www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

LIMITED-DATA TRANSFER LEARNING FOR SEMANTIC SEGMENTATION AND PRE-LABELING OF 3D SHELL CONSTRUCTION LIDAR SCANS

SUBMITTED: July 2025
PUBLISHED: April 2026
EDITOR: Frédéric Bosché
DOI: 10.36680/j.itcon.2026.022

Lukas Rauch, M.Sc (corresponding author)
Institute of Structural Engineering, University of the Bundeswehr Munich, Germany
https://orcid.org/0000-0002-7501-7769
lukas.rauch@unibw.de

Thomas Braml, Univ.-Prof. Dr.-Ing.
Institute of Structural Engineering, University of the Bundeswehr Munich, Germany
https://orcid.org/0000-0002-0745-4588
thomas.braml@unibw.de

SUMMARY*: This study assesses transformer-based 3D semantic segmentation models for detecting structural components in terrestrial laser scans, given that no training data currently exists for shell construction sites. Manual annotation of 3D point clouds is expensive, yet high-quality labels remain essential for supervised computer vision and validation. Automated pre-labeling can cut down annotation effort by shifting human tasks from exhaustive labeling to targeted verification and correction, assuming models can robustly identify the most common structural elements. We designed a three-stage evaluation protocol covering (i) supervised learning, (ii) cross-domain generalization, and (iii) transfer learning with limited labeled data in the target domain to test model generalization in this context. Three transformer architectures (Point Transformer V2, Point Transformer V3, and Swin3D) are evaluated using four established indoor datasets (S3DIS, ScanNetV2, Structured3D, and VASAD) and a custom domain-specific dataset of annotated construction scenes. Training only on the limited construction dataset results in weak generalization. In contrast, pretraining on loosely related synthetic data and fine-tuning on a minimal number of labeled construction scenes enable reliable segmentation of core building components. A sensitivity analysis also showed that just 12 samples are sufficient to calibrate a pretrained model to a specific building type. The models perform well despite differences between synthetic training data and noisy real-world scans. Among the evaluated architectures, Swin3D delivers the best performance, with +18% mIoU improvement through general pretraining, while PTv3 converges faster with fewer target-domain samples. These findings suggest that transfer learning with limited labeled construction data offers a practical foundation for scalable pre-labeling workflows and human-in-the-loop applications in architecture, engineering, and construction.*

KEYWORDS*: shell construction, point cloud, semantic segmentation, structural components, transfer learning.*

REFERENCE*: Rauch, L., & Braml, T. (2026). Limited-data transfer learning for semantic segmentation and pre-labeling of 3D shell construction LiDAR scans. Journal of Information Technology in Construction (ITcon), 31, 477–506. https://doi.org/10.36680/j.itcon.2026.022*

1. INTRODUCTION

Advancements in spatial scene perception have enabled a range of computer vision applications in the construction industry, from automating documentation workflows to supporting autonomous robotics in dynamic site environments. However, recent progress in 3D perception, particularly in segmentation and object detection, has been driven predominantly by research domains such as autonomous driving and indoor navigation in residential or commercial settings. This focus has led to substantial methodological developments. Yet it has also limited the transfer of these advances to active construction sites, which present distinct spatial, material, and structural challenges. As a result, the Architecture, Engineering, and Construction (AEC) domain remains underrepresented in current research on large-scale 3D scene understanding.

Across all phases of a building's life cycle (construction, renovation, retrofitting, and demolition), spatial perception and the reconstruction of as-built Building Information Models (BIM) have been shown to improve coordination, prevent critical design and assembly errors, and reduce rework. (Jiang & Messner, 2023; Reja et al., 2022) The ability to automatically recognize and classify structural components enables progress monitoring, subcontractor tracking, and accurate construction documentation. Moreover, systematic 3D scene understanding can streamline planning processes, particularly in complex retrofit scenarios where precise as-built models are mandatory. Semantic scene understanding and object detection also play a key role in the safe operation of autonomous machines on construction sites, enabling obstacle avoidance and real-time human-machine interaction.

Despite the potential benefits, the development of deep learning methods tailored to construction-specific applications remains hindered by the lack of publicly available 3D datasets. While several benchmark datasets (Armeni et al., 2016; Dai et al., 2017; Zheng et al., 2020) provide semantic annotations for furnished indoor spaces, they are not particularly relevant to shell-construction scenarios. Most notably, they lack representation of the unfinished, cluttered nature of construction sites, and their class labels typically omit fundamental structural and technical building elements. Generating new, domain-specific training and validation data is thus essential. However, acquiring and annotating high-resolution point clouds remains time-consuming and resource-intensive. In this context, automated pre-labeling enabled by transfer learning, despite limited training data, offers a promising strategy to reduce the burden of manual annotation while improving the scalability of semantic segmentation pipelines in construction environments.

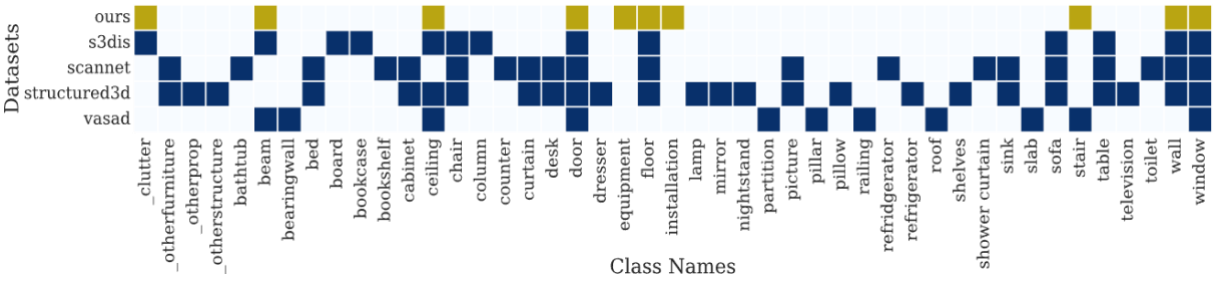

Figure 1: One-hot encoding of class occurrence and overlap between our proposed target validation dataset and four public indoor datasets: S3DIS, ScannetV2, Structured3D, and VASAD. The listed datasets are used to train models for the semantic segmentation of building component groups.

In this work, we address the challenge of semantic component segmentation in the AEC domain with limited available training data. To this end, we design a series of experiments that leverage both established benchmark datasets and a newly collected, domain-specific validation dataset representing shell construction environments. Our evaluation framework comprises three phases: (1) fully supervised training on our custom dataset, serving as a reference for model performance; (2) cross-domain training using publicly available indoor datasets to assess the generalization capability of transformer-based segmentation models; and (3) transfer learning, in which models pretrained on general indoor scenes are fine-tuned on our AEC-specific dataset to improve segmentation accuracy. The experimental design allows us to explore the integration of domain-specific and general-purpose data, compare the effectiveness of different transformer architectures under constrained data availability, and evaluate the impact of pretraining on non-construction-site data for the real-world model performance.

In practical applications, annotated training data are often collected incrementally, particularly in the construction

domain, where data acquisition and labeling are costly and time-consuming. To reflect this reality, a data sensitivity study examines how segmentation performance evolves as additional labeled samples become available for model fine-tuning. This analysis helps assess how efficiently transferred knowledge can be exploited under limited-data conditions and identify the point at which further data yield diminishing performance gains. By doing so, the study offers insights into the practical usefulness and robustness of transfer learning for real-world construction site applications.

Through this study, we aim to demonstrate how existing data from outside the domain can be strategically combined with targeted domain data to improve semantic segmentation for construction applications, thereby helping close the gap between generic computer vision research and the specialized demands of the AEC industry.

2. RELATED WORK

Semantic segmentation has been successfully used in computer vision to classify each pixel in an image into a specific category. This concept applies to classifying each point in a 3D point cloud, effectively partitioning the entire scene into meaningful segments. Unlike traditional classification tasks that assign a single label to the entire image or object, semantic segmentation provides a more detailed understanding by labeling every element in the scene. This becomes increasingly important as 3D sensors become more accessible and affordable, and as autonomous robot automation expands. Supervised deep learning requires large amounts of data, which are not always easy to obtain in sufficient quantities. Therefore, innovative approaches have been developed to make the most of existing datasets. The following paragraphs briefly highlight advances and milestones in deep learning-based point cloud semantic segmentation that lay the foundation for this work.

2.1 General point cloud semantic segmentation

Building on the success of computer vision in image data segmentation (He et al., 2017; Long et al., 2015; Ronneberger et al., 2015), the learning-based pointwise semantic segmentation of unstructured point clouds was made possible for the first time by the publication of PointNet in 2016 (Charles et al., 2017). A detailed summary of the early milestones in subsequent years is beyond the scope of this paper. Therefore, the authors recommend Guo et al.'s survey (Y. Guo et al., 2021) as a comprehensive review of the fundamentals and developments in point cloud segmentation up to the end of 2020. In recent years, transformer-based deep learning architectures have outperformed their competitors (Point-CNNs, Graph-CNNs, and RNNs) across most 3D perception tasks, as evidenced by the comparison of benchmark results (paperswithcode, 2024; Rauch & Braml, 2023; Wu et al., 2022; Yang et al., 2023). These point transformers were inspired by the success of transformers in natural language processing (NLP) (Devlin et al., 2018; Lahoud et al., 2022; Vaswani et al., 2017) and 2D image recognition (Carion et al., 2020; Dosovitskiy et al., 2020; Z. Liu et al., 2021). The core of the transformer model family is a self-attention mechanism that is permutation-equivariant and robust to the cardinality of the input elements. Applying self-attention to 3D point clouds is intuitive since point clouds are essentially sets embedded irregularly in a metric space (Zhao et al., 2021). The naive transformer computes global self-attention across the entire point cloud, enabling long-range attention between scattered point patches (M.-H. Guo et al., 2021). However, this approach leads to high memory and computational costs due to the quadratic complexity of self-attention (Wu et al., 2022). Zhao et al. (Zhao et al., 2021) introduced local attention around each data point (using k nearest neighbors), reducing complexity and making Point Transformers feasible at the scene level. Wu et al.'s Point Transformer V2 (Wu et al., 2022) employed grouped vector attention, allowing the creation of deeper networks, enhanced position encoding, and partition-based pooling to address irregular spatial point distribution. The model's generalization capabilities were further improved by widening the receptive field, as Wu et al. (Wu, Jiang, et al., 2023) achieved through prioritizing simplicity and efficiency. Around the same time, the Stratified Transformer (Lai et al., 2022) adopted a grid-based sliding window attention mechanism from 2D vision's Swin Transformer (Z. Liu et al., 2021), enabling transformer blocks to operate within a sequence of shifted windows on a 3D voxelated point cloud. Swin3D (Yang et al., 2023) enhanced the naive window attention for sparse 3D voxel grids by reimplementing multi-head self-attention. This reduced memory costs from quadratic to linear with respect to the number of sparse voxels per window, allowing for a wider receptive field and improved generalization.

Contrary to the trend toward ever-larger transformer models with more sophisticated local feature aggregation mechanisms, lately, there is a new focus on more efficient MLPs for hierarchical feature extraction. PointMLP (Ma et al., 2022) argues that detailed local information is not the key to point cloud analysis and proposes a

residual-block MLP framework that achieves significantly faster inference speeds with only minor accuracy losses. PointNeXt (Qian et al., 2022) revisits the influential PointNet++ (Qi et al., 2017) and claims that its full potential has not yet been realized. It also suggests that much of the success of modern models is attributable to improved training strategies. PointNeXt uses the same set abstraction and feature propagation blocks as PointNet++, but adds an extra MLP layer at the beginning and expands the architecture with inverted residual MLP (InvResMLP) blocks. The PointNeXt architecture can be scaled by adjusting the number of initial MLP-channels (width) and the number of InvResMLP blocks (depth) to achieve competitive performance. These simpler MLP-based frameworks achieve impressive improvements in training speed and inference latency, thanks to lower memory overhead. This factor should not be underestimated if practical applications limit inference time. However, despite scaling and improved training strategies, accuracy on large-scale benchmarks still falls short of that of modern transformers, which can scale receptive fields for long-range context.

2.2 Pretrained transformer backbones

The integration of large pretrained backbones has driven significant progress in NLP and 2D vision, enabling better task generalization, streamlined network design, and reduced requirements for labeled data and training time (Bao et al., 2021; Devlin et al., 2018; Z. Liu et al., 2021). This approach involves pretraining a general backbone network on broad datasets, which can then be fine-tuned for various downstream tasks, such as segmentation or object detection. PointContrast (Xie et al., 2020), Contrastive Scene Contexts (Hou et al., 2021), DepthContrast (Zhang et al., 2021), and Masked Scene Contrast (Wu, Wen, et al., 2023) utilize unsupervised pretraining on multi-view data from ScanNet (Dai et al., 2017) to learn general 3D representations for a Minkowski U-Net backbone. They demonstrate that pretrained network weights, which are universally applicable and useful for many high-level 3D understanding tasks, can significantly reduce the data required for fine-tuning downstream tasks but remain inferior to state-of-the-art supervised methods. PointContrast also assessed supervised pretraining of the U-Net but found only a minor benefit. Investigations by Yang et al. (Yang et al., 2023) indicated that the advantages of supervised pretraining for U-Net-style networks may be restricted, probably because the convolutional encoder structure is not able to extract data priors as effectively as transformer structures. As a result, they released Swin3D itself, a pretrained transformer backbone designed for comprehensive indoor 3D scene understanding. The supervised training was carried out on the synthetic dataset Structured3D (Zheng et al., 2020) and can be further fine-tuned for downstream tasks. Experiments on point cloud semantic segmentation demonstrate robust domain generalization capabilities across multiple real-world datasets, emphasizing the advantages of transfer learning for small datasets. Recently, pretrained image or CLIP models have been adopted for 3D learning (Dong et al., 2023; Huang et al., 2024), forming a new type of pretrained 3D backbones that utilize even larger general training resources across text and image modalities.

2.3 Point cloud segmentation for architecture, engineering, and construction

The development of deep learning-based segmentation methods for architecture, engineering, and construction (AEC) applications in the built environment and indoor scenes largely depends on a few datasets, such as S3DIS (Armeni et al., 2016a), SceneNN (Hua et al., 2016), ScannetV2 (Dai et al., 2017), and Structured3D (Zheng et al., 2020). The task of segmenting indoor spaces has become a well-established benchmark for demonstrating new models' ability to understand dense point cloud scenes with high-level detail. The positive side effect for the AEC disciplines is that the models simultaneously learn to recognize some basic building component groups (Choy et al., 2019; Schult et al., 2022; Wu et al., 2022). However, these datasets primarily feature interior design and furniture classes, as well as basic building elements such as ceilings, floors, walls, and beams, along with pre-installed door and window elements. This finding makes it apparent that decisive advances in understanding buildings and interiors through computer vision depend on the application requirements of architects focused on interior design. The application of such segmentation systems to engineering disciplines and structural component reconstruction may require more technical understanding and, thus, a broader range of represented building component classes.

Nevertheless, some approaches have used these indoor datasets for structural component segmentation to train machine learning and deep learning models, as well as for validation. Ma et al. (Ma et al., 2020) and Zhang et al. (Zhang & Zou, 2023) used S3DIS and ScannetV2 samples to augment synthetic training data from BIM models with additional real-world point clouds. The results showed that training exclusively on synthetic point clouds from BIM models yields inadequate results, as the domain gap between low-detail CAD geometries and the

variation-rich reality impedes model generalization. Tang et al. (Tang et al., 2022), Chen et al. (Chen et al., 2019), Kim et al. (Kim et al., 2020), Mehranfar et al. (Mehranfar et al., 2023), Park et al. (Park et al., 2022) developed and tested deep learning approaches using the S3DIS dataset to extract structural components, such as ceilings, floors, and walls; and to reconstruct BIM representations from the boundaries of these components. Liang et al. (Liang et al., 2024) used 3D-registered images from S3DIS to fuse established 2D convolutional image classification (AlexNet (Krizhevsky et al., 2012) and GoogleNet (Szegedy et al., 2015)) with material-augmented point cloud semantic segmentation.

In addition to work based on these general datasets, several publications demonstrated structural component segmentation using data with a stronger technical focus on the AEC sector. The VASAD dataset (Langlois et al., 2022) took a synthetic-data approach to segmentation-based scene reconstruction in the context of structural engineering. It consists of six digital computer-aided design (CAD) building models with full-volume descriptions and semantic labels. Synthetic training data can be collected by raytracing virtual laser beams through these CAD models, allowing the generation of theoretically infinite scans from virtually any camera position. For our work, we used the pre-rendered point clouds provided by the VASAD authors, which included a set of 10 ground-truth semantic labels, all of which were considered construction-related. The occurrence of the object classes and their common overlap among the four other datasets is shown as one-hot encoding in Figure 1.

Son et al. (Son & Kim, 2017) collected laser-scanning data from two construction sites where columns and beams are frequently encountered to develop a feature-based component classification algorithm for as-built reconstruction. Zepp et al. (Zepp, 2023) tested a hybrid approach in a single construction site environment to augment laser-scanning data for training tasks with annotations from 2D image segmentation and structure-from-motion reconstruction.

Finally, there are several publications on the topic of segmentation of structural components (arches, moldings, vaults, pillars, etc.) in point cloud data of cultural heritage, such as cathedrals and monastery churches (Cao & Scaioni, 2021; Croce et al., 2021; Malinverni et al., 2019). These sources all use outdated learning techniques that fall short of the performance expected of modern deep learning-based transformer models, and they evaluate their approaches only on limited, specialized data.

Additionally, several papers have addressed the segmentation of Mechanical, Electrical, and Plumbing (MEP) installations on small-scale personal datasets from industrial plants using heuristic methods, such as feature-based region growing for geometric shape recognition (Dimitrov & Golparvar-Fard, 2015; Yin et al., 2021), machine learning-based point cloud classification (Perez-Perez et al., 2021) and synthetic data augmentation to improve deep neural network prediction accuracy (Noichl et al., 2024). However, the limited availability of high-quality training and validation data remains one of the most significant barriers to advancing semantic segmentation for all AEC applications.

2.4 Multi-dataset synergistic training

The strategy of merging multiple data sources to train a single model collaboratively has shown promising results in scenarios with limited training resources (Wu, Tian, et al., 2023). Combining similar datasets to enrich the training pool has proven beneficial in 2D scene understanding, with studies (Kim et al., 2022; Wang et al., 2021) reporting improved model generalization on unseen datasets, even when label spaces differ. However, in the 3D domain, the significant domain gap and the sparsity of datasets can lead to negative transfer, potentially harming model performance when combined naively.

3. THE VALIDATION DATASET

For this study, we created a test dataset of diverse indoor scenes from a shell construction (SC) site for validation, which we refer to as SC-data below. Three spherical RGB point cloud-to-image renderings of representative scenes are shown in Figure 2. The dataset was collected using a FARO Focus M70 terrestrial laser scanner, a device commonly used in construction management and supervision to measure, document, and evaluate ongoing construction processes. Compared to mobile LiDAR systems, the terrestrial scanner produces a very high point density, high 3D accuracy, and a low noise range.

The dataset comprises 36 scenes from a multi-story residential building, including various apartment floor plans and different construction stages. The combined point clouds consist of approximately 3.6×10^8 points, with a

mean neighborhood point density of 6.67 mm (standard deviation = 0.002 mm), calculated based on the 30 nearest neighboring points.

Figure 2: Spherical Point Cloud Renderings of three rooms from the custom validation dataset, collected at a residential apartment building site during shell construction. a) Medium-sized room before plastering work is completed. b) Staircase before plastering work is completed. c) Medium-sized room after plastering work is completed. The scenes are characterized by varying surface textures due to the nature of the construction, challenging lighting conditions, and complex floor plans. They include obstacles and wet spots on the floor, which produce reflections and scanning artifacts. Yellow pixels represent empty canvas pixels where no points were projected.

