

## Article

# Semantic Point Cloud Segmentation with Deep-Learning-Based Approaches for the Construction Industry: A Survey

Lukas Rauch \*  and Thomas Braml

Institute of Structural Engineering, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany; thomas.braml@unibw.de

\* Correspondence: lukas.rauch@unibw.de

**Abstract:** Point cloud learning has recently gained strong attention due to its applications in various fields, like computer vision, robotics, and autonomous driving. Point cloud semantic segmentation (PCSS) enables the automatic extraction of semantic information from 3D point cloud data, which makes it a desirable task for construction-related applications as well. Yet, only a limited number of publications have applied deep-learning-based methods to address point cloud understanding for civil engineering problems, and there is still a lack of comprehensive reviews and evaluations of PCSS methods tailored to such use cases. This paper aims to address this gap by providing a survey of recent advances in deep-learning-based PCSS methods and relating them to the challenges of the construction industry. We introduce its significance for the industry and provide a comprehensive look-up table of publicly available datasets for point cloud understanding, with evaluations based on data scene type, sensors, and point features. We address the problem of class imbalance in 3D data for machine learning, provide a compendium of commonly used evaluation metrics for PCSS, and summarize the most significant deep learning methods developed for PCSS. Finally, we discuss the advantages and disadvantages of the methods for specific industry challenges. Our contribution, to the best of our knowledge, is the first survey paper that comprehensively covers deep-learning-based methods for semantic segmentation tasks tailored to construction applications. This paper serves as a useful reference for prospective research and practitioners seeking to develop more accurate and efficient PCSS methods.



**Citation:** Rauch, L.; Braml, T. Semantic Point Cloud Segmentation with Deep-Learning-Based Approaches for the Construction Industry: A Survey. *Appl. Sci.* **2023**, *13*, 9146. <https://doi.org/10.3390/app13169146>

Academic Editor: Dimitris Mourtzis

Received: 11 July 2023

Revised: 26 July 2023

Accepted: 31 July 2023

Published: 10 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

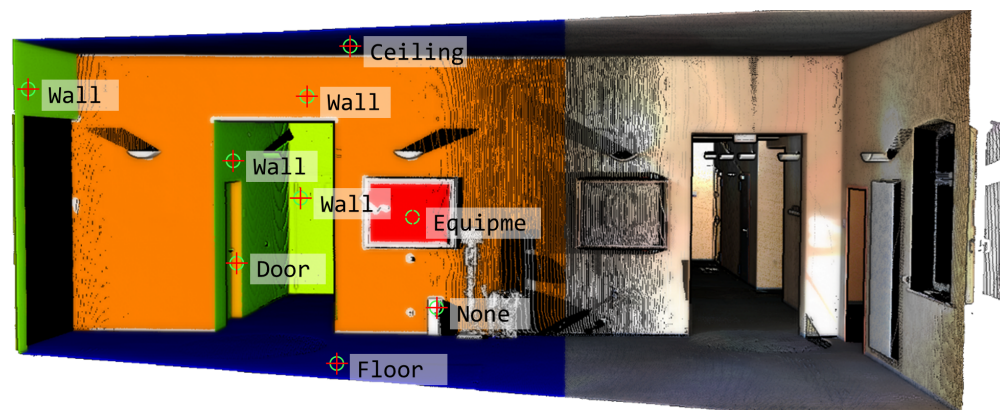
**Keywords:** point cloud; semantic segmentation; deep learning; machine learning; construction; automation; open source; dataset; survey

