10. New York: IEEE. https://doi.org/10.1109/WSC52266.2021.9715301.

Mafipour, M. S., C. Alici, S. S. Shakeel, and A. Kalkavan. 2022. "Semantic segmentation of real and synthetic point cloud data for digital twinning of bridges." In *Proc., 33, Forum Bauinformatik*. Berlin: Springer.

Mehranfar, M., M. A. Vega-Torres, A. Braun, and A. Borrmann. 2024. "Automated data-driven method for creating digital building models from dense point clouds and images through semantic segmentation and parametric model fitting." *Adv. Eng. Inf.* 62 (Oct): 102643. https://doi.org/10.1016/j.aei.2024.102643.

Moselhi, O., M. Ahmed, and A. Bhowmick. 2017. "Multisensor data fusion for bridge condition assessment." *J. Perform. Constr. Facil.* 31 (4): 04017008. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001000.

Neumann, M., D. Borrmann, and A. Nüchter. 2022. "Semantic classification in uncolored 3D point clouds using multiscale features." In *Proc., Int. Conf. on Intelligent Autonomous Systems*, 342–359. New York: Springer. https://doi.org/10.1007/978-3-031-22216-0_24.

Pan, Y., M. Wang, L. Lu, R. Wei, S. Cavazzi, M. Peck, and I. Brilakis. 2024. "Scan-to-graph: automatic generation and representation of highway geometric digital twins from point cloud data." *Autom. Constr.* 166 (Oct): 105654. https://doi.org/10.1016/j.autcon.2024.105654.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga. 2019. "Pytorch: An imperative style, high-performance deep learning library." In *Proc., 32, Advances in Neural Information Processing Systems*. Red Hook, NY: Curran Associates.

Qi, C. R., H. Su, K. Mo, and L. J. Guibas. 2017. "Pointnet: Deep learning on point sets for 3D classification and segmentation." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 652–660. New York: IEEE. https://doi.org/10.1109/CVPR.2017.16.

Rist, C. B., M. Enzweiler, and D. M. Gavrila. 2019. "Cross-sensor deep domain adaptation for lidar detection and segmentation." In *Proc., 019 IEEE Intelligent Vehicles Symp. (IV)*, 1535–1542. New York: IEEE. https://doi.org/10.1109/IVS.2019.8814047.

Rosenberg, C., M. Hebert, and H. Schneiderman. 2005. "Semi-supervised self-training of object detection models." *Proc., 7th IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)*, 29–31. New York: IEEE. https://doi.org/10.1109/ACVMOT.2005.107.

Saito, K., Y. Ushiku, and T. Harada. 2017."Asymmetric tri-training for unsupervised domain adaptation." In *Proc., Int. Conf. on Machine Learning*, 2988–2997. https://doi.org/10.48550/arXiv.1702.08400.

Saltori, C., F. Galasso, G. Fiameni, N. Sebe, E. Ricci, and F. Poiesi. 2022. "Cosmix: Compositional semantic mix for domain adaptation in 3D lidar segmentation." In *Proc., European Conf. on Computer Vision*, 586–602. New York: Springer. https://doi.org/10.48550/arXiv.2207.09778.

Shen, S., Y. Xia, A. Eich, Y. Xu, B. Yang, and U. Stilla. 2023. "SegTrans: Semantic segmentation with transfer learning for MLS point clouds." *IEEE Geosci. Remote Sens. Lett.* 20 (Jul): 1–5. https://doi.org/10.1109/LGRS.2023.3294748.

Wang, Z., S. Ding, Y. Li, M. Zhao, S. Roychowdhury, A. Wallin, G. Sapiro, and Q. Qiu. 2019. "Range adaptation for 3D object detection in lidar." In *Proc., IEEE/CVF Int. Conf. on Computer Vision Workshops*. New York: Springer. https://doi.org/10.1109/ICCVW.2019.00285.

Winiwarter, L., A. M. E. Pena, H. Weiser, K. Anders, J. M. Sánchez, M. Searle, and B. Höfle. 2022. "Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3D laser scanning." *Remote Sens. Environ.* 269 (Feb): 112772. https://doi.org/10.1016/j.rse.2021.112772.

Xia, T., J. Yang, and L. Chen. 2022. "Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning." *Autom. Constr.* 133 (Jan): 103992. https://doi.org/10.1016/j.autcon.2021.103992.