This represents a very high point density and thus a particularly high level of detail compared to image-rendered (Zheng et al., 2020), surface-sampled (Armeni et al., 2016b; Park et al., 2017), or point cloud data sets from mobile lidar scanners (Guo et al., 2024). The scans were acquired over the course of a full day under naturally varying daylight conditions. The data include per-point RGB color values, laser reflection intensities, and point-wise semantic ground-truth labels for nine structural component classes and one background class ("none").

The scenes in the dataset vary in lighting conditions, surface finishes (e.g., brick walls, concrete, and plaster), and levels of contamination caused by interfering elements or rubble. As is typical for terrestrial laser scanning, the data also contain challenging effects such as reflection artifacts from specular surfaces (e.g., wet spots) and measurements associated with transparent objects (e.g., glass windows). The individual scans are not registered, and for each point set, the scanner center is located at the coordinate origin. The orientation around the z-axis does not follow a fixed cardinal direction and therefore appears quasi-random across scenes in the device output. Point cloud preprocessing was intentionally kept to a minimum. The only modification applied to the raw data was a Euclidean distance filter that excluded points more than 25 m from the coordinate origin. This step reduces the number of faulty measurements, often caused by highly reflective surfaces or other sensor disturbances, which would otherwise add no analytical value and may even corrupt the evaluation.

The semantic class set comprises *ceiling, floor, wall, beam, window, door, stairs, equipment, installation*, and a generic *none* class for points that could not be assigned reliably to one of the foreground categories. This class set was defined as a condensed selection of component groups that are both relevant for the technical description and

digital modeling of shell construction interiors and appear sufficiently frequent in construction-site data to justify their representation as independent semantic classes.

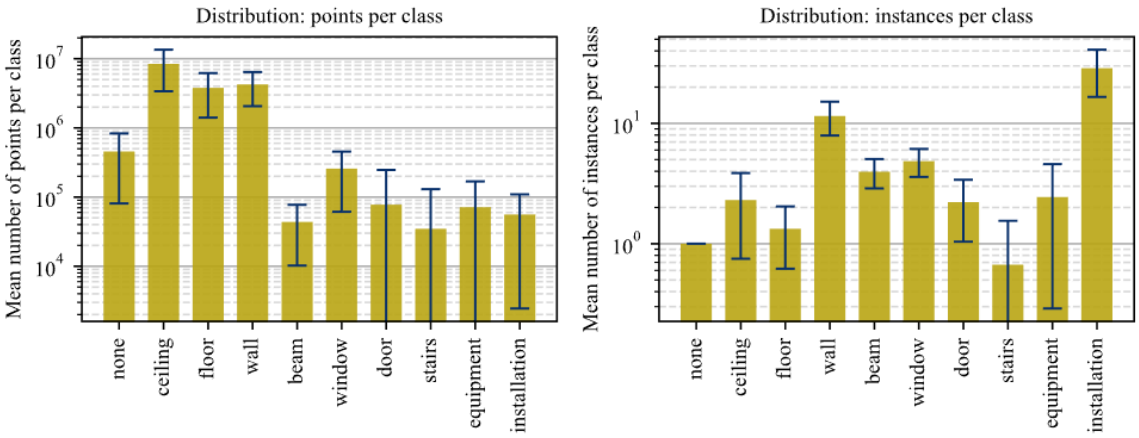

Figure 3: Statistical evaluation of the mean points-per-class distribution and mean instances-per-class distribution per scene in the validation dataset, plotted on a logarithmic scale. The uncertainty bars represent the standard deviation of the statistical sample distribution of the 36 scenes.

Point-wise semantic ground-truth labels were created manually in accordance with a predefined annotation guideline. The guideline documented each semantic class with textual definitions, decision rules for ambiguous cases, and visual examples in the form of screenshots and representative annotation excerpts. The document was maintained as an internal annotation protocol but can be provided upon request for reproducibility. Using this guideline as a rule set, a single domain expert with a civil engineering background annotated all scenes in the open-source software CloudCompare. Given the high cognitive and visual demands of manual point-wise annotation, work across different scenes was intentionally separated by extended breaks, where necessary, to reduce fatigue-related labeling errors and mitigate potential bias caused by day-to-day variability in annotator performance. Annotation was performed by iterative polygon-based segmentation in the 3D viewer. Point subsets were interactively selected and repeatedly refined from multiple viewing directions, enabling components to be separated in true 3D space with actual 3D geometric knowledge and reducing misinterpretations caused by occlusions or scan artifacts that may appear plausible from a single frontal view but become identifiable as noise from lateral perspectives. This is especially important near reflective surfaces and sharp edges, where LiDAR artifacts can appear plausible in 2D projections but are revealed as geometric deviations in the 3D representation.

A second review pass was performed to ensure quality control, after a temporal gap between annotation sessions, bearing in mind the results and scope of the first annotation pass. This gap was intentionally introduced to reduce immediate recall effects and to support a more independent consistency check by the same annotator. In the second pass, all scenes were re-inspected for potential labeling errors and, in particular, for consistency of class boundaries and class assignments across the dataset. Finally, automated sanity checks were applied to ensure that (i) every point was assigned to exactly one class and (ii) no isolated annotation fragment smaller than 10 points remained in the final labels. These checks helped identify residual labeling artifacts and incomplete segments that had been overlooked during manual quality control because of their small size. Fragments failing these checks were automatically reassigned to the background class. Despite these quality-control measures, no quantitative inter-annotator agreement metric was computed. Accordingly, while the single-annotator protocol supports consistent labeling across scenes, some annotator-specific bias may remain, particularly in ambiguous or partially occluded regions.

Figure 3 provides statistical evaluations of the points-per-class and instances-per-class distributions within the validation dataset. This representation helps visualize which classes hold how many points or instances and assess the dataset's internal balance. The wall, floor, and ceiling classes, which together constitute most interior surfaces, account for most of the individual points (94.3%) in the points-per-class view. Correspondingly, these large-area components appear less frequently in the instances-per-class view. Small component instances, such as electrical installation shafts, occur often but constitute only a few scan points. Rare component groups, such as stairs and

beams, are often underrepresented. The column class is not represented at all in this small validation dataset, as all 36 samples were collected from residential buildings where columns are uncommon. At this stage, the class serves as a placeholder for future experiments with more diverse construction sites. The authors acknowledge that this may impact certain averaging validation metrics, such as the mean intersection over union (mIoU). Therefore, the evaluations in Section 5 primarily consider the classes individually, providing deeper insights into the model's confusion and misclassification between them rather than quantifying performance solely through an aggregated metric.

It is recommended to split the dataset into 70/15/15 partitions for training, validation, and testing in supervised machine learning. We carefully selected the validation and test splits to ensure they contain genuinely unseen data, as multiple samples in the dataset may share common content from overlapping scans. The details of the dataset split used in the experiments are provided in Table 1.

Table 1: Data Split for the custom validation dataset used for all experiments in this paper. The 36 samples are split in a 70% / 15% / 15% ratio.

Task	Sample selection	Total sum
Training	0-11, 14, 15, 20-23, 30-35	24
Validation	16-19, 28, 29	6
Testing	12, 13, 24-27	6

4. EXPERIMENTAL DESIGN

With the following experiments, we assess how effectively laser scan data can be pre-labeled using existing transformer models and training data from out-of-domain indoor datasets. The objective is to reduce the manual annotation workload required for generating new data samples in the future. The experiments are designed to address challenges in civil engineering. They focus on pre-segmenting static laser scans of unfinished shell-construction sites to extract structural elements of the building's superstructure from the LiDAR point clouds.

To evaluate how well segmentation performance transfers to the construction-specific target data, we employ three of the latest point cloud transformer architectures: Point Transformer V2 (PTv2) (Wu et al., 2022), Point Transformer V3 (PTv3) (Wu, Jiang, et al., 2023), and Swin3D (Yang et al., 2023). These models are trained in three distinct phases, and their results are compared to assess expected segmentation quality and generalization capability as a pre-labeling engine. The experimental setup for all three phases is illustrated in Figure 4, Figure 5, and Figure 6.

The experiment design is intended to confirm the hypothesis that the available data resources are insufficient to train a component classification model. However, existing datasets are a useful resource for model pretraining, and even with only a few labeled construction-site scenes, Transformer models can be adapted to AEC pre-labeling tasks. A subsequent sensitivity analysis will be conducted to determine the number of additional data points required to calibrate a pretrained general-purpose transformer model for structural component segmentation of the example building class ‘residential dwelling’.

4.1 Phase 1 (baseline learning)

In Phase 1, we adopt a supervised learning approach to train and evaluate the three models for the semantic segmentation task on the shell construction-specific test SC-data. The three models are trained from scratch using the training split of the SC-dataset, which is listed in Table 1. After splitting the data into training, validation, and test sets (70%/15%/15%), only 24 scenes remain for training. Geometric data augmentation techniques are applied to make the most of limited training samples, but complex deep neural networks, such as transformer-based models, with many parameters often lead to overfitting when the training sample size is small. The Phase 1 experiments examine the expected performance of the three different Transformer models with limited data availability, assuming that the training and test data are relatively similar (as they come from the same or comparable construction sites), and whether the PTv3 model, due to its reduced complexity compared to PTv2, is less prone to overfitting under these conditions.

Point Transformer V2 and V3 are both iterations of the original Point Transformer (Zhao et al., 2021). While PTv2

is a direct improvement over Point Transformer by introducing new attention and pooling mechanisms, PTv3 represents a more fundamental redesign. It prioritizes simplicity and efficiency through serialization and streamlined components, enabling significant scaling. This results in notable improvements in speed, memory usage, and ultimate performance. Thereas PTv2 and PTv3 follow a point-based approach explicitly designed for 3D point cloud processing, Swin3D is a voxel-based method. It first converts the input point cloud into a sparse voxel grid and then operates on voxel-level features. The sliding window (Swin) mechanism adapts the Swin Transformer architecture (Z. Liu et al., 2021). Originally developed for 2D images, it was extended to sparse 3D voxel data by transferring its window-based attention concept to the 3D domain. This comparison aims to determine whether either approach (point-based or voxel-based) offers an advantage for the given application.

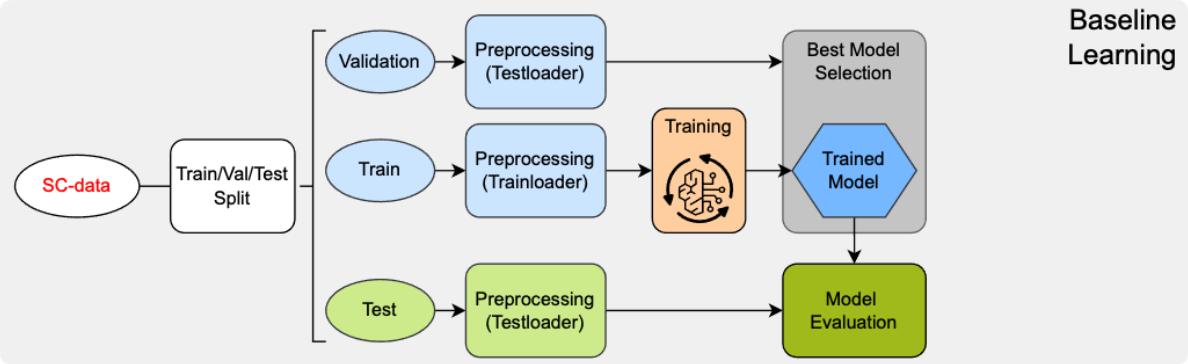

Figure 4: Supervised learning pipeline diagram for the Phase 1 experiments, illustrating the baseline learning workflow used to train and evaluate segmentation models on the SC-dataset.

Finally, the standard supervised learning in Phase 1 serves to establish a performance baseline for subsequent experiments in this study and to assess how well different models improve over classic training with the help of sophisticated training techniques and additional open-source indoor data.

4.2 Phase 2 (cross-domain learning)

In Phase 2, the same three models (PTv2, PTv3, and Swin3D) are trained using a cross-domain learning approach, each time on one of the four datasets: S3DIS (Armeni et al., 2016a), ScannetV2 (Dai et al., 2017), Structured3D (Zheng et al., 2020), and VASAD (Langlois et al., 2022). The cross-domain training procedure is visualized in Figure 5. This second experiment evaluates the segmentation performance on shared classes across different sources and target domains. The eight model/dataset combinations used in the cross-domain learning experiments are listed in Table 2.

Table 2: Dataset - model pairings that were evaluated in the cross-domain experiment.

	ScannetV2	S3DIS	Structured3D	VASAD
PTv2		✓	✓	
PTv3	✓	✓	✓	✓
Swin3D		✓	✓	

All four training datasets share one key characteristic: their point clouds are simulated LiDAR scans, either sampled points from reconstructed surface meshes or from digital CAD geometries. As a result, the sampled data lacks the measurement noise and scan artifacts typically present in real-world laser scans, such as those caused by reflective surfaces and ghost points.

This experiment is designed to evaluate the generalizability from synthetic indoor datasets (out-of-domain) to real-world laser scans of shell construction sites. Specifically, we assess whether semantic segmentation models can transfer knowledge from structured, noise-free environments to noisy, domain-specific TLS data. The cross-domain setting provides a controlled framework to test the robustness and adaptability of learned representations when applied to data with fundamentally different geometric and sensor characteristics. The cross-domain

performance of a machine learning model is defined as the model's performance when there is a distribution shift between the training and test datasets (Baktashmotlagh et al., 2013; Layeghy & Portmann, 2023). Such an approach requires either readily available common features or additional feature-extraction techniques to obtain a common (Booij et al., 2022) or domain-invariant (Layeghy & Portmann, 2023) feature set. In our experiments, the first case applies: we rely exclusively on object classes present in both the training and test SC-datasets and that can be matched directly. The objective remains to segment building component classes of the superstructure.

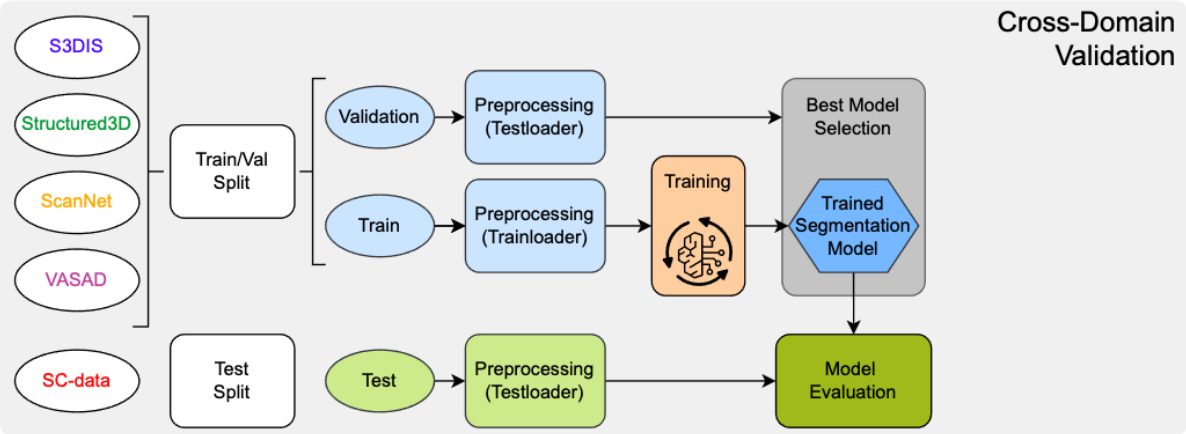

Figure 5: Supervised learning pipeline diagram for the Phase 2 cross-domain experiments, illustrating the workflow used to train segmentation models on four synthetic indoor datasets and evaluate them on SC test data.

Three of the four indoor datasets (S3DIS, ScannetV2, and Structured3D) include numerous classes related to furniture or interior objects, which are irrelevant to the segmentation of structural building components. Figure 1 visualizes the class-annotation overlap across all datasets. Two common strategies exist to handle such class mismatches: either all irrelevant classes are remapped to a shared fallback label (e.g., other = 0), resulting in a classification problem with K_relevant + 1 classes, or the original class labels are retained during training and later ignored during inference. In our setup, we choose the latter. All classes that do not overlap with any of the 10 target classes defined in the SC-dataset are preserved during training (to prevent the model from encountering a semantically indefinite collective term) but are excluded from evaluation and the calculation of performance metrics. Table 8 in the Appendix documents the mapping of the source classes during training and the matching target classes during testing. Source classes with no matching target in the test datasets are mapped to a generic "no_class" object. The results from the cross-domain performance evaluation are subsequently used to identify the most suitable model–dataset pairs for the transfer-learning experiments.

4.3 Phase 3 (transfer learning)

In Phase 3, four selected models from Phase 2 are fine-tuned on the training split of the test SC-dataset using a transfer learning-based knowledge-exchange approach. The transfer learning procedure is visualized in Figure 6. The limited-data point cloud segmentation approach is to learn a segmentor that classifies the query point cloud in terms of new classes, using only 24 support examples. In scenarios where the available amount of data is insufficient to train a deep network from scratch, transferring knowledge from a related pretrained model is a viable option. (Parnami & Lee, 2022) Transfer learning aims to improve a target learner's performance in a target domain (in our case, LiDAR scans from construction sites) by transferring knowledge from a different but related source domain, such as general indoor scenes. This reduces the dependency on large labeled datasets in the target domain. (Zhuang et al., 2021)

For our transfer learning experiments, only the PTv3 and Swin3D models pretrained on S3DIS and Structured3D are considered. The PTv2 model, the ScannetV2 dataset, and the VASAD dataset were excluded from Phase 3. PTv2 is excluded due to poor performance in Phase 1. The models trained on ScannetV2 demonstrated weak generalization in Phase 2, and the ScannetV2 dataset shares the fewest overlapping classes with the target domain. VASAD, being by far the smallest dataset, is not suitable for pretraining large transformer models.

The knowledge transfer in Figure 7, from a pretrained source-domain segmentor to a model operating in the target

domain, is enabled by backbone adoption. During transfer learning, a model is trained on a single task known as the source task in the source domain, where sufficient training data is available. This trained model is then again retrained or fine-tuned on another single task, known as the target task, in the target domain. (Pan & Yang, 2010; Parnami & Lee, 2022)

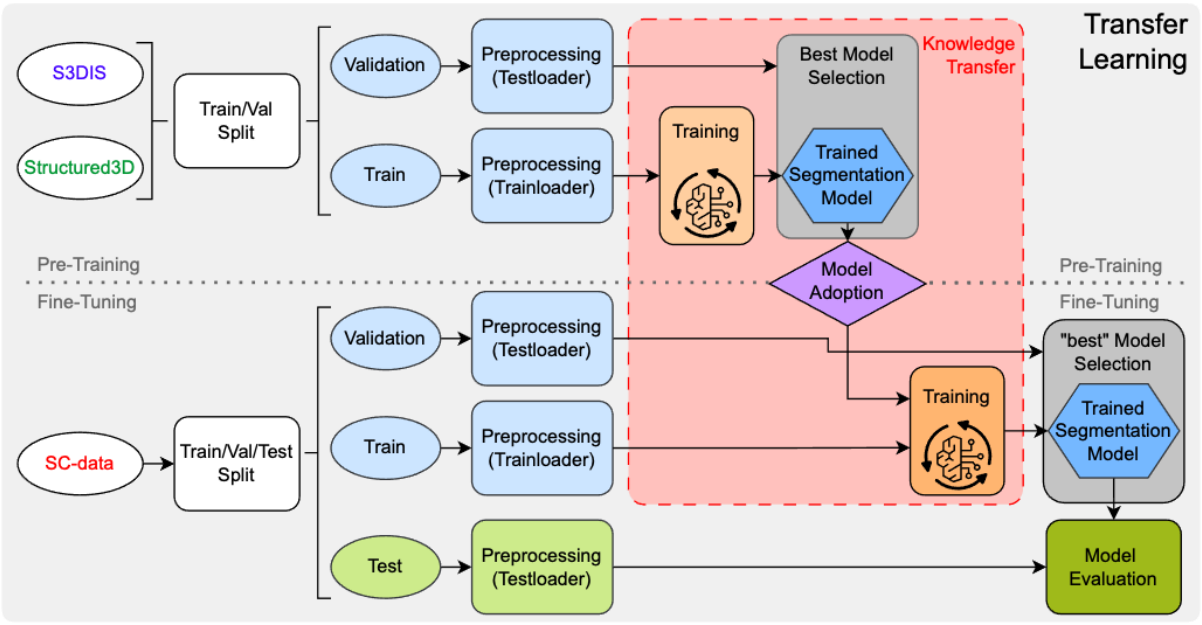

Figure 6: Transfer learning pipeline diagram for the Phase 3 experiments, showing model pretraining on general indoor datasets (S3DIS and Structured3D), followed by model adoption and fine-tuning on the SC-dataset using limited domain-specific data.