## 1. Introduction

To understand the semantic segmentation of point clouds in the 3D domain, it is helpful to first understand the origins of this technique in the 2D domain. Semantic segmentation, in the first place, is a computer vision technique that involves partitioning an image into multiple semantic regions or segments, where each segment corresponds to a specific region of interest within the image. The goal of semantic segmentation is to classify and assign a label to every pixel in an image, indicating the category or class it belongs to. Unlike other types of image segmentation, such as instance segmentation or boundary detection, semantic segmentation focuses on understanding the high-level content of an image rather than individual objects or their boundaries. It aims to capture the meaning or semantics of the image by assigning a meaningful label to each pixel. Before the introduction of machine learning, early adopters of semantic segmentation in images relied on traditional computer vision techniques. They extracted low-level features like color, texture, edges, and gradients from the images. Various segmentation algorithms were then applied to partition the image into regions based on these features. Techniques such as thresholding, region growing, and edge-based segmentation were commonly used [1]. This process aimed to identify distinct regions or boundaries within the image for

further analysis and classification. However, these approaches were limited by the need for handcrafted features and explicit rules, making them less flexible and robust compared to machine-learning-based methods. Machine learning models, such as deep neural networks, have since emerged as powerful tools that can automatically learn and extract complex features from images, significantly improving the accuracy and generalizability of semantic segmentation. The development of convolutional deep neural networks alleviated the need for engineered features and produced a powerful representation that captures texture, shape, and contextual information [2]. The introduction of fully convolutional networks (FCNs) to semantic segmentation tasks alleviated the need for preprocessing images into low-resolution superpixels and was end-to-end trained to predict a semantic label for each pixel in an image [3]. The FCN was able to achieve state-of-the-art performance on the PASCAL VOC 2012 dataset [4] and was the first model to surpass the performance of traditional computer vision techniques.

Similarly, the semantic segmentation of point cloud data refers to the process of classifying individual points in three-dimensional space into semantic categories or labels. It involves assigning a meaningful label to each point based on its characteristics and the context of the surrounding points. The objective is to segment the point cloud into different regions or objects, as shown in Figure 1, where each region or object is associated with a specific semantic class. Deep-learning-based segmentation for point clouds opens up new opportunities for applications, such as robotics, industrial automation, and 3D scene understanding also for the construction industry, enabling machines to perceive and interact with the environment in three-dimensional space. Among various scene understanding problems, 3D semantic segmentation allows for finding accurate object boundaries along with their labels in 3D space, which is useful for fine-grained tasks [5]. The architecture, engineering, and construction (AEC) sector ranks as one of the most intensive fields where vision-based systems are used to facilitate decision-making processes during the construction phase. Construction sites make efficient monitoring extremely tedious and difficult due to clutter and disorder. Due to their intrinsic nature, job sites are prone to management failures such as unsatisfactory construction quality, delays on schedule, extra costs, and even injuries [6]. Multiple surveillance tasks, especially site monitoring, are performed already with the help of image-based computer vision. Adding 3D data has the potential to significantly increase information density and the level of automation. Below, we list practical examples of applications where learning-based point cloud segmentation can be beneficial to quality, efficiency, and safety in construction.



**Figure 1.** Concept of point cloud segmentation with learned building classes. **Right:** raw point cloud with color information. **Left:** semantically enriched point cloud with predicted class labels output by a neural network. All points with the same predicted class are colored in the same color.

Shape and position control in concrete casting and precast installation requires finding accurate object/segment boundaries [7]. Volume calculations in the construction and demolition (C&D) phases can be automated from laser scanner, UAV, or satellite syn-

thetic aperture radar (SAR) footage with the help of semantic segmentation [8,9]. Reverse engineering digital twins of existing buildings (as built and scan to BIM) requires assigning element classes to scan data to reconstruct element geometries [10,11]. Site progress monitoring can be automated through 3D component recognition and comparison with a planned database [12,13].

In addition to many surveillance applications, semantic segmentation of real-time sensor data as a subfield of computer vision is one crucial component for guiding (partially) autonomous robots. We expect robots to take over even more construction tasks in the future, but sites are constantly changing environments and scattered with moving obstacles. Thus, real-time scene understanding will be mandatory. Price drops for sensor hardware have ultimately made it economically feasible to install the necessary hardware. However, software development is needed to utilize the information, as the following examples of non-learning approaches illustrate.

