The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-M-9-2025
30th CIPA Symposium "Heritage Conservation from Bits:
From Digital Documentation to Data-driven Heritage Conservation", 25–29 August 2025, Seoul, Republic of Korea

Assessing Generalization Capability of 3D Semantic Segmentation Algorithms using 3D Point Clouds of Cultural Heritage

Thodoris Betsas [1], Hlias Tsarpalis [1], Andreas Georgopoulos [1]

[1] Lab of Photogrammetry, School of Rural, Surveying and Geoinformatics Engineering, National Technical University of Athens, Greece
betsasth@mail.ntua.gr, h.tsarpalis@gmail.com, drag@central.ntua.gr

Keywords: 3D Semantic Segmentation, Cultural Heritage, Point Clouds, Deep Learning

Abstract

Cultural Heritage (CH) monuments are strongly characterized by detailed architectural elements, inherent complexity, and heterogeneity and therefore present unique challenges regarding 3D Semantic Segmentation (3DSS), which is a useful tool for documentation enhancement and for empowering preservation actions. This study explores the generalization capability of recent deep learning 3DSS architectures applied to cultural heritage (CH) point cloud data. Using the ArCH benchmark, we evaluate five representative models, including PointNet, PointNet++, Point Transformer v1, v2 and Omni-Adaptive CNNs. All models are assessed using a uniform pipeline and limited input features (XYZ and RGB). Both qualitative and quantitative results indicate that Point Transformer v1 achieves strong performance on unseen CH data (61.3 mIoU), suggesting a potential link between architectural design and generalization ability in CH domain. These findings highlighting the need for further research under varying configurations and broader evaluation settings, especially for recent deep learning architecturs e.g., transformers.

1. Introduction

In recent years, Deep Learning (DL) algorithms have demonstrated strong generalization capabilities across indoor, outdoor and hybrid environments. Generalization refers to model's ability to capture the underlying data patterns from the training set while avoiding overfitting (Neyshabur et al., 2017; Zhang et al., 2016). To robustly assess generalization capability, the models are typically evaluated on unseen data, i.e., data not used during training or validation. The notion of "unseen data" can vary. For instance, portions of the same object could be excluded from training and validation sets or entirely new objects that share similar characteristics with those used in training. Evaluating DL models using new objects with similar characteristics, instead of portions of the same object, offers a more rigorous assessment of their generalization capability, indicating model robustness and applicability in novel environments. Cultural Heritage (CH) monuments are characterized by complex architectural elements which exhibit high heterogeneity and variability (Pierdicca et al., 2020). To this end, deep learning architectures for 3D semantic segmentation often struggle to generalize effectively to unseen CH data, despite demonstrating strong performance in less complex environments like e.g., an office. Recent applications in the CH domain revealed a weak generalization capability, especially using unseen data from monuments that were not partially included in the training set (Cao and Scaioni, 2021; Pierdicca et al., 2020). Additionally, the application of recent 3DSS DL architectures, especially transformers, on point clouds of cultural heritage remains underexplored (Cao and Scaioni, 2021; Matrone et al., 2020a; Pierdicca et al., 2020; Zhao et al., 2024). Based on the above, the following research questions are posed:

- Can recent deep learning architectures for 3D semantic segmentation, generalize effectively using cultural heritage data? and
- Which are the factors influencing the generalization capabilities of recent 3D semantic segmentation algorithms in the aspect of CH data?

To address these questions, this study investigates the 3DSS task within the CH domain, by comparing and analysing the generalization capability of recent 3DSS algorithms using CH data. The evaluation procedure is specifically designed to assess the generalization capability of the 3DSS models. Furthermore, an analysis of the generalization capability of each 3DSS model considering its architecture is conducted, aiming at initiating a preliminary exploration into the relationship between model architecture and generalization on the CH domain. To sum up, the contributions of this study are:

- A comparison of recent, underexplored DL 3DSS algorithms using CH data.
- A preliminary exploration into the relationship between 3DSS model architectures and generalization on the CH domain.
- Achieving SoTA performance on the ArCH benchmark (Matrone et al., 2020b), using only the XYZ coordinates and RGB values.

2. Related Work

2.1 Point Cloud 3D Semantic Segmentation

Semantic Segmentation is defined as the association of each element of the data under process with a meaningful label. Using 3D point clouds each 3D point is associated with a label, indicating its category. In general, 3DSS methods are classified into the -Point, -Dimensionality Reduction, -Discretization, -Graph and -Hybrid based methods (Betsas et al., 2025). The Point Based methods use the raw 3D point cloud to extract meaningful features for 3DSS and can be further classified into the Point-wise MLP e.g., PointNet(Qi et al., 2017), and PointNet++(Qi et al., 2017), Point Convolution (Thomas et al., 2019), Recurrent Neural Networks (Huang et al., 2018) and Attention mechanism & Transformers, categories e.g., Point Transformer v1 (PTv1) (Zhao et al., 2021) and Point Transformer v2 (PTV2) (Wu et al., 2022). The Discretization based methods transform the given point cloud to a new 3D or multi-dimensional discrete representation and then apply convolution for 3DSS e.g., OACNNs (Peng et al., 2024). In this effort, four -Point and one -Discretization based methods i.e., PointNet, PointNet++, PTv1, PTv2 and OACNNs, are compared regarding their generalization capability on CH data.