The source model's Phase 2-pretrained backbone is reused in the transfer-learning setup of our experiment. The backbone learned during pretraining to extract features independently of domain labels, while the segmentation head was trained to classify points based on domain-specific labels. In transfer learning, the source backbone is copied into the target model, thereby transferring general prior knowledge to the new task. Meanwhile, the new segmentation head is initialized with random parameters and an output segmentation map shaped to match the SC-data classes, ensuring the required number of semantic target classes. The new model is expected to use the previously learned representations in its backbone to compensate for the small size of the SC-dataset and improve segmentation accuracy. All model parameters remain trainable during fine-tuning, but we reduce the learning rate to 0.001 (compared to 0.006 in Phase 2) in order to prevent drastic changes to the pretrained backbone.

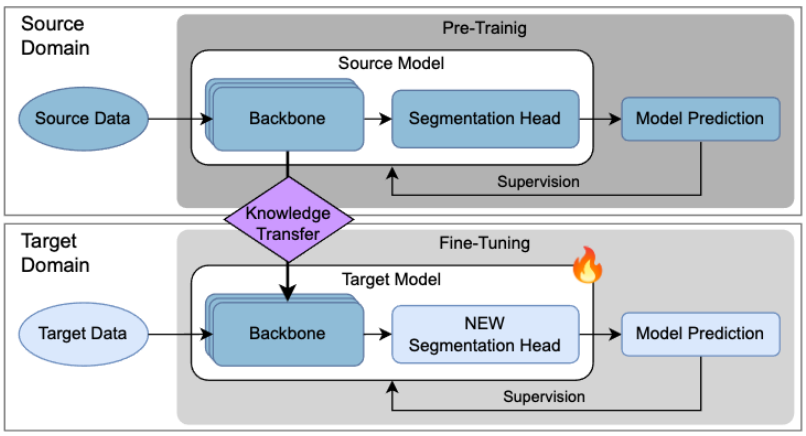

Figure 7: The concept of transfer learning is one in which the knowledge from the previous training is repurposed to improve the downstream tasks.

4.4 Data sensitivity study

A data sensitivity study is conducted in line with the transfer-learning experiments described in Section 4.3 to assess the influence of the number of samples used for fine-tuning. To conduct this study, we split the 24 samples from the training set in Table 1 into four equal parts and repeated supervised fine-tuning of the four pretrained models while increasing the training set size at each reduction stage. Table 3 summarizes the configuration and training set sizes. The maximum potential of transfer learning is achieved when segmentation performance, measured by mIoU, mAcc, or F1-score, converges despite the availability of more training samples.

Table 3: The configuration of the four-stage data sensitivity study.

		stage 1			stage 2			stage 3			stage 4		
model	**dataset**	**train set size**	**train epochs**	**val epochs**	**train set size**	**train epochs**	**val epochs**	**train set size**	**train epochs**	**val epochs**	**train set size**	**train epochs**	**val epochs**
ptv3	s3dis	**6**	200	10	**12**	200	10	**18**	200	10	**24**	800	40
swin3d	s3dis	**6**	200	10	**12**	200	10	**18**	200	10	**24**	800	40
ptv3	structured3d	**6**	200	10	**12**	200	10	**18**	200	10	**24**	800	40
swin3d	structured3d	**6**	200	10	**12**	200	10	**18**	200	10	**24**	800	40

Step by step, adding more data to the training pool and retraining the model with this additional information simulates the fictive concept of updating a classifier in an application scenario, as soon as new annotated data is available. The test results at each stage are tracked and evaluated to provide an informed assessment of the model's converging behavior, taking into account additional data from a similar construction site. In this example, sensitivity analysis is performed only on data from buildings classified as 'residential dwelling'. While the result is formally meaningful only within this specific building class, it is assumed that the model's convergence behavior generalizes to other building classes with similar geometric features.

In all experiments, the training batch size is set to 3 samples per GPU, and the learning rate is dynamically adjusted using a "one-cycle" learning rate scheduler with a maximum of 0.001 and a minimum of 0.0001. In the first three stages, the number of training epochs is limited to 200, which is sufficient for model convergence. In stage 4, the number of training epochs increased to 800 to achieve model convergence with the additional data. Validation on the six validation scenes (data split in Table 1) is performed every 20th epoch. The best validation model parameters from fine-tuning are used for testing at the end of each stage and as the starting point for the subsequent stage.

Based on the interim results from test Phases 1 and 2, only the two best-performing models, PTv3 and Swin3D, will be examined in these sensitivity tests, as was already the case in test Phase 3. The two models, PTv3 and Swin3D, are best paired with the two data sets for pretraining. The Structured3D dataset is the largest publicly available synthetic 3D dataset for indoor scenes. S3DIS contains significantly fewer training samples but consists of real-world scans.

4.5 Evaluation metric

The evaluation framework is designed to comprehensively assess point-wise segmentation performance, consistent with prevailing publications on indoor scene point cloud segmentation. (Hu et al., 2020; Thomas et al., 2019; Wu, Jiang, et al., 2023) We principally report the mean Intersection over Union (mIoU) as the primary evaluation metric, where applicable, and provide the mean class-wise accuracy (mAcc) for reference. Fundamental to various performance metrics is the confusion matrix for classification problems, which compares positive and negative predictions against the two possible outcomes, true and false (Liang et al., 2021).

Accuracy (Acc) is defined as the ratio of correctly predicted points (true positives and true negatives) to the total number of predictions.

$$Acc = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

While accuracy is a simple and widely used metric, it does not account for false positives and is therefore less informative in the presence of class imbalance.

Precision gauges how precise the model is, i.e., it measures the quality of the predictions the model makes. The precision is the fraction of relevant instances among the retrieved instances, and is written as a formula according to the four possible outcomes of the confusion matrix as follow.

$$Precision = \frac{TP}{PP} = \frac{TP}{TP + FP} \tag{2}$$

Precision quantifies the proportion of positive predictions that are actually positive, and thus reflects how well a model predicts positive observations. However, this neglects negative predictions. Recall, also referred to as sensitivity, is the fraction of relevant instances retrieved.

$$Recall = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{3}$$

The recall metric measures how well the model correctly predicts all possible positive observations, relative to the total number of real positives. If precision corresponds to the quality of the model, recall is akin to quantity, i.e., the proportion of true labels found. Precision and recall are not particularly useful metrics when used in isolation. For multi-class classification, per-class recall is equivalent to the per-class accuracy.

The F1-score, also known as the Dice Similarity Coefficient (DSC), is the harmonic mean between precision and recall, which penalizes extreme differences. There appears to be an unavoidable trade-off in retrieval performance between precision and recall, even though having high values of both at the same time would be preferable (Buckland & Gey, 1994). The F1-score combines the two metrics into a single number to address this inverse relationship, making it suitable when precision and recall are in trade-off.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2\,TP}{2\,TP + FP + FN} \tag{4}$$

In point cloud analysis, Intersection over Union (IoU) measures the overlap between the predicted mask and the labeled point cloud. It is used to assess the accuracy of the predicted segmentation area.

$$IoU = \frac{TP}{TP + FP + FN} \tag{5}$$

Due to its formulation, IoU penalizes individual misclassifications more strongly and is therefore better suited to evaluating segmentation performance on imbalanced data. The mean Intersection over Union (mIoU) is the arithmetic mean of the IoU values across all classes and is used to evaluate segmentation performance over the entire dataset. Because it is calculated as an average, each class contributes equally to the final score, regardless of the number of points per class.

Note the similarity between the F1-score and IoU. Although the F1-score essentially doubles the contribution of true positive predictions to the overall score, the two measures are monotonically related.

$$IoU = \frac{F1\text{-}score}{2 - F1\text{-}score} \tag{6}$$

This implies that the two metrics are positively correlated for any fixed ground truth. Both metrics converge to 0 and 1 at the extreme values. However, because true positives have a stronger influence, the F1-score generally yields slightly higher values than IoU, particularly for moderate overlaps, whereas IoU penalizes misclassifications more conservatively. The choice between them ultimately depends on community conventions and the research domain. Consistent with the majority of publications in 3D computer vision, this thesis favors the more conservative IoU metric. In the experiments for Phases 1 and 3, where all training classes match the evaluation target classes, we use mIoU as the primary performance metric and report class-wise recall (mean accuracy, mAcc), and F1-score, as well as overall accuracy (OA) as additional metrics. Using mIoU in our cross-domain experiments (Phase 2) is not meaningful. Since the source classes of the four training datasets do not fully overlap with the target classes of the SC-dataset, the mIoU metric is not interpretable in this setting. Instead, we report individual IoU scores per structural component class and analyze the multi-class confusion matrix.

4.6 Experimental setup

The decision to conduct the experiments using these three models was based on a detailed comparison of the most successful approaches over the past few years in indoor semantic segmentation. (Rauch & Braml, 2023) The experiments focus on transformer-based backbones because they have demonstrated strong performance in 3D semantic segmentation and are particularly well-suited to capturing long-range spatial context, which is critical in sparse, occluded, large-scale scans. Point Transformer V2 (Wu et al., 2022) and V3 (Wu, Jiang, et al., 2023) represent point-wise attention models with different design trade-offs, while Swin3D (Yang et al., 2023) extends the hierarchical Swin paradigm to 3D data and provides an efficient multi-scale representation. These properties are particularly relevant for terrestrial laser scans of shell construction sites, where occlusions, varying density, and long-range context are common.

We additionally conducted a baseline screening study using a reduced experimental setup to compare the selected transformer models with several representative non-transformer approaches. The methods compared were selected to span different modeling families and were trained under the same protocol as described in the Phase 1 experiments. The screening included the following model architectures for comparison: PointNet (Charles et al., 2017) and PointNet++ (Qi et al., 2017) to represent simple segmentation models; PointNeXt (Qian et al., 2022) in the s/b/l/xl configurations to represent competitive MLP backbones; and DeepGCN (Li et al., 2019) and DGCNN (Wang et al., 2019) for graph-based point cloud segmentation.

Across this screening, PTv3 and Swin3D transformer backbones consistently achieved high accuracy scores and exhibited the favorable convergence behavior in our low-data supervised-learning setting. The simple Pointnet and Pointnet++ models performed below average, most likely due to the small receptive fields of static MLPs. The two graph-based models failed in screening due to hardware limitations and the enormous memory required to construct graphs for large point clouds with several million points. Spherical sampling was implemented in the same way as for the transformers, but with an even smaller sample size of 62k points; however, fragmentation had a significant negative impact on the result. The MPL-based architecture of the modern PointNeXt proved to be on par with the transformers, but only in its L/XL configurations with a large number of deep layers, which relativizes its designation as a simple model. Since the screening is intended to justify model selection rather than to make an additional contribution, we provide detailed results only in the Appendix.

We adopt the implementations of the two Point Transformers (Wu et al., 2022; Wu, Jiang, et al., 2023) and the Swin3D (Yang et al., 2023) model from the Pointcept library (Pointcept Contributors, 2023) to carry out the experiments. The Pointcept library provides comprehensive integration of transformer backbones and training pipelines for a variety of point cloud perception tasks, particularly semantic segmentation. To handle large terrestrial laser scan (TLS) point clouds in the training and evaluation pipeline, we implement a custom data loader and a dynamic label translation layer to address class mismatches during cross-domain evaluation. The sheer size of the raw TLS point clouds, with several million points per scan, poses significant challenges for GPU memory and training time. Feeding an entire point cloud into the model is infeasible. Instead, spherical samples of 100,000 points are drawn around random query points during training. This is necessary to prevent video memory overflow and also allows multiple samples to be generated from a single scan, thereby significantly increasing the size of the training pool. During each epoch, model evaluation is performed on a single grid-based subsample of the full point cloud, providing an initial assessment of performance across the entire scan. For final testing, a dense point cloud is fragmented into a sequence of subsamples to ensure complete spatial coverage. Predictions are made on each fragment, and the results are aggregated through class voting, resulting in a full-scene prediction. To further increase data diversity, data augmentation is applied during training in all three phases. Geometric 3D augmentations include center shift, random dropout, random rotation, random flip, random jitter, and voxelization (voxel size: 2.5 cm). Augmentations in feature space include: chromatic auto-contrast, chromatic translation, chromatic jitter, and color normalization.

All models are trained from scratch using identical training parameters. Although Pointcept (Pointcept Contributors, 2023) provides pretrained PTv2 and PTv3 models trained on the S3DIS dataset, we cannot use them because our SC-dataset does not include surface normal vectors. Consequently, we retrain all models using only coordinate and RGB color features, omitting surface normals from the S3DIS, Structured3D, and ScannetV2 datasets. Training is distributed across multiple NVIDIA Tesla V100 SXM3 GPUs, each with 32 GB of video memory.

5. RESULTS

The results from the three phases of supervised training are evaluated to assess how well the transformer models can be adapted using very limited training data to achieve stable segmentation performance for future pre-labeling tasks.

5.1 Phase 1 – baseline evaluation

The baseline learning result serves as the initial benchmark for all subsequent experimental phases. All three models were trained exclusively on the limited training split of the SC-dataset, comprising 24 scenes, to assess the performance achievable without any external or pretraining data. The baseline performance, measured using the intersection over union, accuracy, and F1-score metrics, is summarized in Table 4.

Table 4: Baseline training and test results for 3D semantic segmentation using three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. All models are trained and evaluated on the SC-dataset. Performance metrics include the overall accuracy (OA), class-wise Intersection over Union (IoU), Accuracy (Acc, also referred to as Recall), and F1-score. The best per-class and overall IoU results are highlighted.

class-wise TEST results

	PTv2			**PTv3**			**Swin3D**		
class	**IoU**	**Acc**	**F1**	**IoU**	**Acc**	**F1**	**IoU**	**Acc**	**F1**
none	**0.50**	0.60	0.65	**0.69**	0.76	0.77	**0.60**	0.85	0.64
ceiling	**0.69**	1.00	0.46	**0.97**	0.97	0.96	**0.98**	0.98	0.94
floor	**0.05**	0.05	0.09	**0.98**	1.00	0.84	**0.99**	1.00	0.92
wall	**0.89**	0.98	0.43	**0.97**	0.99	0.74	**0.95**	0.98	0.70
beam	**0.27**	0.97	0.90	**0.57**	0.96	0.94	**0.41**	0.70	0.79
window	**0.19**	0.19	0.32	**0.71**	0.80	0.85	**0.55**	0.61	0.72
door	**0.07**	0.07	0.12	**0.81**	0.87	0.93	**0.56**	0.58	0.73
stairs	**0.14**	0.14	0.25	**0.84**	0.93	0.96	**0.86**	0.90	0.94
equipment	**0.07**	0.07	0.14	**0.61**	0.64	0.77	**0.58**	0.67	0.80
installation	**0.35**	0.44	0.60	**0.07**	0.52	0.67	**0.14**	0.61	0.75

average TEST results

OA	**mIoU**	**mAcc**	**mF1**	**OA**	**mIoU**	**mAcc**	**mF1**	**OA**	**mIoU**	**mAcc**	**mF1**
0.75	**0.32**	0.45	0.40	0.97	**0.72**	0.84	0.84	0.97	**0.66**	0.79	0.79

average TRAIN results

OA	**mIoU**	**mAcc**	**mF1**	**OA**	**mIoU**	**mAcc**	**mF1**	**OA**	**mIoU**	**mAcc**	**mF1**
0.97	**0.74**	0.82	-	0.95	**0.67**	0.80	-	0.93	**0.55**	0.63	-

In conclusion, PTv2 achieves the highest validation score during training, with an mIoU of 0.74 and an mAcc of 0.82. However, its performance drops by half during final testing, reaching only 0.32 in mIoU and 0.45 in accuracy, indicating poor generalization. PTv3, on the other hand, shows more stable behavior, with training scores of 0.67 (mIoU) and 0.80 (mAcc), and improved final test scores of 0.72 (mIoU) and 0.84 (mAcc), respectively. Swin3D performs slightly worse during training but maintains solid test results, achieving 0.66 (mIoU) and 0.79 (mAcc).

The classes that contribute positively to the overall score are the large planar structures, such as ceilings, floors, walls, and, to some extent, stairs. On the other hand, structural components such as beams, doors, and windows negatively affect overall performance, except for PTv3, which demonstrates better generalization. Segmentation of construction site equipment and smaller installation-related objects performs poorly across all models, often leading to complete misclassification. A notable failure mode is exhibited by PTv2, which, despite comparable performance to PTv3 and Swin3D during training, shows a significant decline in final testing, achieving a high intersection over union value only for the wall class.

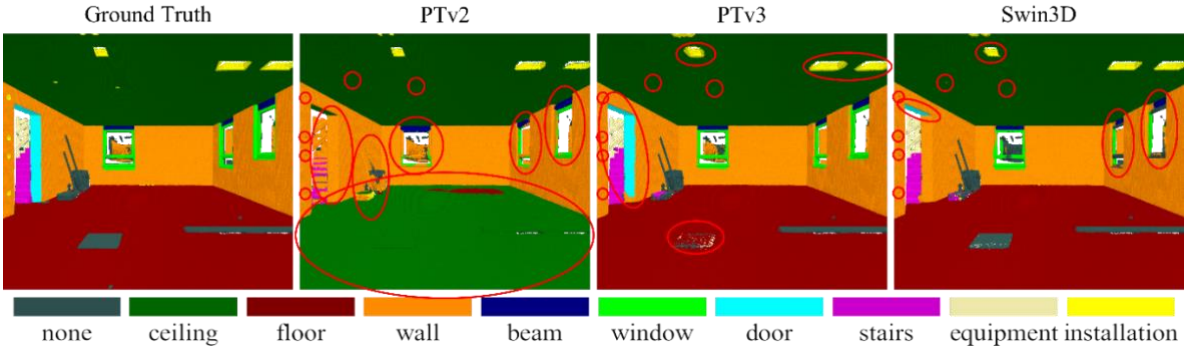

Figure 8: Baseline test results for 3D semantic segmentation using three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. The models were trained and tested on the SC-dataset, which focuses on scenes from shell construction sites. The figure shows one representative scene from the test dataset. Each semantic class is visualized in a distinct color, with the color legend displayed at the bottom. The first column displays the ground truth for reference. Incorrect or imprecise segmentations are marked with red ellipses. A complete version of this figure is provided in the appendix.

Table 5: Cross-domain test results for 3D semantic segmentation using three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. The models are trained on four public indoor datasets (S3DIS, ScannetV2, Structured3D, and VASAD) and evaluated on the SC-dataset. Performance metrics include class-wise Intersection over Union (IoU), Accuracy (Acc, also referred to as Recall), and F1-score. The best per-class and IoU results are highlighted. A hyphen (–) indicates that no matching prediction exists between the source and target class definitions.