Xiao, A., J. Huang, D. Guan, K. Cui, S. Lu, and L. Shao. 2022. "Polarmix: A general data augmentation technique for lidar point clouds." *Adv. Neural Inf. Process. Syst.* 35 (Dec): 11035–11048. https://doi.org/10.48550/arXiv.2208.00223.

Xiao, A., J. Huang, K. Liu, D. Guan, X. Zhang, and S. Lu. 2024. "Domain adaptive LiDAR point cloud segmentation via density-aware self-training." *IEEE Trans. Intell. Transp. Syst.* 25 (10): 13627–13639. https://doi.org/10.1109/TITS.2024.3386865.

Xu, J., W. Yang, L. Kong, Y. Liu, R. Zhang, Q. Zhou, and B. Fei. 2024. "Visual foundation models boost cross-modal unsupervised domain adaptation for 3d semantic segmentation." *IEEE Trans. Intell. Transp. Syst.*. https://doi.org/10.48550/arXiv.2403.10001.

Yang, J., S. Shi, Z. Wang, H. Li, and X. Qi. 2021. "ST3D: Self-training for unsupervised domain adaptation on 3D object detection." In *Proc., IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 10368–10378. New York: IEEE. https://doi.org/10.48550/arXiv.2103.05346.

Yang, J., S. Shi, Z. Wang, H. Li, and X. Qi. 2022a. "ST3D++: Denoised self-training for unsupervised domain adaptation on 3D object detection." *IEEE Trans. Pattern Anal. Mach. Intell.* 45 (5): 6354–6371. https://doi.org/10.1109/TPAMI.2022.3216606.

Yang, T., Y. Zou, X. Yang, and E. del Rey Castillo. 2024. "Domain knowledge-enhanced region growing framework for semantic segmentation of bridge point clouds." *Autom. Constr.* 165 (Sep): 105572. https://doi.org/10.1016/j.autcon.2024.105572.

Yang, X., E. del Rey Castillo, Y. Zou, and L. Wotherspoon. 2023. "Semantic segmentation of bridge point clouds with a synthetic data augmentation strategy and graph-structured deep metric learning." *Autom. Constr.* 150 (Jun): 104838. https://doi.org/10.1016/j.autcon.2023.104838.

Yang, X., E. del Rey Castillo, Y. Zou, L. Wotherspoon, and Y. Tan. 2022b. "Automated semantic segmentation of bridge components from large-scale point clouds using a weighted superpoint graph." *Autom. Constr.* 142 (Oct): 104519. https://doi.org/10.1016/j.autcon.2022.104519.

Zhai, R., J. Zou, Y. He, and L. Meng. 2022. "BIM-driven data augmentation method for semantic segmentation in superpoint-based deep learning network." *Autom. Constr.* 140 (Aug): 104373. https://doi.org/10.1016/j.autcon.2022.104373.

Zhang, H. X., and Z. Zou. 2023. "Quality assurance for building components through point cloud segmentation leveraging synthetic data." *Autom. Constr.* 155 (Nov): 105045. https://doi.org/10.1016/j.autcon.2023.105045.

Zhang, S., X. Li, M. Zong, X. Zhu, and R. Wang. 2017. "Efficient kNN classification with different numbers of nearest neighbors." *IEEE Trans. Neural Networks Learn. Syst.* 29 (5): 1774–1785. https://doi.org/10.1109/TNNLS.2017.2673241.

Zhang, Y., C. Zhou, and D. Huang. 2024. "STAL3D: Unsupervised domain adaptation for 3D object detection via collaborating self-training and adversarial learning." *IEEE Trans. Intell. Veh.* 9 (11): 7339–7350. https://doi.org/10.1109/TIV.2024.3397194.

Zou, L., H. Tang, K. Chen, and K. Jia. 2021. "Geometry-aware self-training for unsupervised domain adaptation on object point clouds." In *Proc., IEEE/CVF Int. Conf. on Computer Vision*, 6403–6412. New York: IEEE. https://doi.org/10.48550/arXiv.2108.09169.

Zou, Y., Z. Yu, B. Kumar, and J. Wang. 2018. "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training." In *Proc., European Conf. on Computer Vision (ECCV)*, 289–305. Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-030-01219-9_18.