2.2 3D Semantic Segmentation on CH data

Traditional machine learning (ML) algorithms, such as Random Forest, have demonstrated strong performance in 3DSS involving applications in CH domain. However, these methods typically exhibit limited generalization capabilities when applied to previously unseen scenes (Grilli et al., 2019; Grilli and Remondino, 2019). Moreover, achieving improved generalization with traditional ML approaches often necessitates a specialized manual extraction and evaluation of handcrafted features (Grilli and Remondino, 2020). To handle this limitation, Pierdicca et al. (2020) built on top of the DGCNN (Phan et al., 2018) network proposing a modified version of it, suitable for 3DSS on CH data. In detail, the modified DGCNN incorporates as input the original and normalized 3D coordinates, the colors, expressed in HSV color space, and the normal vectors. The proposed architecture was applied on the ArCH dataset achieving SoTA results in terms of generalization capability (Matrone et al., 2020a). Moreover, Cao and Scaioni 2021 also built on top of the DGCNN network and proposed the 3DLEB-Net, a label-efficient network that aimed to improve generalization capability while reducing the required amount of labeled data during training. Recently, Zhao et al. (2024) proposed the DSC-Net aiming to capture the fine details of the ArCH dataset by including a discriminative spatial contextual attention mechanism. The DSC-Net achieved high-end results using k-fold cross validation on the ArCH datasets, demonstrating the effectiveness and potential of attention-based methods in the CH domain. Another line of methods, use the mature 2D Semantic Segmentation methods on images and then project the labels in 3D space using voting techniques (Murtiyoso et al., 2022), achieving high-end results even in CH data (Pellis et al., 2022a,b). However, the projection process unavoidably leads to a loss of spatial and semantic information. In this effort, we mainly compare Point based methods because they theoretically preserve the fine-grained spatial information presented in detailed 3D point clouds.

3. Experiment Setup and Methodology

In general, the ArCH dataset includes 17 annotated scenes acquired using various sensors, including DSLR and LiDAR. Of these, 15 scenes are designated for training and 2 for testing. In this effort, each 3DSS model was trained on the 15 training scenes and validated using the “B_SMV” scene. The “A_SMG” test scene, which was excluded during training and validation, served as a fully unseen dataset to evaluate generalization. After training, the model checkpoint achieving the highest validation mIoU was selected for testing on scene A. All the experiments were conducted using only the XYZ coordinates and RGB values i.e., excluding normals etc. Additionally, the hardware specifications of the computing system used for training, validation and testing of the recent 3DSS algorithms, are presented in Table 1. In this effort, five deep learning 3DSS algorithms were

	CPU	GPU	RAM
Asus ROG Zephyrus G15	Ryzen 9 5900HS 3,3 GHz	NVDIA GeForce RTX 3070 (8GB)	40 GB DDR5

Table 1. Hardware specifications of the computing system used for training and evaluating the 3DSS models

trained and evaluated on CH data, the PointNet, PointNet++, PTv1, PTv2 and OACNNs. To achieve that, existing GitHub implementations of them were exploited (Table 2). Specifically, Pointcept (Pointcept-Codebase, 2025) is an open-source codebase designed for point cloud perception tasks. It incorporates a wide range of recent 3DSS algorithms, along with preprocessing pipelines tailored to commonly used 3DSS benchmarks. Notably, Pointcept includes the original implementations of PTv1, PTv2 and OA-CNNs, along with many configuration files. However, the ArCH dataset is not natively supported within the Pointcept repository, and therefore, the necessary configuration and preprocessing files required to apply the included 3DSS methods to ArCH are not provided. To address this gap, all the auxiliary files, such as configuration files and preprocessing scripts, have been developed to enable the application of PTv1, PTv2 and OA-CNNs to the ArCH dataset within Pointcept. Finally, to facilitate the application of PointNet and PointNet++ on the ArCH dataset using their PyTorch implementation (Table 2), a dedicated data loader script has been developed. Apart from the necessary auxiliary files, the DL algorithms typically involve numerous hyperparameters that require careful tuning to optimize performance; however, in this study the best settings provided for each model, regarding the S3DIS benchmark, were retained without further modification. Key examples include the learning rate, weight decay, and the number of training epochs, among others. Table 3 provides a summary of selected hyperparameters and implementation details specific to each algorithm considered in this study.

DL Algorithm	GitHub Repository
PointNet	https://github.com/yanx27/Pointnet_Pointnet2_pytorch
PointNet++	
Point Transformer v1	https://github.com/Pointcept/Pointcept
Point Transformer v2	
Omni-Adaptive CNNs	

Table 2. GitHub repositories for each DL 3DSS architecture

Algorithm	Epochs	Learning Rate	Weight Decay	Optimizer	Training Time(h)	Batch Size	Loss	Validation mIoU
PointNet	20	0.001	0.005	Adam	3	16	Negative Log Likelihood	0.059
PointNet++	20	0.001	0.005	Adam	3	16	Negative Log Likelihood	0.035
PTv1	100	0.006	0.05	AdamW	33	1	Cross Entropy	0.648
PTv2	100	0.006	0,05	AdamW	22	1	Cross Entropy	0.375
OA-CNNs	100	0.001	0.02	AdamW	15	1	Cross Entropy	0.328

Table 3. Summary of the main hyperparameters and implementation details for each deep learning algorithm evaluated in this study