	S3DIS									**ScannetV2**		
	PTv2			PTv3			Swin3D			PTv3		
Class	**IoU**	**Acc**	**F1**	**IoU**	**Acc**	**F1**	**IoU**	**Acc**	**F1**	**IoU**	**Acc**	**F1**
none	**0.06**	0.11	0.08	**0.04**	0.16	0.10	**0.14**	0.33	0.21	-	-	-
ceiling	**0.95**	0.97	0.75	**0.73**	0.77	0.63	**0.96**	0.99	0.78	-	-	-
floor	**0.87**	0.89	0.73	**0.59**	0.90	0.62	**0.84**	0.86	0.73	**0.35**	1.00	0.47
wall	**0.73**	0.79	0.42	**0.65**	0.71	0.41	**0.69**	0.81	0.30	**0.61**	0.87	0.35
beam	**0**	0	0	**0**	0	0	**0**	0	0	-	-	-
window	**0.16**	0.50	0.36	**0.25**	0.39	0.30	**0.03**	0.03	0.05	**0.29**	0.57	0.47
door	**0.05**	0.06	0.12	**0.11**	0.15	0.26	**0**	0	0	**0.33**	0.74	0.71
stairs	-	-	-	-	-	-	-	-	-	-	-	-
equipment	-	-	-	-	-	-	-	-	-	-	-	-
installation	-	-	-	-	-	-	-	-	-	-	-	-

	Structured3D									**VASAD**		
	PTv2			PTv3			Swin3D			PTv3		
Class	**IoU**	**Acc**	**F1**	**IoU**	**Acc**	**F1**	**IoU**	**Acc**	**F1**	**IoU**	**Acc**	**F1**
none	-	-	-	-	-	-	-	-	-	-	-	-
ceiling	**0.98**	0.99	0.84	**0.99**	0.99	0.80	**0.99**	1.00	0.83	0	0	0
floor	**0.94**	0.97	0.88	**0.97**	0.98	0.91	**0.97**	0.99	0.82	**0.30**	0.99	0.47
wall	**0.94**	0.99	0.51	**0.37**	0.38	0.32	**0.95**	0.99	0.53	**0.79**	0.87	0.41
beam	-	-	-	-	-	-	-	-	-	**0.09**	0.64	0.58
window	**0.35**	0.55	0.52	**0.35**	0.56	0.46	**0.36**	0.68	0.56	**0.15**	0.39	0.43
door	**0.01**	0.01	0.01	**0.01**	0.01	0.01	0	0	0.01	**0.00**	0.00	0.00
stairs	-	-	-	-	-	-	-	-	-	**0.28**	0.78	0.73
equipment	-	-	-	-	-	-	-	-	-	-	-	-
installation	-	-	-	-	-	-	-	-	-	-	-	-

Figure 12 in the Appendix shows qualitative results of the baseline experiment across five representative scenes. A selected scene is shown in Figure 8 to contextualize these results. Each semantic class is visualized in a distinct color; the color legend is shown at the bottom of the image, and the model identifier is displayed at the top of each grid image. The first column shows the ground truth labels for reference. The selected scene illustrates the overall result, and the phenomena mentioned are also observed in other scenes.

PTv2 misclassifies the floor as the ceiling, resulting in low IoU for the floor class. By contrast, PTv3 and Swin3D consistently perform well at separating ceiling, floor, and wall points. However, all three models struggle to separate large dominant segments from smaller adjacent structures. Segment contours are often blurred, particularly in areas such as ceiling installations and around door openings, where the wall class extends into the doorframe area. Small details, such as electrical boxes adjacent to the door, are often integrated into the surrounding wall segments. Window segmentation remains a persistent challenge, with only partial detections and a general lack of accuracy across models.

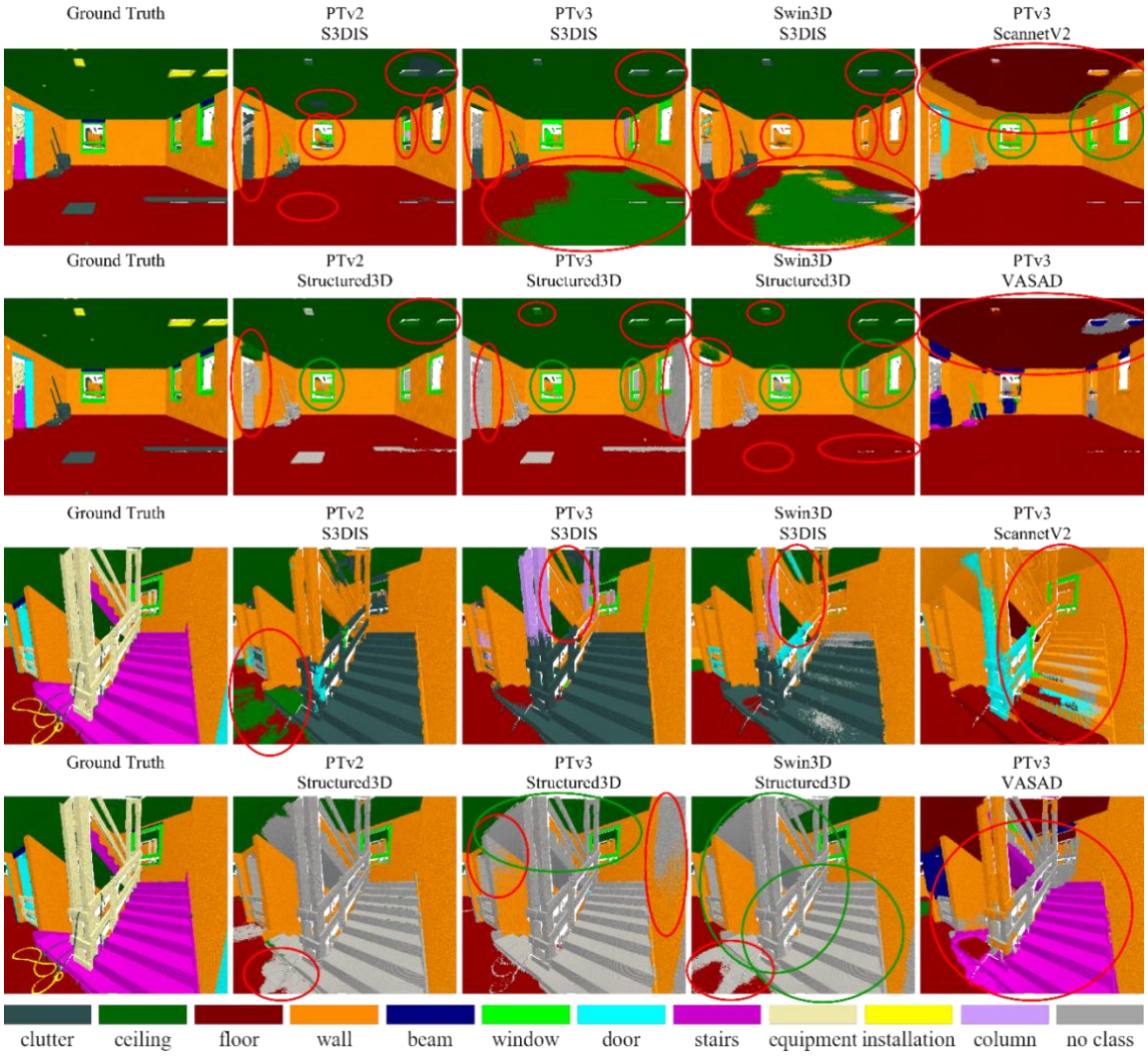

Figure 9: Cross-domain test results for 3D semantic segmentation using three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. The models are trained on four public indoor datasets (S3DIS, ScannetV2, Structured3D, and VASAD) and tested on the SC-dataset, which focuses on scenes from shell construction sites. The figure displays two representative scenes from the test set. Each semantic class is visualized using a unique color; the color legend is shown at the bottom. Source-domain class labels with no corresponding target in the SC-dataset are mapped to the auxiliary category no_class (gray). The first column shows the ground truth for comparison. Incorrect or imprecise segmentations are highlighted with red ellipses. A complete version of this figure is provided in the appendix.

5.2 Phase 2 – cross-domain evaluation

In Phase 2, cross-domain learning is evaluated by training models on out-of-domain datasets and inferring from TLS laser scans, with a focus on common object classes shared across domains. The three models, PTv2, PTv3, and Swin3D, were trained on the datasets S3DIS, ScannetV2, Structured3D, and VASAD using only coordinate and RGB color features. For cross-domain testing on the SC-dataset, semantic class predictions were generated from the source-domain class labels and subsequently mapped to the corresponding target classes in the SC-dataset to ensure a standardized comparison. The results, measured by the three metrics intersection over union, accuracy, and F1-score, are summarized in Table 5. A hyphen (-) indicates that no matching prediction exists between the source and target class definitions.

Certain target classes, such as installation, equipment, stair, and beam, could not be segmented by the models in most configurations because their class labels are not present in the S3DIS, ScannetV2, or Structured3D training datasets. Although VASAD includes dedicated labels for beam and stairs, the achieved segmentation quality is very low, with IoU scores of 0.09 and 0.28, respectively. For the ceiling class, high IoU segmentation scores of up to 0.99 are achieved when models are trained on either S3DIS or Structured3D, except for the PTv3+S3DIS combination, which falls short at 0.73 IoU. Segmentation results for the floor class are most inconsistent, ranging from values of 0.94 IoU and above (for models trained on Structured3D) to as low as 0.3 (for PTv3 trained on ScannetV2 or VASAD). In contrast, the segmentation of wall elements is comparatively uniform, albeit with less favorable results: Models trained on S3DIS achieve an average IoU of 0.70 on vertical walls. PTv3 is slightly better than this when trained on VASAD and slightly worse when trained on ScannetV2. However, when trained on Structured3D, the PTv2 and Swin3D models achieve excellent scores for the wall class, whereas PTv3 fails. Window and door elements cannot be effectively segmented in the cross-domain setting, with IoU consistently below 0.35. Construction equipment and technical building installations are not considered because they are not represented in any of the four general indoor datasets.

Figure 13 and Figure 14 in the Appendix present qualitative results from five representative SC-dataset scenes. Two selected scenes are shown in Figure 10 to illustrate these findings in context. All source-domain classes that do not correspond to any target class in the SC-dataset are assigned to a no_class category. The class mapping used for this translation is documented in the Appendix Table 8. Given that the ScannetV2 dataset lacks a dedicated ceiling class, the model is not trained to recognize ceiling regions. As a result, ceiling points are typically assigned to either the floor or the nearest wall class, leading to fewer, larger segments with reduced structural detail. Similarly, the PTv3+VASAD model misclassifies ceiling areas, though for a different reason. The original VASAD dataset provides annotations for both slab and suspended ceiling classes. During the construction phase, however, both ceilings and floors are typically formed from concrete slabs, as the model correctly identifies. This labeling scheme, however, does not allow a clear distinction between the slab's intended use (ceiling or floor) or its functional role within the building structure.

Overall, cross-domain knowledge transfer from S3DIS and Structured3D to TLS laser scans remains inconsistent. While basic structures and even some complex elements, such as windows, are recognized in certain scenes, all models frequently fail to segment walls and floors correctly. In room-scale scenes, most structural components are correctly segmented. However, the staircase scene demonstrates the limitations of cross-domain learning: only VASAD includes a stair class, which PTv3 can detect, albeit with imprecise segment boundaries. In more complex scenes with varying ground-floor levels, tilted ceilings, occluded or poorly visible regions, and nested or irregular floor plans, segmentation quality degrades significantly.

5.3 Phase 3 – transfer learning evaluation

The effectiveness of the knowledge transfer segmentation approach is assessed by evaluating the transfer of knowledge from out-of-domain datasets to real-world TLS scans of shell construction sites. Four model/dataset combinations were examined. In Phase 2, the PTv3 and Swin3D models were pretrained on the S3DIS and Structured3D indoor datasets. In Phase 3, these pretrained models were further fine-tuned using the limited training split of 24 scenes from the SC-dataset. The performance results, measured by mIoU and mAcc, are summarized in Table 6.

On average, all four configurations achieve a mean IoU above 0.80, with Swin3D pretrained on Structured3D yielding the highest results at 0.84 (mIoU) and 0.91 (mAcc). Per-class analysis shows that planar components such

as ceilings, walls, and floors consistently reach IoU scores of 0.97 or higher across all configurations. For beam objects, results range from 0.57 IoU (PTv3 with Structured3D pretraining) to 0.63 IoU (Swin3D with the same pretraining). Classification performance for the stair class shows minimal variance, with IoU scores around 0.90 across all models. In contrast, the installation class exhibits the highest variation and the lowest overall scores, with a maximum IoU of only 0.63.

Table 6: Transfer learning test results for 3D semantic segmentation using two model architectures: Point Transformer V3 (PTv3) and Swin3D. The models are pretrained on two public indoor datasets (S3DIS and Structured3D), then fine-tuned and evaluated on the SC-dataset. Performance metrics include the overall accuracy (OA), class-wise Intersection over Union (IoU), Accuracy (Acc, also referred to as Recall), and F1-score. Percentage values right next to the performance number show the improvement in relation to the baseline in Table 4. The best per-class and overall IoU results are highlighted.

class-wise TEST results

	S3DIS											
	PTv3+SC						**Swin3D+SC**					
class	**IoU**		**Acc**		**F1**		**IoU**		**Acc**		**F1**	
none	**0.73**	*+4%*	**0.84**	*+8%*	**0.79**	*+2%*	**0.75**	*+15%*	**0.82**	*-3%*	**0.81**	*+17%*
ceiling	**1.00**	*+3%*	**1.00**	*+3%*	**0.96**	*–0%*	**0.99**	*+1%*	**0.99**	*+1%*	**0.97**	*+3%*
floor	**0.99**	*+0%*	**1.00**	*-0%*	**0.93**	*+9%*	**<u>0.99</u>**	*+0%*	**1.00**	*+0%*	**0.93**	*+1%*
wall	**0.98**	*+1%*	**0.99**	*+0%*	**0.83**	*+9%*	**0.98**	*+2%*	**0.99**	*+2%*	**0.80**	*+11%*
beam	**0.59**	*+2%*	**0.95**	*–1%*	**0.94**	*+0%*	**0.60**	*+19%*	**0.95**	*+26%*	**0.95**	*+16%*
window	**0.74**	*+3%*	**0.78**	*–2%*	**0.85**	*+1%*	**0.79**	*+25%*	**0.85**	*+24%*	**0.88**	*+16%*
door	**0.83**	*+2%*	**0.92**	*+5%*	**0.94**	*+2%*	**0.92**	*+36%*	**0.95**	*+37%*	**0.97**	*+24%*
stairs	**0.91**	*+6%*	**0.93**	*–0%*	**0.96**	*+1%*	**<u>0.91</u>**	*+5%*	**0.94**	*+4%*	**0.97**	*+2%*
equipment	**0.75**	*+14%*	**0.78**	*+15%*	**0.88**	*+10%*	**0.72**	*+14%*	**0.73**	*+6%*	**0.84**	*+4%*
installation	**0.54**	*+47%*	**0.71**	*+19%*	**0.82**	*+15%*	**0.36**	*+21%*	**0.73**	*+12%*	**0.84**	*+9%*

average TEST results

OA		**mIoU**		**mAcc**		**mF1**		**OA**		**mIoU**		**mAcc**		**mF1**	
0.99	*+2%*	**0.80**	*+8%*	**0.89**	*+5%*	**0.89**	*+5%*	**0.99**	*+2%*	**0.80**	*+14%*	**0.90**	*+11%*	**0.90**	*+10%*

class-wise TEST results

	Structured3D											
	PTv3+SC						**Swin3D+SC**					
class	**IoU**		**Acc**		**F1**		**IoU**		**Acc**		**F1**	
none	**0.75**	*+6%*	**0.82**	*+7%*	**0.82**	*+5%*	**<u>0.75</u>**	*+15%*	**0.81**	*–4%*	**0.82**	*+17%*
ceiling	**1.00**	*+3%*	**1.00**	*+3%*	**0.97**	*+1%*	**<u>1.00</u>**	*+2%*	**1.00**	*+2%*	**0.96**	*+2%*
floor	**0.99**	*+1%*	**1.00**	*–0%*	**0.94**	*+10%*	**0.99**	*–0%*	**1.00**	*+0%*	**0.93**	*+1%*
wall	**0.98**	*+1%*	**0.99**	*+0%*	**0.85**	*+11%*	**<u>0.98</u>**	*+3%*	**0.99**	*+2%*	**0.87**	*+18%*
beam	**0.57**	*+0%*	**0.98**	*+2%*	**0.95**	*+1%*	**<u>0.64</u>**	*+23%*	**0.98**	*+28%*	**0.96**	*+17%*
window	**0.81**	*+10%*	**0.85**	*+5%*	**0.90**	*+5%*	**<u>0.85</u>**	*+30%*	**0.90**	*+30%*	**0.92**	*+20%*
door	**0.91**	*+10%*	**0.97**	*+10%*	**0.98**	*+6%*	**<u>0.93</u>**	*+38%*	**0.96**	*+39%*	**0.98**	*+25%*
stairs	**0.90**	*+6%*	**0.96**	*+2%*	**0.97**	*+2%*	**0.88**	*+2%*	**0.96**	*+6%*	**0.97**	*+3%*
equipment	**0.75**	*+14%*	**0.79**	*+15%*	**0.88**	*+10%*	**<u>0.79</u>**	*+21%*	**0.82**	*+15%*	**0.90**	*+10%*
installation	**0.62**	*+55%*	**0.76**	*+24%*	**0.86**	*+19%*	**<u>0.63</u>**	*+49%*	**0.76**	*+15%*	**0.86**	*+11%*

average TEST results

OA		**mIoU**		**mAcc**		**mF1**		**OA**		**mIoU**		**mAcc**		**mF1**	
0.99	*+4%*	**0.83**	*+10%*	**0.91**	*+7%*	**0.91**	*+7%*	**0.99**	*+6%*	**<u>0.84</u>**	*+18%*	**0.92**	*+13%*	**0.92**	*+12%*

Figure 15 in the Appendix presents qualitative results from five representative scenes. A selection of two scenes is shown in Figure 10 to contextualize the results from Table 6, and to visualize improvements over the baseline. The pretrained knowledge transfer setup yields consistent predictions for the fundamental components of the building superstructure across all model/dataset combinations. Under visual inspection, most windows and door frames are

segmented with high confidence and refined contours. After fine-tuning on limited-domain data, the transformers segment even small objects, such as electrical box installations in ceilings and tiny power inlets near doors. Temporary elements can now be separated from the floor plane, and fine-tuning appears sufficient for recognizing building-type-specific components, such as lintel beams above windows. In the challenging stairwell scene, all models successfully segment the lower staircase section with improved edge detail. However, segmentation of the upper staircase remains problematic due to occlusions caused by temporary railings, which all models struggle with.