Deep learning algorithms performing 3DSS are commonly evaluated using a range of performance metrics, such as Accuracy, Precision, Recall, F1-score, and Intersection over Union (IoU), among others. In this effort, the per-class Accuracy (Eq 1), Precision (Eq 2), Recall (Eq 3), F1-score (Eq 4) and IoU (Eq 5) as well as their mean values are calculated. Of course, the Pointcept codebase includes the implementations of these metrics; however, we also evaluated the performance of each algorithm

using SciKit Learn (SciKit-Learn, 2025) python module, for consistency among the compared methods.

$$OAcc = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - score = 2\frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (5)$$

where: TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

Data augmentation is a well-established technique employed to enhance the generalization capability of DL models. Specifically, various transformations such as rotation, scaling, and jittering, are applied to the training data in order to generate additional, synthetic instances. The primary objective of data augmentation is to increase the diversity and variability within the training set, thereby enabling the model to generalize more effectively to unseen data. The Pointcept codebase contains different data augmentation techniques, defined in the configuration file of each 3DSS DL model. In the present study, data augmentation was enabled; however, the default settings provided for each model were retained without further modification. In Table 4, the specific data augmentation strategies applied during the training of each 3DSS method are summarized.

Data Augmentation	PTv1	PTv2	OACNNs
CenterShift	✓	✓	✓
RandomScale	✓	✓	✓
RandomFlip	✓	✓	✓
RandomJitter	✓	✓	✓
ChromaticAutoContrast	✓	✓	✓
ChromaticTranslation	✓	✓	✓
ChromaticJitter	✓	✓	✓
GridSample	✓	✓	✓
SphereCrop	✓	✓	✓
CenterShift	✓	✓	✓
NormalizeColor	✓	✓	✓
RandomDropout			✓
RandomRotate (x, y, z)			✓
ElasticDistortion			✓
ShufflePoint			✓

Table 4. Data augmentation techniques applied during training to PTv1, PTv2, and OACNNs models

4. Experimental Results

In general, the performance of the 3DSS DL methods on the ArCH benchmark, is evaluated through both quantitative metrics and qualitative visualizations. This section presents the evaluation results for the PointNet, PointNet++, PTv1, PTv2 and OACNNs methods. For each method, the best performing model, selected based on the validation performance, was applied to the “SMG_A” test scene of the ArCH dataset. The quantitative results are summarized in Tables 5, 6, 7, 8, and 9, which present the mean and per-class, IoU, Accuracy, Precision, Recall and F1-score metrics, respectively. Complementing these metrics, Figures 1, 2, and 3 illustrate different visual comparisons among the recent DL methods.

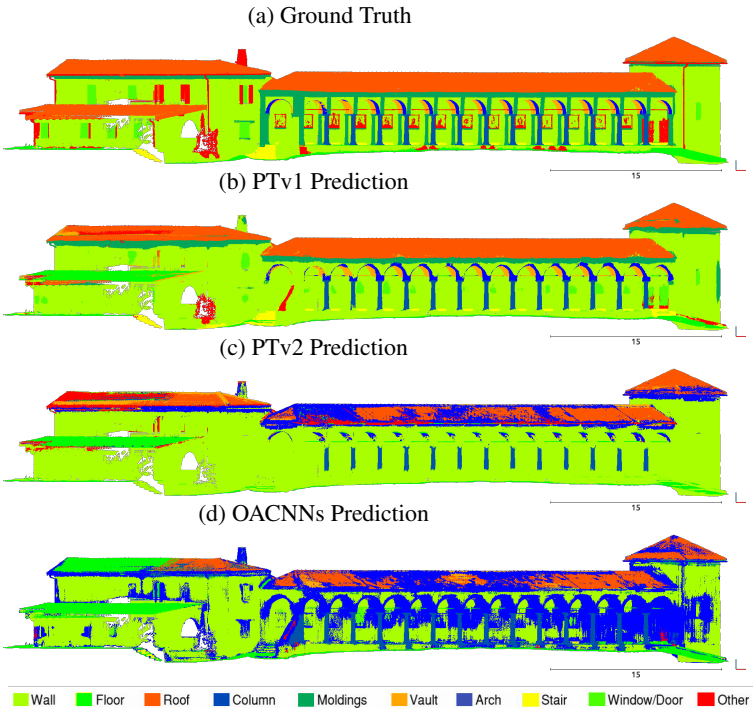

Figure 1. Qualitative Analysis on SMG_A test scene (Left View)