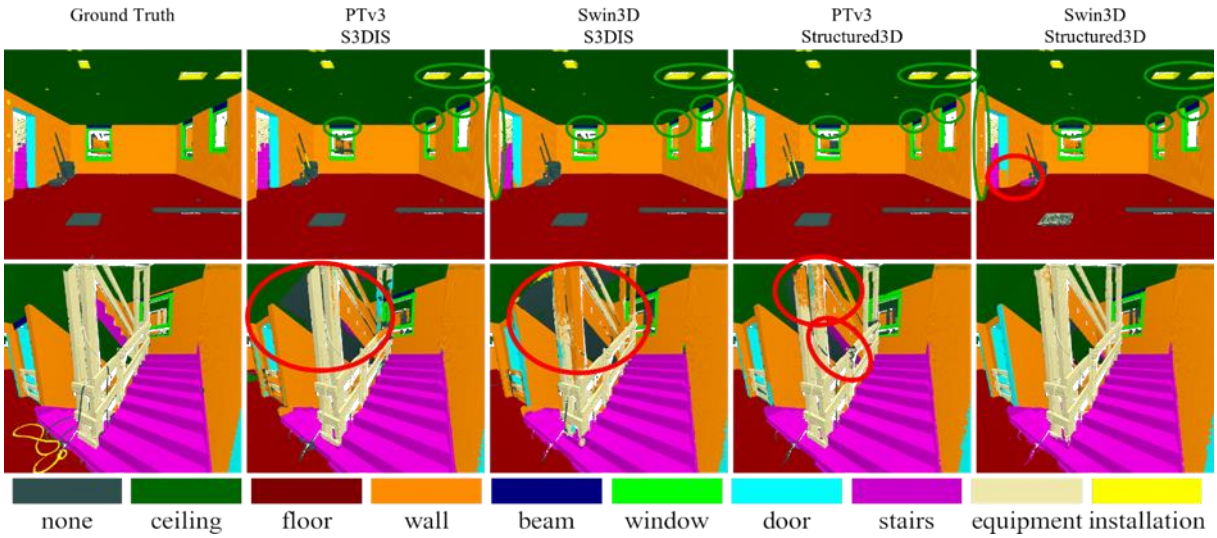

Figure 10: Transfer learning test results for 3D semantic segmentation using two model architectures: Point Transformer V3 (PTv3) and Swin3D. The models are trained on two public indoor datasets (S3DIS and Structured3D), then fine-tuned and tested on the SC-dataset, which focuses on scenes from shell construction sites. The figure displays two representative scenes from the test set. Each semantic class is visualized using a unique color; the color legend is shown at the bottom. The first column shows the ground truth for reference. Incorrect or imprecise segmentations are highlighted in red, while correct predictions are marked in green. A complete version of this figure is provided in the appendix.

Table 7: Results of the data sensitivity study for 3D semantic segmentation using two model architectures: Point Transformer V3 (PTv3) and Swin3D. The models are pretrained on two public indoor datasets (S3DIS and Structured3D) and then fine-tuned incrementally with varying numbers of domain-specific training samples. Evaluation is performed on the SC-dataset. The four stages correspond to training data sets of 6, 12, 18, and 24 samples. Performance metrics include mean Intersection over Union (mIoU), mean class Accuracy (mAcc, also referred to as mean Recall), and mean class F1-score (mF1). The best result for each model–dataset pairing is highlighted with an underscore.

		stage 1				stage 2				stage 3				stage 4			
model	**dataset**	**OA**	**mIoU**	**mAcc**	**mF1**	**OA**	**mIoU**	**mAcc**	**mF1**	**OA**	**mIoU**	**mAcc**	**mF1**	**OA**	**mIoU**	**mAcc**	**mF1**
PTv3	S3DIS	0.96	**0.51**	0.61	0.56	0.98	**0.76**	0.87	0.87	0.99	**0.79**	0.89	0.89	0.99	**0.82**	0.89	0.89
Swin3D	S3DIS	0.96	**0.40**	0.44	0.30	0.98	**0.62**	0.71	0.71	0.98	**0.77**	0.89	0.89	0.99	**0.82**	0.91	0.91
PTv3	Struc.3D	0.98	**0.61**	0.71	0.68	0.99	**0.82**	0.90	0.90	0.99	**0.83**	0.91	0.91	0.99	**0.83**	0.90	0.90
Swin3D	Struc.3D	0.96	**0.33**	0.38	0.24	0.99	**0.71**	0.82	0.82	0.99	**0.83**	0.91	0.91	0.99	**0.85**	0.92	0.92

5.4 Data sensitivity study

The results of the sensitivity study are presented in Table 7 and Figure 11 reporting the performance metrics mean intersection over union, accuracy, and F1-score across all 10 classes. The findings show that the most significant

performance gain from adding scenario-specific training data occurs at the second stage, corresponding to a training set size of 12 samples. The PTv3 model converges faster, but the Swin3D model closes the gap, and at the end of Stage 3, all four combinations are roughly equal. This sensitivity study indicates that adding more, yet similar, data yields diminishing returns. For both models, regardless of their pretraining source, performance converges after 12 to 18 samples, which are sufficient to represent the characteristics of the construction site in the validation dataset.

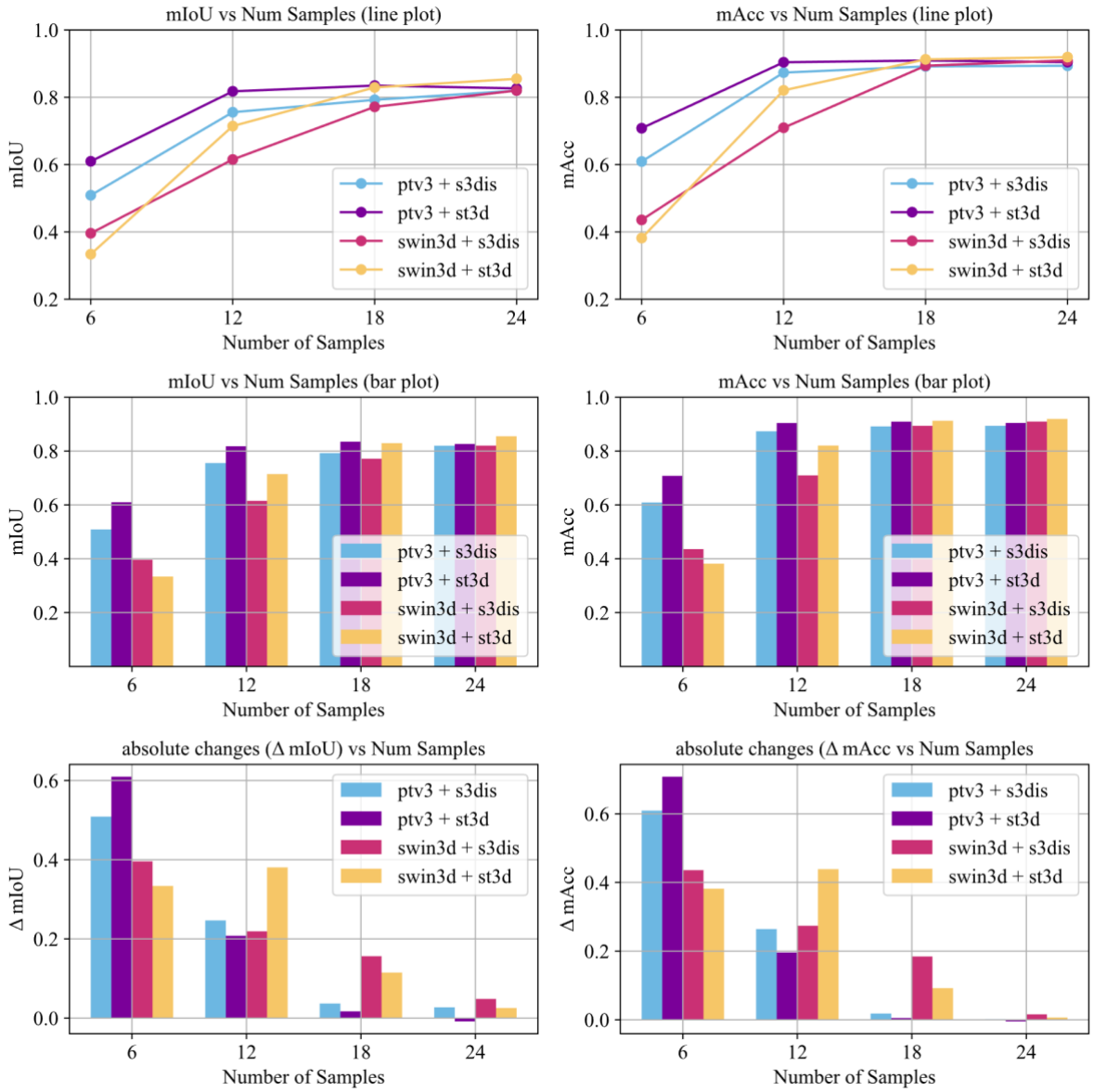

Figure 11: Results of the four-stage data sensitivity study. Model performance is evaluated using mean Intersection over Union (mIoU) and mean Accuracy (mAcc), and the results are plotted as a function of the number of samples used for model fine-tuning. The first row illustrates the gradual performance change as the number of samples increases. The second row shows the performance at each of the four stages. The third row highlights the absolute change (Δ) in performance relative to the previous stage.

The comparison with the results in Table 6 further indicates that the gradual addition of training data has minimal impact on the final model's performance. The overall deviation between the average test results in Table 6 (where the model was trained once on all available data), and those in Table 7 (where data is added incrementally) is

1.35% for mIoU and 0.31% for mAcc. These differences fall within the range of numerical variations observed across individual training runs, even when initialized with identical parameter seeds.

6. DISCUSSION

This study aims to examine how modern transformer models perform under data-scarce conditions and to evaluate their potential for a data-pre-labeling system in the AEC domain. This requires the model to reliably classify at least the major building components, such as walls, ceilings, and floor slabs. Furthermore, the application would benefit greatly if the model could recognize smaller structural parts such as columns and beams. The idea is that if raw data can be adequately pre-segmented, the manual effort required to segment more complex components is greatly reduced. Since object segmentation in point cloud data from site surveys is essential for engineering services—including construction progress monitoring, as-built reconstruction, BIM modeling, quality control, quantity surveying, and billing—and is also a prerequisite for autonomous robots on construction sites, automation in this area is critically needed.

In the current situation, where no publicly available 3D data exists to train and validate AI models for classifying technical building components in a shell-construction environment, advanced computer vision techniques can reduce the amount of training data needed for supervised learning. Our results provide evidence that, even with limited annotated data, high segmentation performance can be achieved for the main components, and that gradual fine-tuning can be an effective tool for pre-segmenting structural elements in LiDAR scans of shell construction sites. We evaluated the model's performance across three phases: a standard supervised training phase with only a few samples, a cross-domain evaluation phase using generic indoor data unrelated to the test domain, and a final transfer-learning stage to fine-tune the top two models through knowledge transfer.

The results indicated that, for naive supervised learning, large transformer models with millions of parameters are difficult to train using a single small batch, even when, as in the current case, the training and test data come from the same construction site. Although two of the latest 3D transformer models (PTv3 and Swin3D) achieved fairly good results on large primitive objects such as walls, ceilings, and floors, they lack the detail needed at object-segment boundaries and for distinguishing objects in close proximity, making them unreliable for production use. Transition areas around windows and door frames, as well as the more complex staircase scene, pose a generalization problem in which models tend to favor the more frequent classes in the dataset over underrepresented ones during classification. At the same time, the previous PTv2 model could not reproduce the high validation scores observed during training, not even for the major classes, indicating a lack of generalizability and overfitting due to insufficient training data volume and variety.

Prior work has reported that transformer-based vision models can be less sample-efficient than simpler architectures when trained from scratch on small datasets. (Y. Liu et al., 2021; X. Ma et al., 2022; Qian et al., 2022) Although a comprehensive comparison of different backbones and the selected transformers was not the aim of this study, we obtained results from some simpler architectures during the initial baseline screening that can help verify this hypothesis. Based on these results, we cannot determine whether simpler models perform better in our limited-data setup. While the scaled-up version of PointNet++ in the purely MLP-based PointNeXt performed similarly to the modern Point Transformer V3 and the Swin3D transformer, there was no clear superiority. Instead, all models were similarly limited by the available data. Because this study requires precise segmentation for the outlined purpose of a stable pre-labeling engine, at least for the frequently occurring structural elements, the intermediate results of the Phase 1 experiment were judged insufficient and motivated more advanced data utilization.

Although no construction site-specific datasets are available for training, several sources provide sample data from building interiors. We found that these samples partially overlap with the classes we require, which justified testing their generalizability. The second phase of our experiments confirmed that training solely on non-specific data is insufficient to develop a reliable component segmentation model that meets the technical standards of AEC disciplines. We trained on four popular indoor datasets (S3DIS, ScannetV2, Structured3D, and VASAD), which cover residential, educational, and office environments with overlapping classes such as walls, ceilings, floors, windows, doors, and beams. However, the segmentation results on our domain-shifted construction site data were inconsistent. The main observation was that even frequently occurring classes, such as walls, which are present in both the source and target datasets, were not reliably classified.

The causes of the domain transfer deficiency can be narrowed down to three main factors. First, there is a gap between synthetic training data and real-world laser-scanning point clouds. All four auxiliary datasets used in this study are based on point sampling from surface meshes and lack key LiDAR characteristics, such as measurement noise, irregular point density, and sensor-induced artifacts. While sampling from digital geometry models provides an ideal representation of objects in point cloud format, measurements from laser sensors often encounter issues such as ghost points at sharp edges, specular-surface artifacts, and measurement errors in chaotic environments. The models are not prepared for this type of irregularity when trained on sampled data.

Second, there is a gap between furnished interior spaces used for training and unfinished shell environments used for testing. The models rely on RGB color as an input feature; however, object appearance changes substantially between the construction and post-occupancy stages, as walls, ceilings, and floors are painted or covered, and furnishings conceal raw-material surfaces. This results in a significant shift in visual cues available to the model. Beyond color, geometric surface features also differ between the two domains. Training data from interior environments and mesh-based reconstructions tend to show smoother, more regular surfaces. In contrast, real-world laser scans from construction sites capture rough, uneven textures caused by unplastered brickwork, unscreeded floor slabs, and pollution from building materials. These differences in both appearance and surface geometry reduce the model's robustness when transferring from finished interiors to active construction sites.

Third, model performance is further affected by semantic inconsistencies in ground-truth annotations across datasets, posing a fundamental challenge to cross-domain generalization. Doors well exemplify this issue, but similar mismatches also occur for other classes, such as beams and slabs in the VASAD dataset and clutter objects in S3DIS. In datasets designed for general indoor scene understanding, such as S3DIS, Structured3D, and ScanNet, a door is typically annotated as the installed door leaf and, in some cases, its frame. In contrast, during construction, doors are not yet installed. Therefore, in the SC dataset, the door class refers to the rough wall opening intended for later installation. This discrepancy in class definitions hampers effective knowledge transfer, as door openings in shell construction data do not correspond to the learned semantic concept of a door. A similar semantic conflict occurs in the VASAD dataset, where floors and ceilings are collectively labeled as a single "slab" class, further obscuring distinctions.

The experiments in Phase 2 clearly showed that these general datasets are insufficient to directly segment components in construction site laser scans. However, transformer architectures, in particular, benefit from extensive pretraining (Akkaya et al., 2024; Yang et al., 2023), which we confirmed through transfer learning using a small amount of domain-specific data. Performance improvements from the transfer learning experiment in Table 6 show that underrepresented classes especially benefit from pretrained general model knowledge, and the impact of the domain gap diminishes when a small amount of scene-specific data is available. The average segmentation performance improved by 10–15% with transfer learning, and class-wise improvements of up to 55% were observed for the underrepresented technical installation class. The supplementary sensitivity analysis showed that just 12 to 18 annotated sample scans are sufficient to calibrate a pretrained model to a specific building type. In practice, this is a negligible effort given the productivity gains from subsequent automation enabled by the fine-tuned model. Point Transformer V3 represents point-wise serial attention models, while Swin3D extends the hierarchical sliding-window attention approach to 3D data and provides an efficient multi-scale voxel representation. In our experiments, Swin3D achieves slightly higher overall performance gains, reaching a total mIoU of 0.84 across all 10 classes when pretrained on the large-scale Structured3D dataset. This does not prove that self-attention on sparse voxels is categorically superior to serialized point cloud attention for component segmentation in our target domain, as the difference falls within the range of statistical uncertainty for small samples, and the experiments were not optimized through hyperparameter tuning to achieve the best possible results in each case. However, hierarchical Swin3D attention seems to provide advantages for small objects with relatively few scan points per instance, as observed in the "installation" and "beam" classes, where the gap between the two models is much larger. In terms of efficiency, a clear trend emerges from the data sensitivity study: the PTv3 model converges to the final potential earlier in both cases with fewer sample scenes.

This series of experiments aimed to analyze transformer behavior and its improvement through fine-tuning with limited data, rather than to establish a performance benchmark. Hyperparameter tuning was deliberately kept minimal, with parameters selected based on published settings from the Pointcept repository (Pointcept Contributors, 2023) for each dataset. Most importantly, the authors maintained consistent hyperparameters across different experimental phases to reliably assess the impact of the data. Data augmentation is essential for small

and imbalanced datasets. Therefore, significant augmentations in the geometric and color feature spaces were central to all experiments, as mentioned in Section 4.6. However, since no additional tests were performed without data augmentation, the ultimate effect of augmentation remains unquantified.

We also acknowledge a performance constraint due to the exclusion of surface normal vectors as input features. Human-made structural components are mostly flat, rectangular geometries, and their segmentation benefits from incorporating surface-topology features. However, surface normals cannot be measured directly by LiDAR scanners, and reconstructing these vectors is often prone to error bias. Still, reconstructed surface vectors may be useful, and the effect of estimated surface normal vectors should be explored in future research. Another opportunity arises from the additional use of reflection intensity, a scalar feature measured directly by the scanner, to help distinguish materials or objects with comparable surface roughness. Since intensity cannot be reliably simulated, it is absent from most synthetic training datasets. Its potential for domain-specific segmentation tasks remains largely unexplored and should be the focus of future studies.

An important methodological limitation concerns the distinction between within-site generalization and true cross-site transferability. Although separate training, validation, and test scenes were used, all SC samples originate from a single residential shell-construction project acquired with the same scanning setup and under comparable construction conditions. Therefore, the reported results demonstrate robustness to intra-project variation, but they do not yet establish generalization to unseen construction sites. This limitation is particularly relevant for transformer-based models, whose high representational capacity can lead to the learning of project-specific geometric, visual, and acquisition-related regularities when only a small number of domain-specific samples is available. The behavior of PTv2 in Phase 1, where strong validation results were not matched by comparable test performance, illustrates the risk of overfitting. In addition, the limited out-of-domain generalization observed in Phase 2 indicates that pretraining on generic indoor datasets alone is insufficient to ensure robust performance under substantial domain shift. The improvements observed in Phase 3 should therefore be interpreted as evidence for site-adaptive transfer learning rather than universal generalization. In practical terms, the models appear promising as pre-labeling tools when a limited amount of annotated target-site data is available, whereas their robustness across different building types, construction stages, sensor configurations, and annotation conventions remains to be validated.

So far, this study indicates that limited training data in the AEC domain can be partially mitigated by transformer-based pretraining followed by fine-tuning, provided that a small set of annotated target-domain scans is available. For now, this was done using a restricted test dataset from a single residential building, which is a limitation to be discussed. At the time of this publication, we were constrained to 36 samples, as acquiring and annotating such data is very time-consuming. Given the very promising results, work is currently underway to expand this dataset on a large scale, including a variety of buildings from the education, commercial, and public sectors (not only new buildings but also renovated structures), as well as a more complete set of component classes. (Rauch & Braml, 2025)

Whether the same transfer learning strategy extends to other building types has not yet been verified and remains an important subject for future evaluation. In this context, it was not possible to evaluate the classification of an object class “column,” even though it is generally a key structural element for modeling the supporting framework in BIM and as-built applications. However, columns are rare in residential buildings and are therefore missing from the previous SC-dataset. This will change with a larger selection of buildings, which also applies to the included but currently underrepresented classes of beams and stair objects. Although stair elements achieved relatively high IoU scores, no definitive conclusions can be drawn about their overall recognition. On the other hand, the example of frequently occurring doors illustrates how transformers quickly learn new object properties across domains.