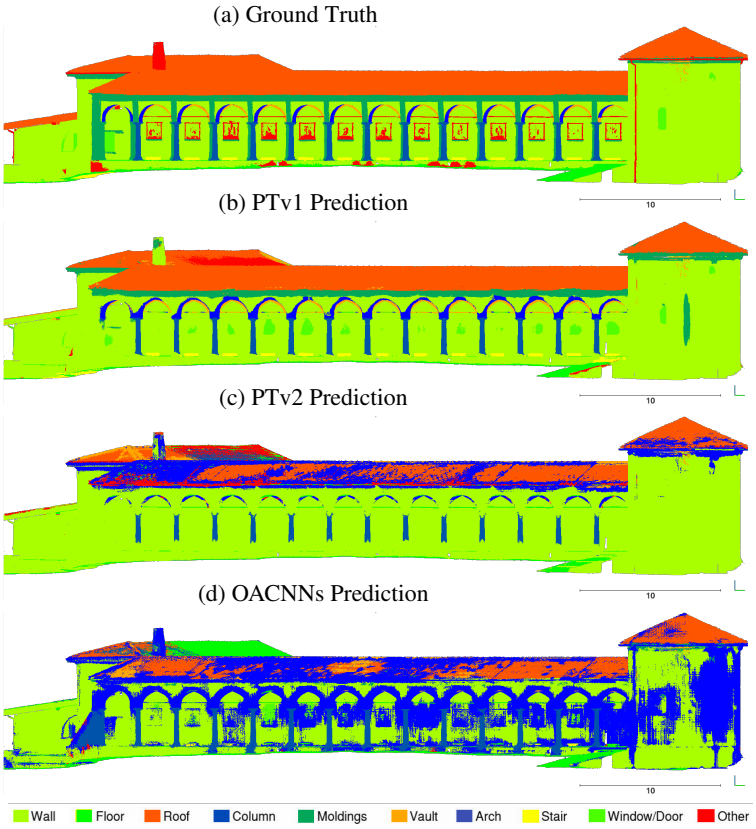

Figure 2. Qualitative Analysis on SMG_A test scene (Right View)

DL Model	Training	Validation	Test	mIoU	Arch	Column	Moldings	Floor	Door Window	Wall	Stairs	Vault	Roof	Other
PointNet	all	B	A	0.098	0.000	0.000	0.000	0.000	0.000	0.428	0.000	0.000	0.547	0.000
PointNet++	all	B	A	0.038	0.000	0.000	0.000	0.000	0.000	0.380	0.000	0.000	0.000	0.000
DGCNN	all	B	A	0.376	0.001	0.233	0.108	0.614	0.085	0.681	0.282	0.555	0.826	n/a
DGCNN Mod+3Dfeat	all	B	A	0.599	0.210	0.795	**0.451**	**0.867**	0.087	0.765	0.391	**0.866**	**0.963**	n/a
OACNNs	all	B	A	0.237	0.037	0.519	0.047	0.47	0.067	0.471	0.152	0.347	0.228	0.033
PTv1	all	B	A	0.556 (**0.613**)	**0.539**	**0.881**	0.364	0.731	**0.190**	**0.778**	**0.550**	0.804	0.674	0.04
PTv2	all	B	A	0.199	0.023	0.416	0.003	0.453	0.000	0.698	0.001	0.090	0.248	**0.055**

Table 5. The mIoU and Per-class IoU for each 3DSS DL method evaluated on the “A_SMG” test scene. PTv1 achieves 61.3% mIoU excluding the “Other” class, similarly to the compared methods. The results for the DGCNN and its variants were collected by the ArCH benchmark.

DL Model	Training	Validation	Test	mAcc	Arch	Column	Moldings	Floor	Door Window	Wall	Stairs	Vault	Roof	Other
DGCNN	all	B	A	0.784	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
DGCNNMod+3Dfeat	all	B	A	0.914	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
OACNNs	all	B	A	0.889	0.64	0.98	0.926	0.869	0.987	0.79	0.991	0.916	0.828	**0.958**
PTv1	all	B	A	**0.963**	**0.989**	**0.997**	**0.943**	**0.957**	0.986	**0.902**	**0.994**	**0.972**	**0.928**	0.942
PTv2	all	B	A	0.909	0.870	0.981	0.925	0.872	**0.992**	0.849	0.991	0.878	0.826	0.911

Table 6. The mAcc and Per-class Accuracy for each 3DSS DL method evaluated on the “A_SMG” test scene. The results for the DGCNN and its variants were collected by the ArCH benchmark.

DL Model	Training	Validation	Test	mPre	Arch	Column	Moldings	Floor	Door Window	Wall	Stairs	Vault	Roof	Other
DGCNN	all	B	A	0.822	0.001	0.886	0.173	0.883	0.186	0.729	0.389	0.625	0.959	n/a
DGCNNMod+3Dfeat	all	B	A	**0.917**	0.532	0.849	0.650	**0.957**	0.135	**0.879**	0.466	**0.891**	0.975	n/a
OACNNs	all	B	A	0.584	0.038	0.527	**0.772**	0.489	0.143	0.84	**0.841**	0.798	**0.993**	**0.405**
PTv1	all	B	A	0.737	**0.646**	**0.998**	0.705	0.797	**0.272**	0.808	0.61	0.823	0.976	0.127
PTv2	all	B	A	0.379	0.026	0.570	0.515	0.500	0.000	0.705	0.013	0.421	0.954	0.088

Table 7. The mPrec and Per-class Precision for each 3DSS DL method evaluated on the “A_SMG” test scene. The results for the DGCNN and its variants were collected by the ArCH benchmark.

DL Model	Training	Validation	Test	mPre	Arch	Column	Moldings	Floor	Door Window	Wall	Stairs	Vault	Roof	Other
DGCNN	all	B	A	0.784	0.002	0.240	0.226	0.668	0.136	0.912	0.509	0.833	0.856	n/a
DGCNNMod+3Dfeat	all	B	A	**0.914**	0.258	0.925	**0.596**	0.903	0.196	0.855	0.710	0.969	**0.988**	n/a
OACNNs	all	B	A	0.431	**0.926**	**0.973**	0.048	**0.934**	0.112	0.518	0.156	0.381	0.229	0.035
PTv1	all	B	A	0.759	0.767	0.883	0.431	0.899	**0.390**	**0.956**	**0.85**	**0.972**	0.686	0.074
PTv2	all	B	A	0.310	0.196	0.607	0.003	0.829	0.000	0.986	0.001	0.102	0.251	**0.125**

Table 8. The mRec and Per-class Recall for each 3DSS DL method evaluated on the “A_SMG” test scene. The results for the DGCNN and its variants were collected by the ArCH benchmark.

DL Model	Training	Validation	Test	mF1	Arch	Column	Moldings	Floor	Door Window	Wall	Stairs	Vault	Roof	Other
DGCNN	all	B	A	0.794	0.001	0.378	0.196	0.761	0.157	0.811	0.441	0.714	0.905	n/a
DGCNNMod+3Dfeat	all	B	A	**0.915**	0.347	0.886	**0.622**	**0.929**	0.160	0.867	0.563	**0.928**	**0.982**	n/a
OACNNs	all	B	A	0.379	0.073	0.684	0.091	0.642	0.126	0.641	0.264	0.515	0.373	0.064
PTv1	all	B	A	0.736	**0.701**	**0.937**	0.535	0.845	**0.320**	**0.876**	**0.710**	0.892	0.806	0.093
PTv2	all	B	A	0.275	0.045	0.588	0.007	0.624	0.000	0.822	0.001	0.165	0.397	**0.104**

Table 9. The mF1-score and Per-class F1-score for each 3DSS DL method evaluated on the “A_SMG” test scene. The results for the DGCNN and its variants were collected by the ArCH benchmark.

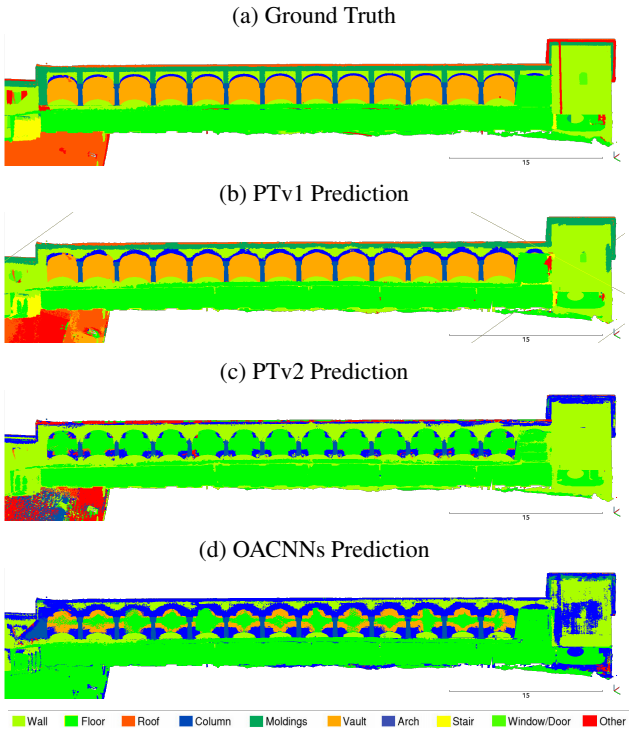

Figure 3. Qualitative Analysis on “SMG_A” test scene (Vaults)

Furthermore, the training loss per epoch for each method is depicted in Figure 4 and the evaluation loss per epoch for each method is presented in Figure 5