Finally, the implementation of semantic component segmentation in production-grade workflows for downstream applications remains an open area for study within the AEC community. Some recent publications on scan-to-BIM and 3D reconstruction from point cloud data suggest using semantic segmentation to find the positions and boundary shapes of structural elements and to generate information-rich CAD models. (Chen et al., 2019; Park et al., 2022; Son & Kim, 2017; Tang et al., 2022; Xiang et al., 2023) In general, these reconstruction pipelines use semantic segmentation to filter out basic components, clustering (mainly a density-based clustering algorithm, DBSCAN (Ester et al., 1996)) to separate class segments into individual object instances, and cubic boundary-box fitting to determine object sizes. The actual creation of a data-based 3D model typically involves plane fitting and

boundary-constrained optimization, encoded with expert knowledge to place walls within a floor plan or to model local concave properties, resulting in a volumetric CAD description of structural elements. Since there are no uniform benchmarks for this, and the authors of these studies predominantly evaluate their approaches qualitatively without establishing quantitative metrics, no further attempts were made in this study to assess improvements in 3D reconstruction. Instead, we refer to the statement by Tang et al. (Tang et al., 2022). Their results show that errors in the BIM reconstruction are mainly caused by poor segmentation. These publications also rely heavily on the S3DIS dataset for developing their algorithms, which, as we have demonstrated, causes fundamental issues when applied to real laser scans of construction sites.

Future work should extend the presented approach by incorporating larger, more diverse datasets that span multiple building types and a broader range of semantic component classes. Combining several large-scale datasets for pretraining, as demonstrated in recent Point Prompt Training for 3D representation learning (Wu, Tian, et al., 2023), represents a promising direction for further improving model generalization. In addition, modern few-shot learning strategies, such as Seg-NN (Zhu et al., 2024), offer an effective means of mitigating the data-hungry nature of deep segmentation models when only limited annotated data are available. Finally, integrating these models into other practical construction workflows presents a valuable opportunity, as real-world engineering applications stand to benefit significantly from more robust, transferable segmentation methods and the automation of 3D scene understanding.

7. CONCLUSION

This study presents a comprehensive evaluation of transformer-based deep learning models for 3D semantic segmentation of laser scans in shell construction environments. Using a three-phase experimental setup, we examined the limitations of supervised learning with limited training data, the challenges of cross-domain generalization from synthetic indoor datasets, and the potential of limited-data transfer learning to address data scarcity. In a sensitivity analysis, we further examined the critical data threshold needed to adapt a general-pretrained transformer model for a specific building type. Our results show that, while naive supervised training was inadequate for deep neural networks in semantic segmentation, fine-tuning produces robust segmentation results even with just a few domain-specific samples, achieving mean IoU scores above 0.80. Despite these advancements, the study also highlights fundamental issues with using synthetic datasets for construction-specific tasks, including domain mismatches in labeling conventions, missing classes, and the lack of LiDAR-specific features such as measurement noise, surface normals, and reflection intensity. Future research should focus on integrating these features, expanding the annotated dataset to cover more building types and object classes, and exploring deployment scenarios for pre-labeling workflows on actual construction sites. Overall, this study confirms the feasibility of applying transformer-based limited-data transfer learning for structural component segmentation in laser-scanning data from construction sites and lays the foundation for scalable, semi-automated annotation pipelines in the AEC domain. We also encourage researchers to incorporate transfer learning into their problem-specific methods and to develop small, annotated datasets of their own, because, as demonstrated, it often requires only a few samples to significantly improve results.

DATA AVAILABILITY

The data underlying this study, including the annotated point cloud samples used for training, validation, and testing, are available from the corresponding author upon reasonable request. The material will be provided under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits use, sharing, adaptation, distribution, and reproduction in any medium or format, provided appropriate credit is given to the original authors and source, a link to the license is included, and any modifications are indicated.

REFERENCES

Akkaya, I. B., Kathiresan, S. S., Arani, E., & Zonooz, B. (2024). Enhancing performance of vision transformers on small datasets through local inductive bias incorporation. Pattern Recognition, 153, 110510. https://doi.org/10.1016/j.patcog.2024.110510

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016a). 3D Semantic Parsing of Large-Scale Indoor Spaces. 1534–1543. https://doi.org/10.1109/cvpr.2016.170

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016b). 3D Semantic Parsing of Large-Scale Indoor Spaces. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1534–1543. https://doi.org/10.1109/CVPR.2016.170

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016c). 3D Semantic Parsing of Large-Scale Indoor Spaces. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1534–1543. https://doi.org/10.1109/CVPR.2016.170

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., & Salzmann, M. (2013). Unsupervised Domain Adaptation by Domain Invariant Projection. 2013 IEEE International Conference on Computer Vision, 769–776. https://doi.org/10.1109/ICCV.2013.100

Bao, H., Dong, L., Piao, S., & Wei, F. (2021). BEiT: BERT Pre-Training of Image Transformers. https://doi.org/10.48550/ARXIV.2106.08254

Booij, T. M., Chiscop, I., Meeuwissen, E., Moustafa, N., & Hartog, F. T. H. den. (2022). ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Data Sets. IEEE Internet of Things Journal, 9(1), 485–496. https://doi.org/10.1109/JIOT.2021.3085194

Buckland, M., & Gey, F. (1994). The relationship between Recall and Precision. Journal of the American Society for Information Science, 45(1), 12–19. https://doi.org/10.1002/(SICI)1097-4571(199401)45:1%3C12::AID-ASI2%3E3.0.CO;2-L

Cao, Y., & Scaioni, M. (2021). LABEL-EFFICIENT DEEP LEARNING-BASED SEMANTIC SEGMENTATION OF BUILDING POINT CLOUDS AT LOD3 LEVEL. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2021, 449–456. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-449-2021

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2005.12872

Charles, R. Q., Su, H., Kaichun, M., & Guibas, L. J. (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 77–85. https://doi.org/10.1109/CVPR.2017.16

Chen, J., Kira, Z., & Cho, Y. K. (2019). Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction. Journal of Computing in Civil Engineering, 33(4). https://doi.org/10.1061/(asce)cp.1943-5487.0000842

Choy, C., Gwak, J., & Savarese, S. (2019). 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3070–3079. https://doi.org/10.1109/cvpr.2019.00319

Croce, V., Bevilacqua, M. G., Caroti, G., & Piemonte, A. (2021). CONNECTING GEOMETRY AND SEMANTICS VIA ARTIFICIAL INTELLIGENCE: FROM 3D CLASSIFICATION OF HERITAGE DATA TO H-BIM REPRESENTATIONS. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2021, 145–152. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-145-2021

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Niessner, M. (2017). ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2432–2443. https://doi.org/10.1109/CVPR.2017.261

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/ARXIV.1810.04805

Dimitrov, A., & Golparvar-Fard, M. (2015). Segmentation of building point cloud models including detailed architectural/structural features and MEP systems. Automation in Construction, 51, 32–45. https://doi.org/10.1016/j.autcon.2014.12.015

Dong, R., Qi, Z., Zhang, L., Zhang, J., Sun, J., Ge, Z., Yi, L., & Ma, K. (2023). Autoencoders as Cross-Modal Teachers: Can Pretrained 2D Image Transformers Help 3D Representation Learning? (arXiv:2212.08320). arXiv. https://doi.org/10.48550/arXiv.2212.08320

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. https://doi.org/10.48550/ARXIV.2010.11929

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, 226–231.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., & Hu, S.-M. (2021). PCT: Point cloud transformer. Computational Visual Media, 7(2), 187–199. https://doi.org/10.1007/s41095-021-0229-5

Guo, Y., Li, Y., Ren, D., Zhang, X., Li, J., Pu, L., Ma, C., Zhan, X., Guo, J., Wei, M., Zhang, Y., Yu, P., Yang, S., Ji, D., Ye, H., Sun, H., Liu, Y., Chen, Y., Zhu, J., & Liu, H. (2024). LiDAR-Net: A Real-Scanned 3D Point Cloud Dataset for Indoor Scenes. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 21989–21999. https://doi.org/10.1109/CVPR52733.2024.02076

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2021). Deep Learning for 3D Point Clouds: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(12), 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. arXiv. https://doi.org/10.48550/ARXIV.1703.06870

Hou, J., Graham, B., Niesner, M., & Xie, S. (2021). Exploring Data-Efficient 3D Scene Understanding with Contrastive Scene Contexts. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15582–15592. https://doi.org/10.1109/CVPR46437.2021.01533

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2020). RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds (arXiv:1911.11236). arXiv. https://doi.org/10.48550/arXiv.1911.11236

Hua, B.-S., Pham, Q.-H., Nguyen, D. T., Tran, M.-K., Yu, L.-F., & Yeung, S.-K. (2016). SceneNN: A Scene Meshes Dataset with aNNotations. 2016 Fourth International Conference on 3D Vision (3DV), 92–101. https://doi.org/10.1109/3DV.2016.18

Huang, X., Huang, Z., Li, S., Qu, W., He, T., Hou, Y., Zuo, Y., & Ouyang, W. (2024). Frozen CLIP Transformer Is an Efficient Point Cloud Encoder. Proceedings of the AAAI Conference on Artificial Intelligence, 38(3), 2382–2390. https://doi.org/10.1609/aaai.v38i3.28013

Jiang, Z., & Messner, J. I. (2023). Computer Vision Applications In Construction And Asset Management Phases: A Literature Review. Journal of Information Technology in Construction, 28, 176–199. https://doi.org/10.36680/j.itcon.2023.009

Kim, D., Tsai, Y.-H., Suh, Y., Faraki, M., Garg, S., Chandraker, M., & Han, B. (2022). Learning Semantic Segmentation from Multiple Datasets with Label Shifts. https://doi.org/10.48550/ARXIV.2202.14030

Kim, T., Cho, W., Matono, A., & Kim, K.-S. (2020). PinSout: Automatic 3D Indoor Space Construction from Point Clouds with Deep Learning. Proceedings of the 28th International Conference on Advances in Geographic Information Systems, 211–214. https://doi.org/10.1145/3397536.3422343

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, 1, 1097–1105.

Lahoud, J., Cao, J., Khan, F. S., Cholakkal, H., Anwer, R. M., Khan, S., & Yang, M.-H. (2022). 3D Vision with Transformers: A Survey (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2208.04309

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., & Jia, J. (2022). Stratified Transformer for 3D Point Cloud Segmentation. https://doi.org/10.48550/ARXIV.2203.14508

Langlois, P.-A., Xiao, Y., Boulch, A., & Marlet, R. (2022). VASAD: A Volume and Semantic dataset for Building Reconstruction from Point Clouds. 2022 26th International Conference on Pattern Recognition (ICPR), 4008–4015. https://doi.org/10.1109/ICPR56361.2022.9956356

Layeghy, S., & Portmann, M. (2023). Explainable Cross-domain Evaluation of ML-based Network Intrusion Detection Systems. Computers and Electrical Engineering, 108, 108692. https://doi.org/10.1016/j.compeleceng.2023.108692

Li, G., Müller, M., Thabet, A., & Ghanem, B. (2019). DeepGCNs: Can GCNs Go as Deep as CNNs? (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1904.03751

Liang, H., Yeoh, J. K. W., & Chua, D. K. H. (2024). Material augmented semantic segmentation of point clouds for building elements. Computer-Aided Civil and Infrastructure Engineering, 39(15), 2312–2329. https://doi.org/10.1111/mice.13198

Liang, Z., Li, Z., Xu, S., Tan, M., & Jia, K. (2021). Instance Segmentation in 3D Scenes using Semantic Superpoint Tree Networks. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2763–2772. https://doi.org/10.1109/ICCV48922.2021.00278

Liu, Y., Sangineto, E., Bi, W., Sebe, N., Lepri, B., & De Nadai, M. (2021). Efficient Training of Visual Transformers with Small Datasets. https://doi.org/10.48550/ARXIV.2106.03746

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. https://doi.org/10.48550/ARXIV.2103.14030

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

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

Ma, X., Qin, C., You, H., Ran, H., & Fu, Y. (2022). Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework (arXiv:2202.07123). arXiv. https://doi.org/10.48550/arXiv.2202.07123

Malinverni, E. S., Pierdicca, R., Paolanti, M., Martini, M., Morbidoni, C., Matrone, F., & Lingua, A. (2019). DEEP LEARNING FOR SEMANTIC SEGMENTATION OF 3D POINT CLOUD. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2-W15, 735–742. 27th CIPA International Symposium “Documenting the past for a better future” (Volume XLII-2/W15) - 1–5 September 2019, Ávila, Spain. https://doi.org/10.5194/isprs-archives-XLII-2-W15-735-2019

Mehranfar, M., Braun, A., & Borrmann, A. (2023). Automatic creation of digital building twins with rich semantics from dense RGB point clouds through semantic segmentation and model fitting. https://mediatum.ub.tum.de/doc/1712296/8amdrixpmn8z7yeaab5ha87q3.2023_Mehranfar_EG-ICE.pdf

Noichl, F., Collins, F. C., Braun, A., & Borrmann, A. (2024). Enhancing point cloud semantic segmentation in the data-scarce domain of industrial plants through synthetic data. Computer-Aided Civil and Infrastructure Engineering, mice.13153. https://doi.org/10.1111/mice.13153

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

paperswithcode. (2024). 3D Semantic Segmentation [Computer software]. https://paperswithcode.com/task/3d-semantic-segmentation

Park, J., Kim, J., Lee, D., Jeong, K., Lee, J., Kim, H., & Hong, T. (2022). Deep Learning–Based Automation of Scan-to-BIM with Modeling Objects from Occluded Point Clouds. Journal of Management in Engineering, 38(4), 04022025. https://doi.org/10.1061/(ASCE)ME.1943-5479.0001055

Park, J., Zhou, Q.-Y., & Koltun, V. (2017). Colored Point Cloud Registration Revisited. IEEE International Conference on Computer Vision, 143–152. https://doi.org/10.1109/iccv.2017.25

Parnami, A., & Lee, M. (2022). Learning from Few Examples: A Summary of Approaches to Few-Shot Learning (arXiv:2203.04291). arXiv. https://doi.org/10.48550/arXiv.2203.04291

Perez-Perez, Y., Golparvar-Fard, M., & El-Rayes, K. (2021). Segmentation of point clouds via joint semantic and geometric features for 3D modeling of the built environment. Automation in Construction, 125, 103584. https://doi.org/10.1016/j.autcon.2021.103584

Pointcept Contributors. (2023). Pointcept: A Codebase for Point Cloud Perception Research [Computer software]. https://github.com/Pointcept/Pointcept

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1706.02413

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H. A. A. K., Elhoseiny, M., & Ghanem, B. (2022). PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2206.04670

Rauch, L., & Braml, T. (2023). Semantic Point Cloud Segmentation with Deep-Learning-Based Approaches for the Construction Industry: A Survey. Applied Sciences, 13(16), 9146. https://doi.org/10.3390/app13169146

Rauch, L., & Braml, T. (2025). Rohbau3D: A Shell Construction Site 3D Point Cloud Dataset. Scientific Data, 12(1), 1478. https://doi.org/10.1038/s41597-025-05827-7

Reja, V. K., Varghese, K., & Ha, Q. P. (2022). Computer vision-based construction progress monitoring. Automation in Construction, 138, 104245. https://doi.org/10.1016/j.autcon.2022.104245

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (Vol. 9351, pp. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., & Leibe, B. (2022). Mask3D: Mask Transformer for 3D Semantic Instance Segmentation (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2210.03105

Son, H., & Kim, C. (2017). Semantic as-built 3D modeling of structural elements of buildings based on local concavity and convexity. Advanced Engineering Informatics, 34, 114–124. https://doi.org/10.1016/j.aei.2017.10.001

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Tang, S., Li, X., Zheng, X., Wu, B., Wang, W., & Zhang, Y. (2022). BIM generation from 3D point clouds by combining 3D deep learning and improved morphological approach. Automation in Construction, 141, 104422. https://doi.org/10.1016/j.autcon.2022.104422

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). KPConv: Flexible and Deformable Convolution for Point Clouds (arXiv:1904.08889). arXiv. https://doi.org/10.48550/arXiv.1904.08889

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need (Version 7). arXiv. https://doi.org/10.48550/ARXIV.1706.03762

Wang, L., Li, D., Liu, H., Peng, J., Tian, L., & Shan, Y. (2021). Cross-Dataset Collaborative Learning for Semantic Segmentation in Autonomous Driving. https://doi.org/10.48550/ARXIV.2103.11351

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic Graph CNN for Learning on Point Clouds (arXiv:1801.07829). arXiv. http://arxiv.org/abs/1801.07829

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., & Zhao, H. (2023, December 15). Point Transformer V3: Simpler, Faster, Stronger. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2024. https://doi.org/10.48550/arXiv.2312.10035

Wu, X., Lao, Y., Jiang, L., Liu, X., & Zhao, H. (2022, October 12). Point Transformer V2: Grouped Vector Attention and Partition-based Pooling. Conference on Neural Information Processing Systems (NeurIPS) 2022. https://doi.org/10.48550/arXiv.2210.05666

Wu, X., Tian, Z., Wen, X., Peng, B., Liu, X., Yu, K., & Zhao, H. (2023). Towards Large-scale 3D Representation Learning with Multi-dataset Point Prompt Training. https://doi.org/10.48550/ARXIV.2308.09718

Wu, X., Wen, X., Liu, X., & Zhao, H. (2023). Masked Scene Contrast: A Scalable Framework for Unsupervised 3D Representation Learning. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9415–9424. https://doi.org/10.1109/CVPR52729.2023.00908

Xiang, Z., Rashidi, A., & Ou, G. (2023). Integrating Inverse Photogrammetry and a Deep Learning–Based Point Cloud Segmentation Approach for Automated Generation of BIM Models. Journal of Construction Engineering and Management, 149(9), 04023074. https://doi.org/10.1061/JCEMD4.COENG-13020

Xie, S., Gu, J., Guo, D., Qi, C. R., Guibas, L., & Litany, O. (2020). PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), Computer Vision – ECCV 2020 (Vol. 12348, pp. 574–591). Springer International Publishing. https://doi.org/10.1007/978-3-030-58580-8_34

Yang, Y.-Q., Guo, Y.-X., Xiong, J.-Y., Liu, Y., Pan, H., Wang, P.-S., Tong, X., & Guo, B. (2023). Swin3D: A Pretrained Transformer Backbone for 3D Indoor Scene Understanding (arXiv:2304.06906). arXiv. https://doi.org/10.48550/arXiv.2304.06906

Yin, C., Wang, B., Vincent J.L. Gan, Gan, V. J. L., Wang, M., & Cheng, J. C. P. (2021). Automated semantic segmentation of industrial point clouds using ResPointNet. Automation in Construction, 130, 103874. https://doi.org/10.1016/j.autcon.2021.103874

Zepp, M. (2023). Hybrid semantic clustering of 3D point clouds in construction (p. 388 KB) [Application/pdf]. Ruhr-Universität Bochum. https://doi.org/10.13154/294-10133

Zhang, H. X., & Zou, Z. (2023). Quality assurance for building components through point cloud segmentation leveraging synthetic data. Automation in Construction, 155, 105045. https://doi.org/10.1016/j.autcon.2023.105045

Zhang, Z., Girdhar, R., Joulin, A., & Misra, I. (2021). Self-Supervised Pretraining of 3D Features on any Point-Cloud. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 10232–10243. https://doi.org/10.1109/ICCV48922.2021.01009

Zhao, H., Jiang, L., Jia, J., Torr, P., & Koltun, V. (2021, September 26). Point Transformer. IEEE International Conference on Computer Vision (ICCV) 2021. https://doi.org/10.48550/arXiv.2012.09164

Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., & Zhou, Z. (2020a). Structured3D: A Large Photo-Realistic Dataset for Structured 3D Modeling. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), Computer Vision – ECCV 2020 (Vol. 12354, pp. 519–535). Springer International Publishing. https://doi.org/10.1007/978-3-030-58545-7%5C_30

Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., & Zhou, Z. (2020b). Structured3D: A Large Photo-Realistic Dataset for Structured 3D Modeling. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), Computer Vision – ECCV 2020 (Vol. 12354, pp. 519–535). Springer International Publishing. https://doi.org/10.1007/978-3-030-58545-7%5C_30

Zhu, X., Zhang, R., He, B., Guo, Z., Liu, J., Xiao, H., Fu, C., Dong, H., & Gao, P. (2024). No Time to Train: Empowering Non-Parametric Networks for Few-Shot 3D Scene Segmentation. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3838–3847. https://doi.org/10.1109/CVPR52733.2024.00368

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE, 109(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

APPENDIX A: BASELINE SCREENING STUDY

Table 8: Baseline test results for 3D semantic segmentation. All models are trained and evaluated on the SC-dataset. Reported performance metrics include class overall accuracy (OA), mean Precision (mPrec), mean Accuracy (mAcc, also called mean Recall), mean F1-score (mF1), and mean Intersection over Union (mIoU). The best results per metric are highlighted. (*) This indicates the test results obtained by dividing the test point clouds into smaller sections to prevent GPU memory overflow. The fragments were sampled with region overlap, and the final predictions were selected using maximum class voting.

Model Architecture	OA	mAcc	mIoU	mPrec	mF1
PointNet (Charles et al., 2017)	0.37	0.16	0.08	0.10	0.08
PointNet++ (Qi et al., 2017)	0.94	0.43	0.39	0.51	0.36
DeepGCN (*) (Li et al., 2019)	0.65	0.24	0.19	0.34	0.19
DGCNN (*) (Wang et al., 2019)	0.65	0.23	0.19	0.15	0.17
PointNeXt (s) (Qian et al., 2022)	0.95	0.52	0.44	0.53	0.44
PointNeXt (b) (Qian et al., 2022)	0.97	0.64	0.59	0.60	0.58
PointNeXt (l) (Qian et al., 2022)	0.98	0.76	0.69	0.76	0.73
PointNeXt (xl) (Qian et al., 2022)	0.98	0.77	0.69	0.83	0.75
Point Transformer V2 (Wu et al., 2022)	0.75	0.45	0.32	0.76	0.40
Point Transformer V3 (Wu et al., 2023)	0.97	0.84	0.72	0.88	0.84
Swin3D (Yang et al., 2023)	0.97	0.79	0.66	0.85	0.79

APPENDIX B: CROSS-DOMAIN CLASS MAPPING

Table 9: Class encodings for cross-domain experiments. The table lists the mappings from the four source-domain datasets (S3DIS, ScanNetV2, Structured3D, and VASAD) to the target classes of the construction-specific SC-dataset. Labels that cannot be mapped to one of the 10 target classes are assigned to the auxiliary category no_class.

S3DIS				**ScannetV2**			
source		: target		source		: target	
ID	**class name**		**ID** **class name**	**ID**	**class name**		**ID** **class name**
0	ceiling	:	1 ceiling	0	wall	:	3 wall
1	floor	:	2 floor	1	floor	:	2 floor
2	wall	:	3 wall	2	cabinet	:	-1 no_class
3	beam	:	4 beam	3	bed	:	-1 no_class
4	column	:	-1 no_class	4	chair	:	-1 no_class
5	window	:	5 window	5	sofa	:	-1 no_class
6	door	:	6 door	6	table	:	-1 no_class
7	table	:	-1 no_class	7	door	:	6 door
8	chair	:	-1 no_class	8	window	:	5 window
9	sofa	:	-1 no_class	9	bookshelf	:	-1 no_class
10	bookcase	:	-1 no_class	10	picture	:	-1 no_class
11	board	:	-1 no_class	11	counter	:	-1 no_class
12	clutter	:	0 none	12	desk	:	-1 no_class
				13	curtain	:	-1 no_class
				14	refridgerator	:	-1 no_class
				15	shower curtain	:	-1 no_class
				16	toilet	:	-1 no_class
				17	sink	:	-1 no_class
				18	bathtub	:	-1 no_class
				19	_otherfurniture	:	-1 no_class

Structured3D				**VASAD**			
source		: target		source		: target	
ID	**class name**		**ID** **class name**	**ID**	**class name**		**ID** **class name**
0	wall	:	3 wall	0	beam	:	4 beam
1	floor	:	2 floor	1	bearingwall	:	3 wall
2	cabinet	:	-1 no_class	2	ceiling	:	1 ceiling
3	bed	:	-1 no_class	3	door	:	6 door
4	chair	:	-1 no_class	4	slab	:	2 floor
5	sofa	:	-1 no_class	5	partition wall	:	3 wall
6	table	:	-1 no_class	6	pillar	:	-1 no_class
7	door	:	6 door	7	railing	:	-1 no_class
8	window	:	5 window	8	roof	:	-1 no_class
9	picture	:	-1 no_class	9	stair	:	7 stairs
10	desk	:	-1 no_class	10	window	:	5 window
11	shelves	:	-1 no_class				
12	curtain	:	-1 no_class				
13	dresser	:	-1 no_class				
14	pillow	:	-1 no_class				
15	mirror	:	-1 no_class				
16	ceiling	:	1 ceiling				
17	refrigerator	:	-1 no_class				
18	television	:	-1 no_class				
19	nightstand	:	-1 no_class				
20	sink	:	-1 no_class				
21	lamp	:	-1 no_class				
22	otherstructure	:	-1 no_class				
23	_otherfurniture	:	-1 no_class				
24	_otherprop	:	-1 no_class				

APPENDIX C: QUALITATIVE RESULTS – BASELINE LEARNING

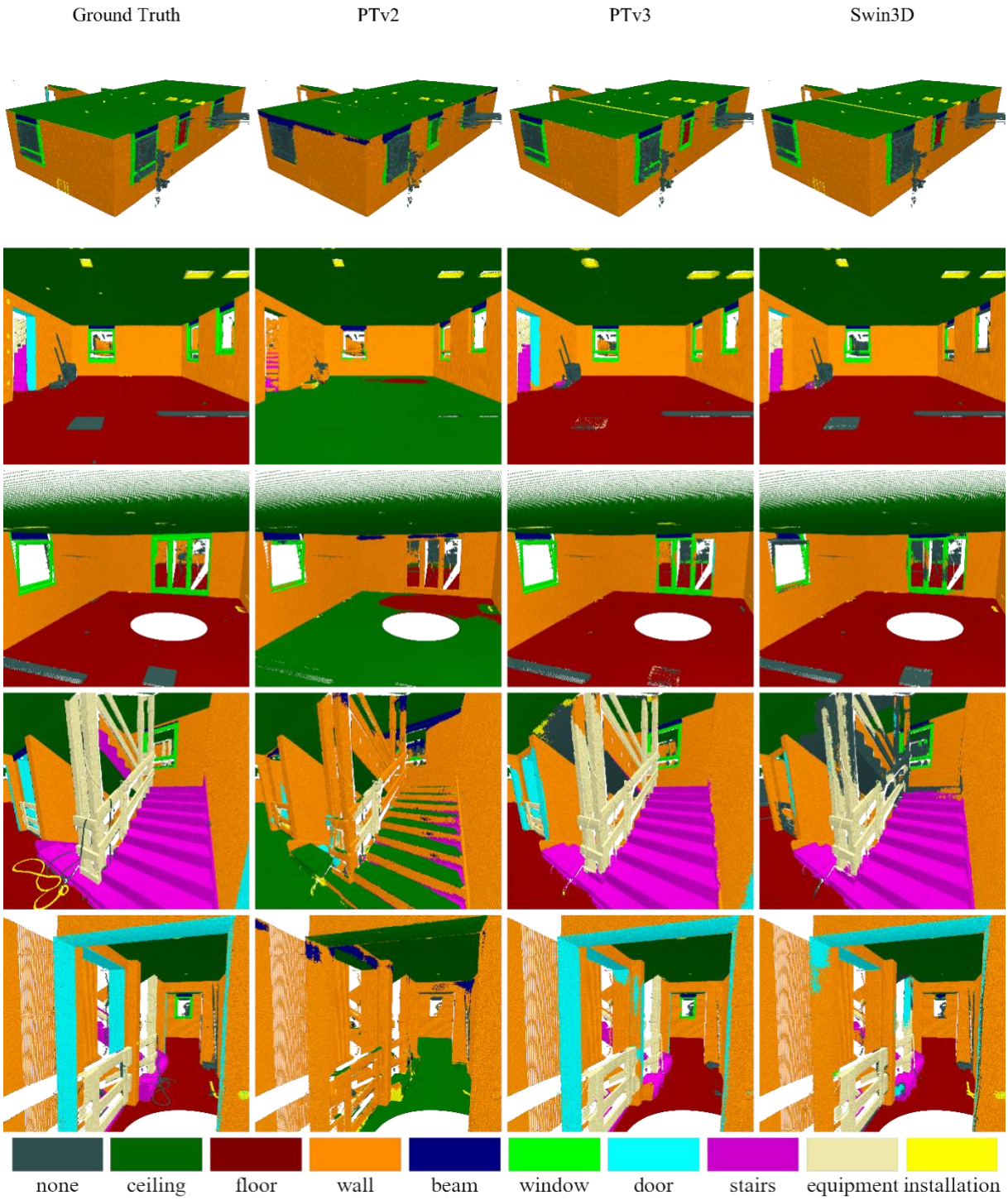

Figure 12: Baseline test results for 3D semantic segmentation. This figure shows inference results from the baseline training experiment using three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. Each model is trained and evaluated on a custom validation dataset focused on shell construction site scenes. Each column presents five representative predictions per model, with class labels in distinct colors illustrating segmentation performance across various architectural components.

APPENDIX D: QUALITATIVE RESULTS – CROSS-DOMAIN LEARNING

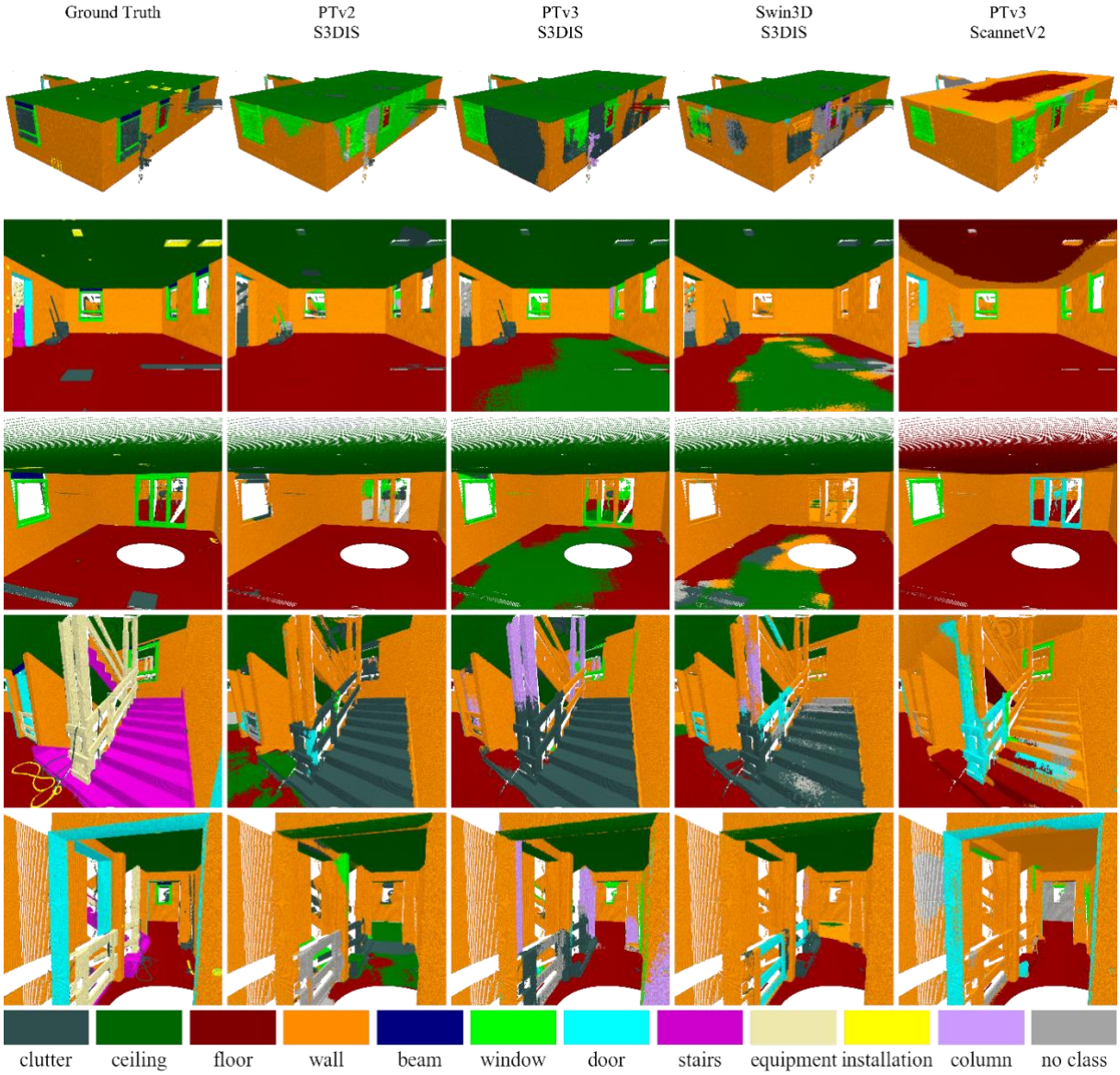

Figure 13: Transfer learning results for 3D semantic segmentation (Part 1). This figure shows inference results from the transfer learning experiment using three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. Each model is trained on one of two source datasets (S3DIS or ScannetV2) and evaluated on a custom validation dataset focused on shell construction site scenes. Each column presents five representative predictions per model, with class labels in distinct colors illustrating segmentation performance across various architectural components. Labels that cannot be mapped to one of the 10 target classes, as defined in Table 8, are assigned to the auxiliary category no_class.

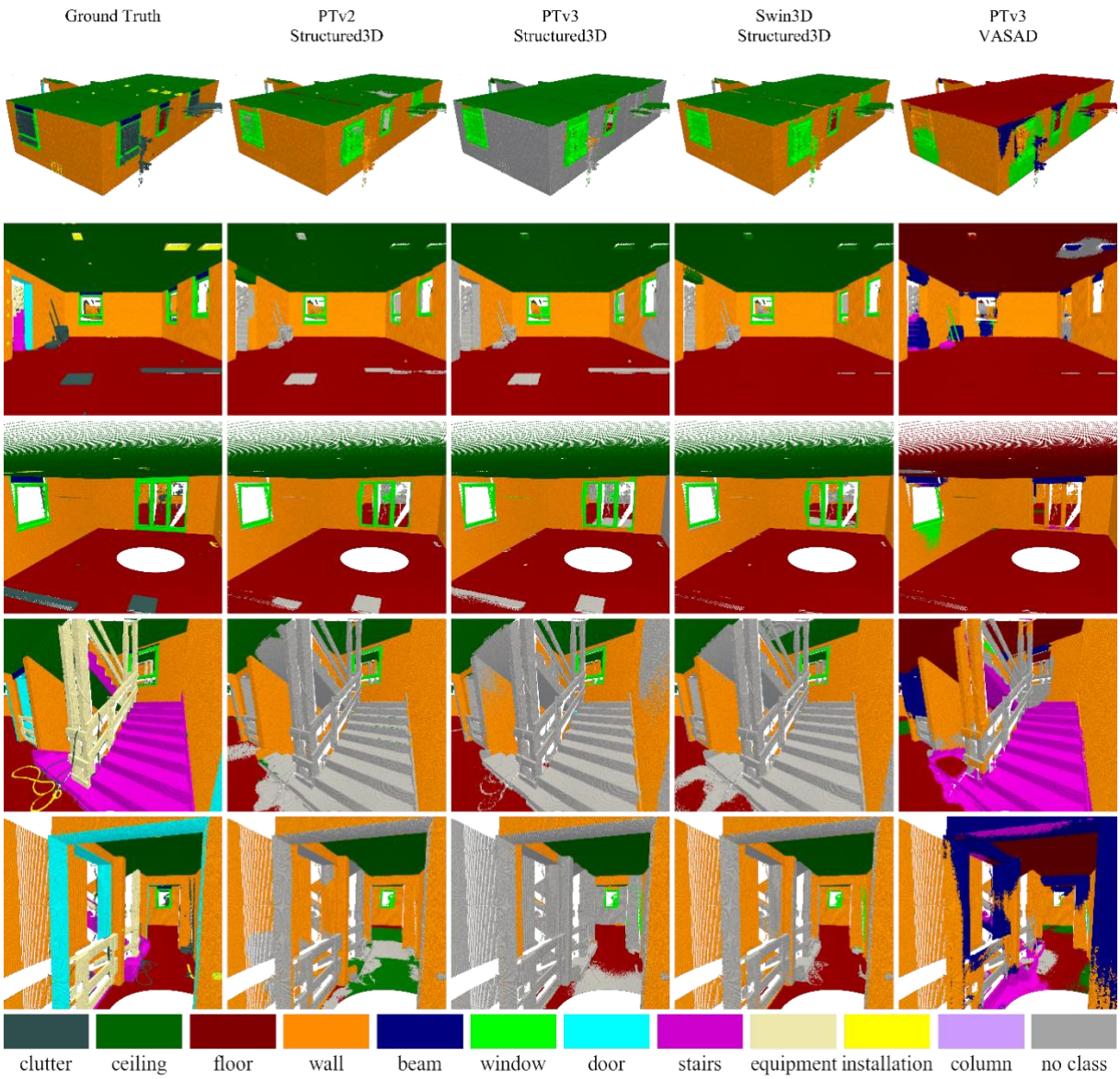

Figure 14: Transfer learning results for 3D semantic segmentation (Part 2). The figure shows inference results from the transfer learning experiment using three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. Each model is trained on one of two source datasets (Structured3D or VASAD) and evaluated on a custom validation dataset focused on shell construction site scenes. Each column displays five representative predictions per model, with class labels in distinct colors illustrating segmentation performance across various architectural components. Labels that cannot be mapped to one of the 10 target classes, as defined in Table 8, are assigned to the auxiliary category no_class.