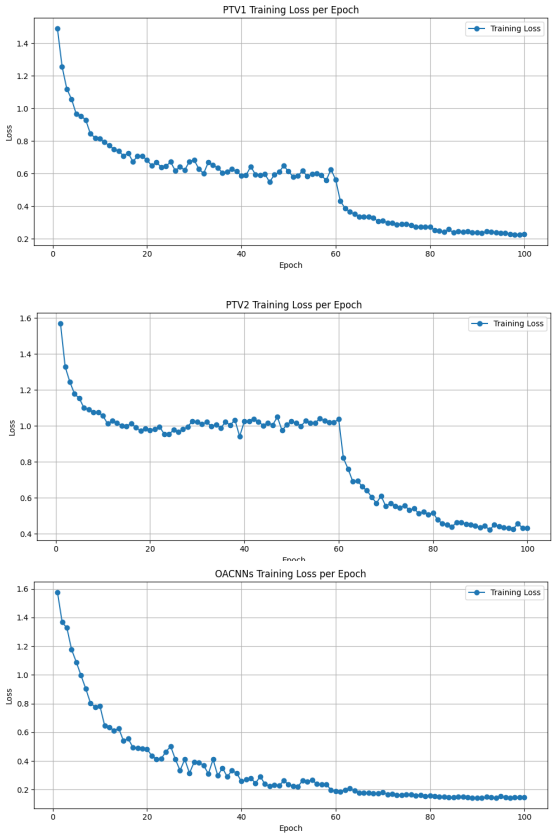

Figure 4. The training loss for PTv1, v2 and OACNNs models

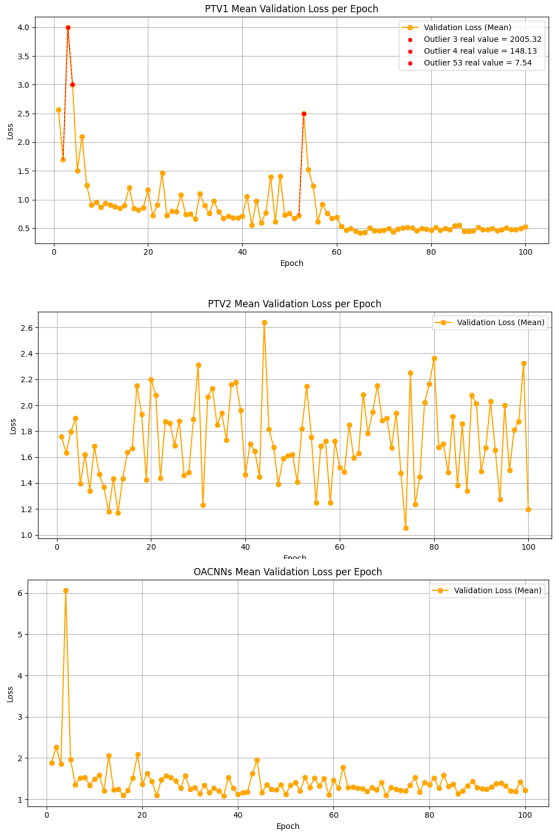

Figure 5. The validation loss for PTv1, v2 and OACNNs models

5. Discussion

3D Semantic Segmentation plays a significant role in 3D digitization and analysis of CH monuments. In general, CH monuments are characterized by high variability, complexity and fine details, revealing extreme challenges for the 3DSS algorithms e.g. generalization capability. Most of the methods studying DL 3DSS on CH data use either the original or a variant of the DGCNN network (Cao and Scaioni, 2021; Cao et al., 2022; Matrone et al., 2020a; Pierdicca et al., 2020). Additionally, traditional DL methods like PointNet, PointNet++ struggle to learn the complex patterns included in CH monuments (Tsarpalis, 2025). To the best of our knowledge, recent 3DSS algorithms, like PTv1, PTv2 and OACNNs have not been evaluated on CH 3D point clouds yet, despite of their SoTA performance on other domains. The experimental results presented in this study reveal a strong performance of PTv1 on th ArCH benchmark compared to previous SoTA methods e.g., DGCNN and even to recent architectures i.e., PTv2 and OACNNs, using only XYZ and RGB. Based on the qualitative and quantitative analysis, PTv1 seems to better handle the class imbalance problem, achieving improved results on the minor classes (Table 10, Figure 6 and 7), compared to SoTA. Also, PTv1 achieves high-end results, comparing to both SoTA and recent DL 3DSS methods, on the repetitive classes like Column, Arch, and Vault (Figure 7). However, compared to PTv1, DGCNN Mod+3Dfeat achieves high-end results on classes like Floor and Roof with a strong per-class IoU difference of +13.6% and 28.9% respectively (Table 5). The reduced performance of PTv1 on the Roof category is mainly due to the misclassifciation of it as Floor. Figures 1 and 2 revealed a strong performance of PTv1 on the class Roof; however, when the roof is flat and in a different level, PTv1 misclassifies it as Floor (Figure 8) and thus the

IoU performance is significantly reduced. This could be attributed to the fact that the training data of the ArCH benchmark, primarily consists of gable and hip roofs, rather than shed or flat roof types. As a result, the network may mainly confuse the roof with the upper floor surface (Figure 8). Finally, in the dominant class of Wall, PTv1 achieve a slightly better IoU performance than SoTA with approximately +1% difference. Wall performance is mainly reduced due to the misclassifications of Moldings class. PTv1 achieve to classiy Modlings class when it is located under the Roof class; however it struggles to distinguished it from Wall (Figure 1, 2, 3, 6 and 7)

Class	Percentage (Training set)	SoTA IoU	PTv1 IoU	Difference
Arch	4%	21%	53.9%	+32.9%
Columns	2.3%	79.5%	88.1%	+8.6%
Door Window	5%	0.87%	19%	+18.3%
Stairs	0.7%	39.1%	55%	+15.9%

Table 10. Per-Class IoU difference, for minor classes, between SoTA and PTv1

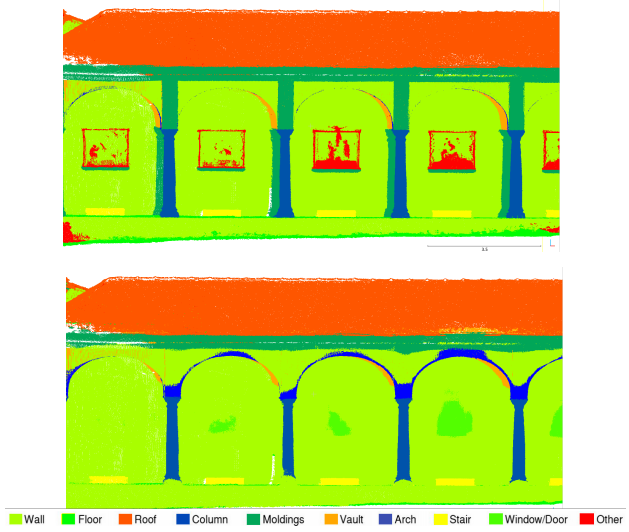

Figure 6. Close view on the performance of PTv1 on stairs class (yellow). Ground Truth (Up), PTv1 (Down)

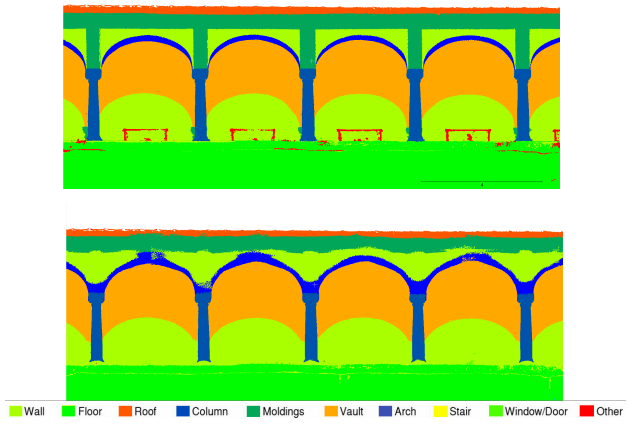

Figure 7. Close view on the performance of PTv1 on repetitive class e.g., columns, arches and vaults. Additionally, a close view to columns, arch and moldings minor classes. Ground Truth (Up), PTv1 (Down)

Figure 8. Close view on the performance of PTv1 on Roof class. Ground Truth (Up & Middle), PTv1 (Down)

The ArCH benchmark includes more chapel scenes than porticos (Cao et al., 2022). Also, there is not a predefined validation scene. Additionally, it provides two test scenes, one chapel

(SMG_B) and one portico (SMG_A). To assess the generalization capability of the 3DSS methods, we chose the chapel scene as the validation set and the portico scene as the test set. This choice, allows the effective monitoring of the learning process and a fair assessment of the generalization capability. In Figures 4 and 5 the per epoch training and evaluation loss plots are respectively presented for each method. In general, the training losses are decreased smoothly. PTv1 and PTv2 experience a plateau from epoch 20 to 60; however after the application of the loss scheduler on epoch 60 both of them seem to further reduce the training loss. The difference between them occur to the validation loss. PTv1 achieves to gradually standardize the validation loss after epoch 60, while PTv2 continues to fluctuate, revealing that the learned patterns are not descriptive for the "B_SMV" scene. Moreover, OACNNs training plot is smoothly decreasing during the entire training period; however the validation loss follows that of PTv2 also revealing a struggle of learned patterns describing the "B_SMV" scene. All the experiments revealed a higher validation than training loss as expected. Table 3 presents for PTv1, PTv2 and OA-CNNs a validation mIoU of 64.8%, 37.5% and 32.8%, respectively. Meanwhile, PTv1, PTv2 and OACNNs achieve a test mIoU of 61.3%, 21.4% and 25.9%, respectively. PTv2 and OACNNs experienced a -16.1% and -6.9% drop while PTv1 only a -3.5% drop, between the validation mIoU and test mIoU.

In general, the ArCH dataset contains fine-grained 3D point clouds of monuments with high variability and heterogeneity. The vector attention of PTv1 seem to better capture the fine-detailed CH data patterns than the other 3DSS architectures, including PTv2. Specifically, PTv1 vector attention mechanism modulates each channel of the value vector by a separate vector attention. While offering high representational capacity, this approach leads to drastically increase the learnable parameters of the network, as we go deeper, flirting with overfitting and preventing the deployment of deeper networks (Wu et al., 2022). This was led Wu et al. (2022), grouping the channels and applying group attention, reducing the learnable parameters and increasing models' depth. Despite PTv2 being an architectural successor to PTv1 and generally demonstrating superior performance across various benchmarks (Wu et al., 2022), this study reveals a seemingly contradictory finding: PTv1 achieved better generalization capability than PTv2 on the ArCH dataset. A hypothetical explanation of this finding could be that PTv1 attention as a more expressive implementation captures better the underlying patterns of the highly detailed CH data; however, in this study we do not exhaust models' fine-tuning and therefore, further systematic hyperparameter optimization and architectural exploration are necessary to draw definitive conclusions regarding the generalization capabilities and the factors influencing performance for PTv1, PTv2, OACNNs and other DL 3DSS methods, in CH domain. Furthermore, it is important to take into account the computational constraints encounter during this study. Table 1 presents the hardware specifications of the system used during training and inference, revealed limited computational resources. Consequently, the large point clouds were cropped into many smaller scenes to facilitate processing. This necessary process could potentially hinder the extraction of global features leading to lower, than the optimal, performance. Despite these limitations, PTv1 achieved high-end results compared to SoTA methods in CH domain. The comparative assessment of PTv1 and PTv2 architectures in this study underscores a potential association between their network architecture and their generalization capabilities, particularly in the context of highly detailed 3D point cloud data. In general, the inference time is significantly influenced by the available computation power (Table 1). In the presented experiments the inference time varies significantly. Specifically, although the PTv1 algorithm achieves the best results, it requires nearly 5 hours to process the "SMG_A" scene. In comparison, PTv2 requires approximately 2 hours, while OACNNs complete the inference of "SMG_A" scene, in about 1 hour. These results highlight a critical trade-off between semantic segmentation performance and computational efficiency. Overall, transformers seem to have the capability to express the underlying patterns in CH data and achieve high-end results in 3DSS using a proper hyper parameter tuning and sufficient computational resources.