APPENDIX E: QUALITATIVE RESULTS – TRANSFER LEARNING

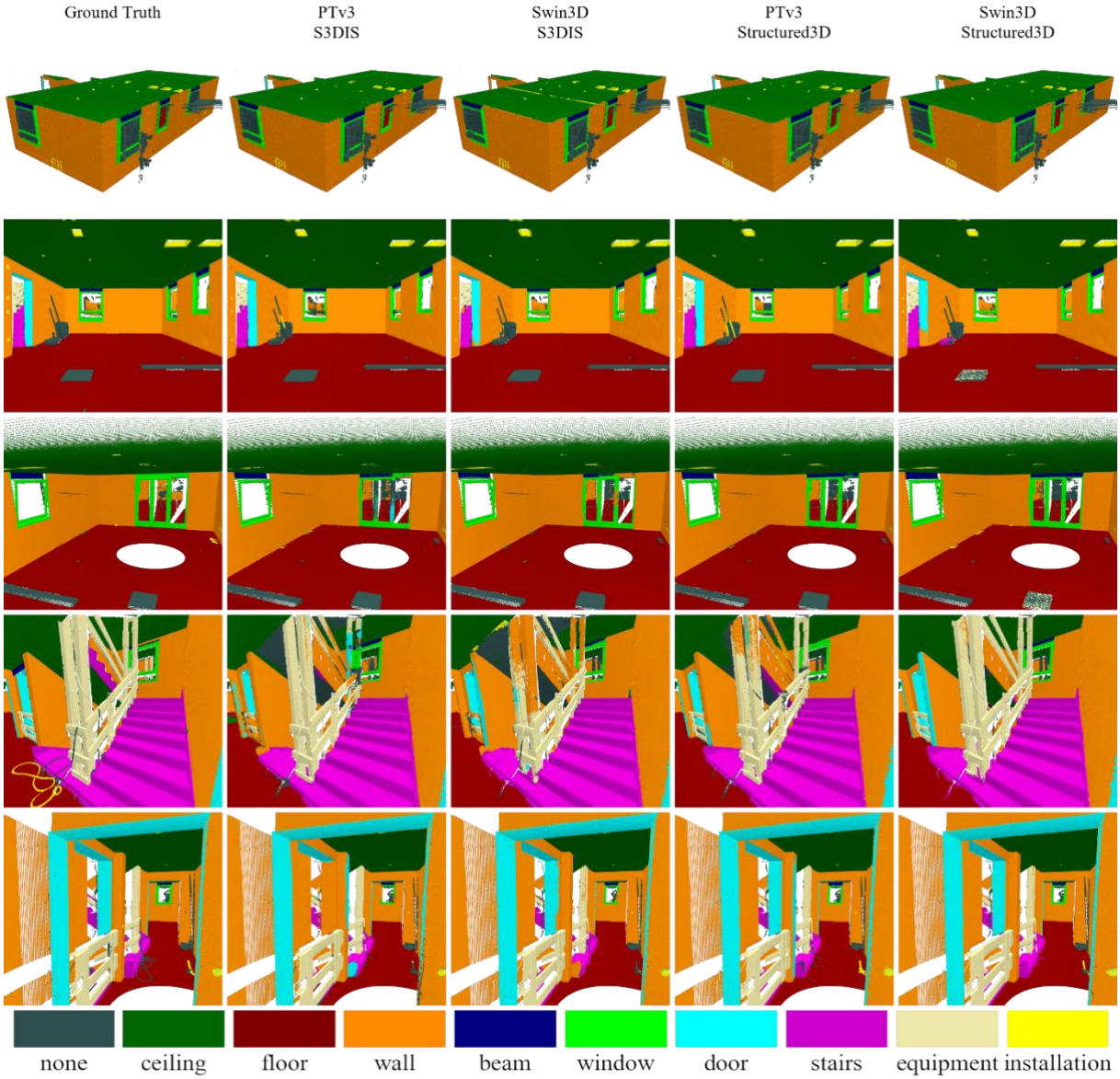

Figure 15: Transfer learning results for 3D semantic segmentation. The figure shows inference outputs from the transfer learning experiment using two model architectures: Point Transformer V3 (PTv3) and Swin3D. Each model is pretrained on one of two datasets (S3DIS or Structured3D) and then fine-tuned and evaluated on a custom validation dataset focused on shell construction site scenes. Each column presents five representative predictions per model, with class labels in distinct colors illustrating segmentation performance across different architectural components.

APPENDIX F: CONFUSION MATRIX – BASELINE TRAINING

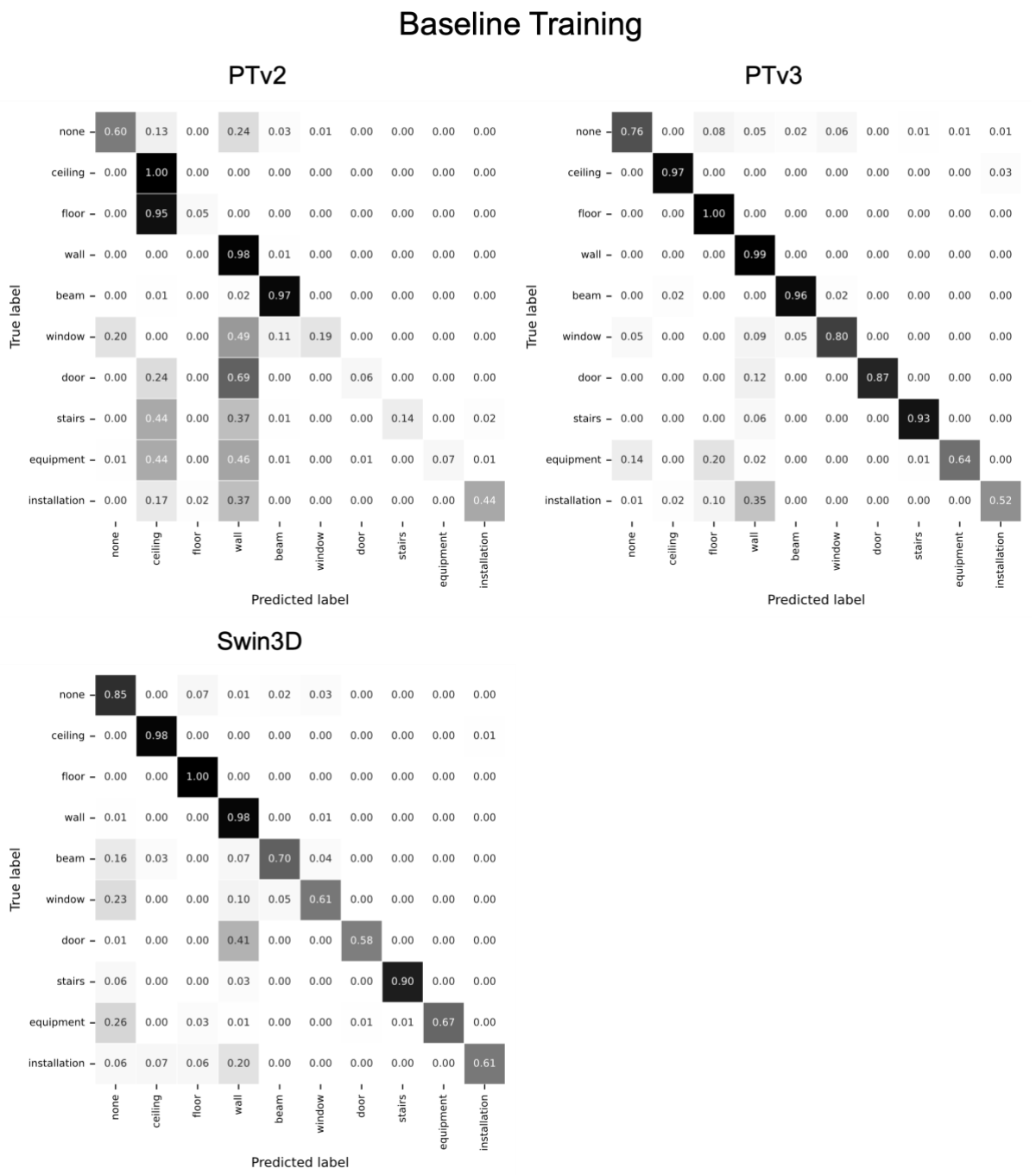

Figure 16: Confusion matrices from the baseline training experiments for 3D semantic segmentation of 10 building component classes in LiDAR point clouds from shell construction sites. Results are shown for three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. All models were trained and evaluated on a custom validation dataset focused specifically on shell construction site scenes.

APPENDIX G: CONFUSION MATRIX – CROSS-DOMAIN EVALUATION

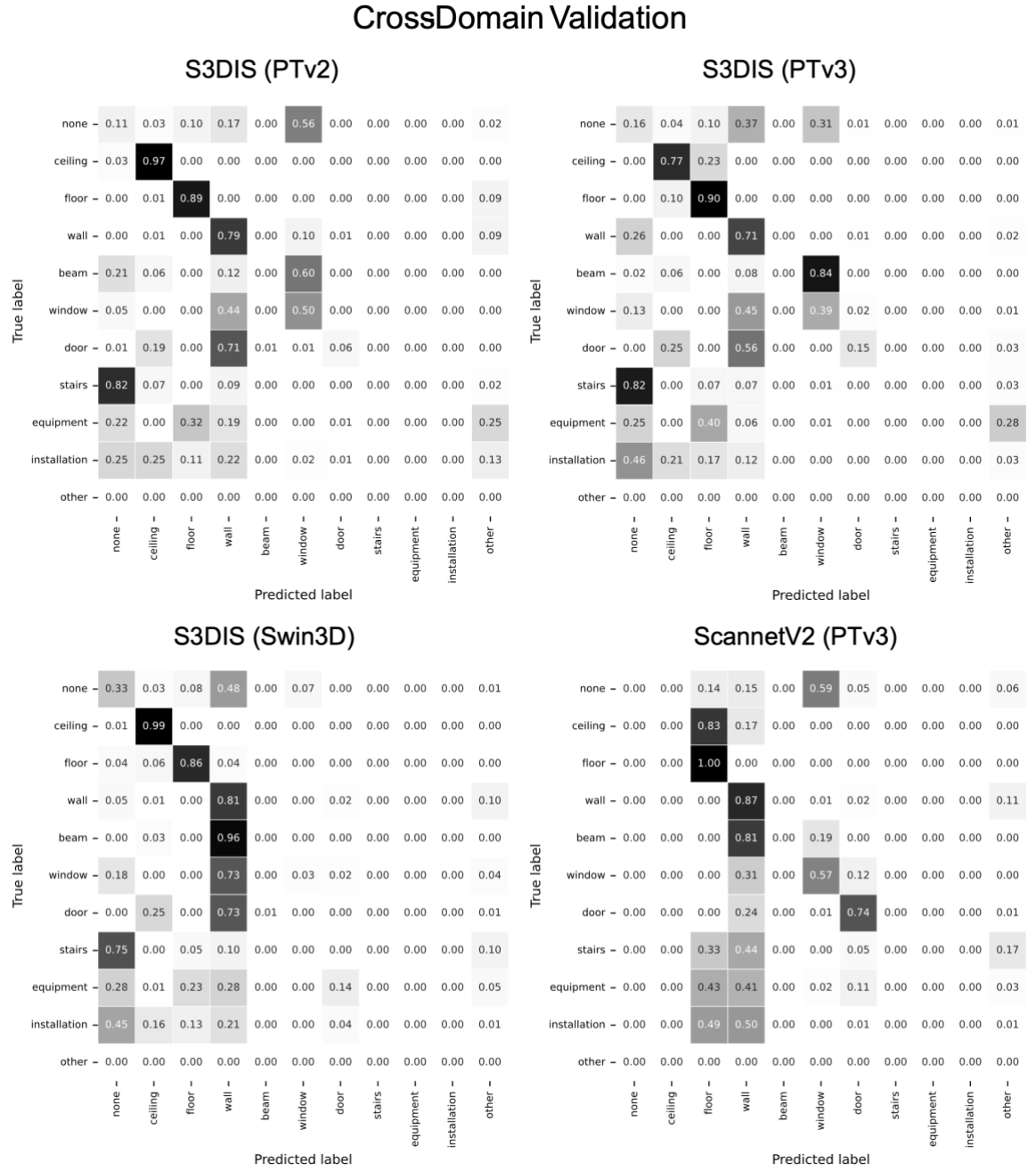

Figure 17 Part 1: Confusion matrices from the cross-domain training experiments for 3D semantic segmentation of 10 building component classes in LiDAR point clouds from shell construction sites. Results are shown for three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. All models were trained on one of two source datasets (S3DIS or ScannetV2) and evaluated on a custom validation dataset focused on shell construction site scenes.

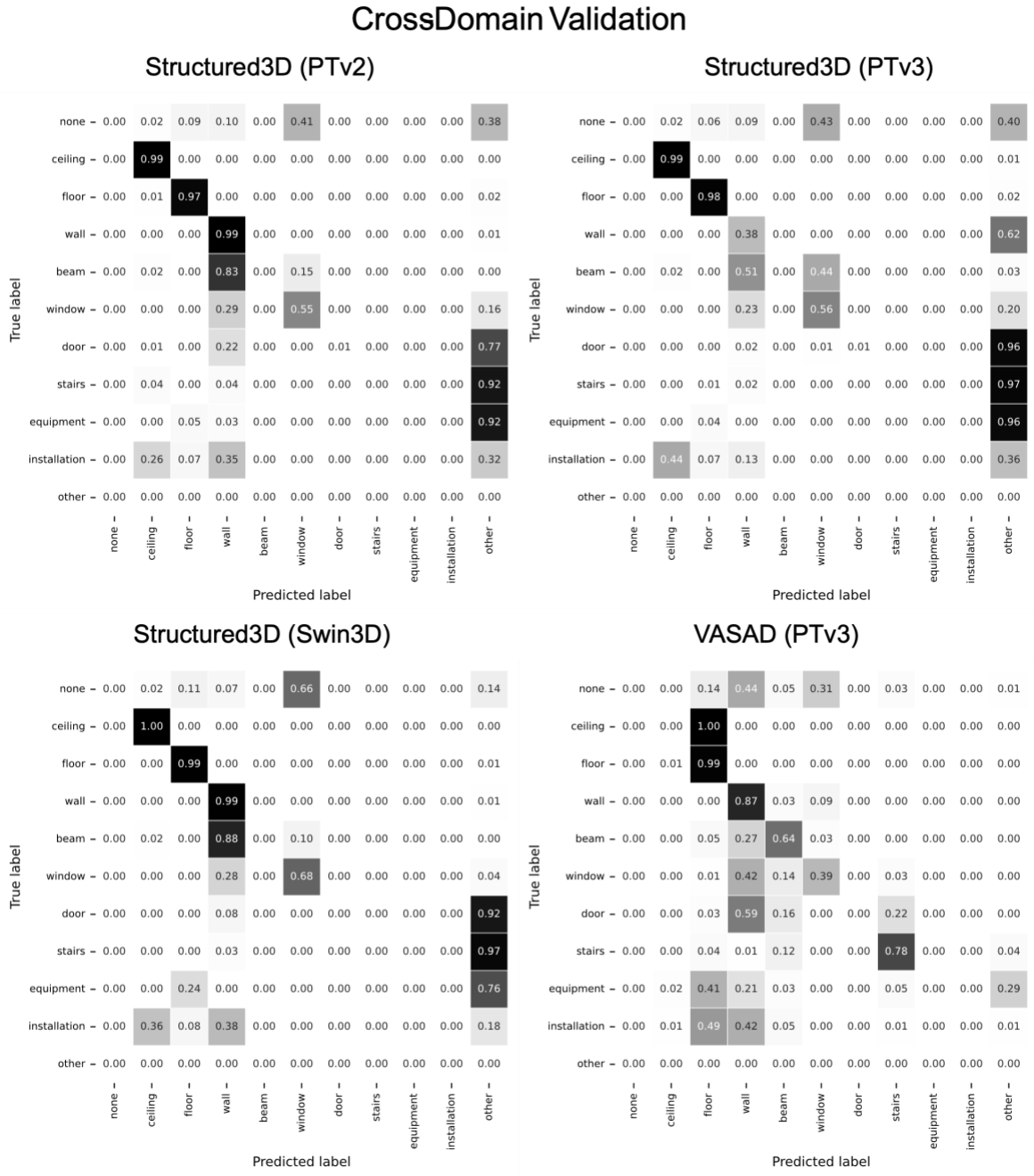

Figure 18 Part 2: Confusion matrices from the cross-domain training experiments for 3D semantic segmentation of 10 building component classes in LiDAR point clouds from shell construction sites. Results are shown for three model architectures: Point Transformer V2 (PTv2), Point Transformer V3 (PTv3), and Swin3D. All models were trained on one of two source datasets (Structured3D or VASAD) and evaluated on a custom validation dataset focused on shell construction site scenes.

APPENDIX H: CONFUSION MATRIX – TRANSFER LEARNING

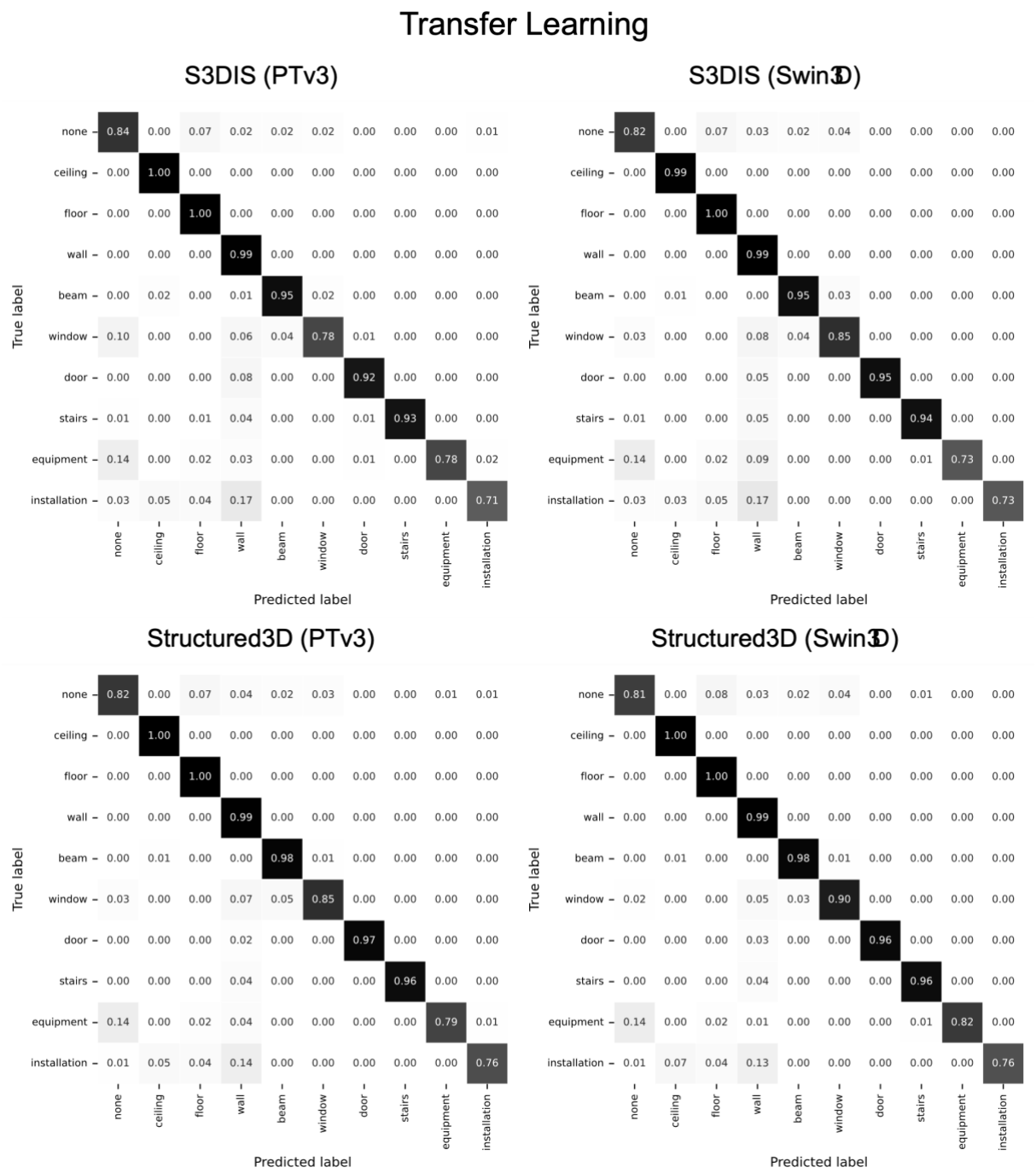

Figure 19: Confusion matrices from the transfer learning experiments for 3D semantic segmentation of 10 building component classes in LiDAR point clouds from shell construction sites. Results are shown for two model architectures: Point Transformer V3 (PTv3) and Swin3D. All models were pretrained on one of two datasets (S3DIS or Structured3D) and then fine-tuned and evaluated on a custom validation dataset focused on shell construction site scenes.