6. Conclusions

This study, presents a comparative assessment of the generalization capability of PointNet, PointNet++, PTv1, PTv2 and OACNNs algorithms using the ArCH benchmark. All the algorithms are evaluated on the benchmark by assessing, detailed qualitative and quantitative results, revealing high-end results for PTv1. While this is contradictory as PTv2 being an architectural successor to PTv1, comparing PTv1 and PTv2 architectures revealed a potential association between their network architecture and their generalization capabilities, particularly in the context of highly detailed 3D point cloud data. However, further systematic hyperparameter optimization and architectural exploration are necessary to draw definitive conclusions regarding the generalization capabilities and the factors influencing performance for PTv1, PTv2, and OACNNs in CH domain, despite the great performance achieved by PTv1 regarding the challenging CH data. Afterall, transformers seem to have the capability to express the underlying patterns in CH data and achieve high-end results in 3DSS using a proper hyper parameter tuning and sufficient computational resources.

References

Betsas, T., Georgopoulos, A., Doulamis, A., Grussenmeyer, P., 2025. Deep Learning on 3D Semantic Segmentation: A Detailed Review. *Remote Sensing*, 17(2), 298.

Cao, Y., Scaioni, M., 2021. 3DLEB-Net: Label-Efficient Deep Learning-Based Semantic Segmentation of Building Point Clouds at LoD3 Level. *Applied Sciences*, 11(19). https://www.mdpi.com/2076-3417/11/19/8996.

Cao, Y., Teruggi, S., Fassi, F., Scaioni, M., 2022. A comprehensive understanding of machine learning and deep learning methods for 3d architectural cultural heritage point cloud semantic segmentation. *Italian Conference on Geomatics and Geospatial Technologies*, Springer, 329–341.

Grilli, E., Özdemir, E., Remondino, F., 2019. Application of machine and deep learning strategies for the classification of heritage point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 447–454.

Grilli, E., Remondino, F., 2019. Classification of 3D digital heritage. *Remote Sensing*, 11(7), 847.

Grilli, E., Remondino, F., 2020. Machine learning generalisation across different 3D architectural heritage. *ISPRS International Journal of Geo-Information*, 9(6), 379.

Huang, Q., Wang, W., Neumann, U., 2018. Recurrent slice networks for 3d segmentation of point clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Matrone, F., Grilli, E., Martini, M., Paolanti, M., Pierdicca, R., Remondino, F., 2020a. Comparing machine and deep learning methods for large 3D heritage semantic segmentation. *ISPRS International Journal of Geo-Information*, 9(9), 535.

Matrone, F., Lingua, A., Pierdicca, R., Malinverni, E. S., Paolanti, M., Grilli, E., Remondino, F., Murtiyoso, A., Landes, T., 2020b. A benchmark for large-scale heritage point cloud semantic segmentation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 1419–1426.

Murtiyoso, A., Pellis, E., Grussenmeyer, P., Landes, T., Masiero, A., 2022. Towards semantic photogrammetry: Generating semantically rich point clouds from architectural close-range photogrammetry. *Sensors*, 22(3), 966.

Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N., 2017. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30.

Pellis, E., Murtiyoso, A., Masiero, A., Tucci, G., Betti, M., Grussenmeyer, P., 2022a. 2D to 3D Label propagation for the semantic segmentation of Heritage building point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 861–867.

Pellis, E., Murtiyoso, A., Masiero, A., Tucci, G., Betti, M., Grussenmeyer, P., 2022b. An image-based deep learning workflow for 3d heritage point cloud semantic segmentation. *9th International Workshop 3D-ARCH” 3D Virtual Reconstruction and Visualization of Complex Architectures”. 2–4 March 2022, Mantua, Italy*, 46, ISPRS, 426–434.

Peng, B., Wu, X., Jiang, L., Chen, Y., Zhao, H., Tian, Z., Jia, J., 2024. Oa-cnns: Omni-adaptive sparse cnns for 3d semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21305–21315.

Phan, A. V., Le Nguyen, M., Nguyen, Y. L. H., Bui, L. T., 2018. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, 108, 533–543.

Pierdicca, R., Paolanti, M., Matrone, F., Martini, M., Morbidoni, C., Malinverni, E. S., Frontoni, E., Lingua, A. M., 2020. Point cloud semantic segmentation using a deep learning framework for cultural heritage. *Remote Sensing*, 12(6), 1005.

Pointcept-Codebase, 2025. Pointcept: A codebase for point cloud perception research. [Accessed 22-05-2025].

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

SciKit-Learn, 2025. scikit-learn: machine learning in Python &x2014; scikit-learn 1.6.1 documentation — scikit-learn.org. [Accessed 22-05-2025].

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L. J., 2019. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Tsarpalis, H., 2025. 3d semantic segmentation using machine learning. Master’s thesis, School of Rural, Surveying and Geoinformatics Engineering, National Technical University of Athens, Athens, Greece. Unpublished (In Greek).

Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H., 2022. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35, 33330–33342.

Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhao, H., Jiang, L., Jia, J., Torr, P. H., Koltun, V., 2021. Point transformer. *Proceedings of the IEEE/CVF international conference on computer vision*, 16259–16268.

Zhao, J., Liu, R., Hua, X., Yu, H., Zhao, J., Wang, X., Yang, J., 2024. DSC-Net: learning discriminative spatial contextual features for semantic segmentation of large-scale ancient architecture point clouds. *Heritage Science*, 12(1), 274.