Wang et al. [14] used colored point cloud segmentation to estimate the position of precast concrete rebar elements. With appropriate training data and a learning-based approach, this can be applied to all types of construction equipment that robots interact with. Automated guided vehicles (AGVs) require vehicle guidance and collision avoidance. This can be performed with cameras, but imaging sensors rely on good lighting conditions. This is not always the case on construction sites, and light detection and ranging (LiDAR) can be a solution for this [15]. Finally, semantic-enhanced sensor data can dramatically improve the overall safety of construction sites where humans and machines work together. Ray and Teizer [16] used ray tracing to calculate the blind spots of construction equipment, but the algorithm cannot distinguish between multiple obstacles. Machine safety decision making can be improved with learning-based point cloud understanding.

The scope of this paper is to focus on learning-driven PCSS with deep neural networks, which can learn deep geometric connections of complex structures by transforming the feature space into a high-dimensional representation problem. The early adopters, PointNet [17] and PointNet++ [18], have proven that deep learning techniques are fundamental for many computer vision tasks in the sparse 3D point cloud domain. Multiple strategies were published in the following years, transferring approaches from image-based computer vision and natural language processing to point cloud processing (convolution-, recurrent-, graph-, and transformer-based methods). Within the civil engineering domain, only a few papers covered the use of deep learning to perform PCSS. Some remarkable publications automated as-built BIM generation [19,20], building reconstruction [21,22], and bridge part semantic segmentation [23,24]. Nevertheless, up to now, there is no settled trend for the civil engineering branch, and most approaches only feature outdated PointNet network architectures.

There have been previous reviews of deep learning for construction applications, from which we stand out. Guo et al. [25] and He et al. [26] have provided extensive surveys on deep learning with point clouds, with the latter focusing specifically on semantic segmentation. However, these surveys primarily concentrate on 3D scene understanding with a general approach. Further, they do not address the developments in the recently dominant Transformer network architectures. Other related review studies, such as those conducted by Jacobsen and Teizer [27], Akinosho et al. [28], and Khallaf and Khallaf [29], have explored machine learning applications in the construction industry. While they touch on point-cloud-based object classification, they focus on imaging methods for construction site monitoring and lack an in-depth and unbiased comparison of various deep learning methods specifically tailored for point cloud segmentation. Therefore, this paper presents an unbiased assessment and detailed comparison of the most popular deep learning methods developed in recent years, emphasizing their applicability to point cloud segmentation in the construction industry. For the first time, we address a critical challenge in this domain—the scarcity of industry-specific datasets for training deep learning models. To fill this gap, we have compiled an extensive table of relevant training and validation datasets, which, to our knowledge, has not been previously published to this extent.

By offering novel insights and original perspectives in comparison to existing review studies, our research reinforces its significance and contribution to the broader field of deep-learning-based approaches for the construction industry. Our goal with this article is to support future research in the fields of applied civil engineering and computer vision by establishing a solid foundation for further exploration. By outlining the most recent methodologies and resources available for training and validating new algorithms, we hope to unify benchmarking efforts and encourage wider participation within the research community.

The structure of this paper is as follows: Section 2 describes the methodology adopted in this study. Section 3 addresses the state of the art in industry-specific datasets for the construction domain. Section 4 discusses the issue of unbalanced datasets during training and presents potential solutions. Section 5 introduces and compares the most popular developments in deep-learning-based point cloud segmentation in recent years. Section 6 examines the findings of this survey, and finally, Section 7 concludes the paper.

## 2. Methodology

To provide a thorough view of PCSS with application in civil engineering, this literature review followed the steps described below. First, we searched for all topic-related publicly available datasets and compiled a look-up table of the resulting datasets. We started the internet search from the free and open [paperswithcode.com](https://paperswithcode.com) [30] database and expanded this list by identifying further datasets mentioned in papers that used these datasets for training and validation on point cloud scene understanding and semantic segmentation tasks. These papers were found in a conducted grid search with the following online search engines: [scopus.com](https://scopus.com), [semanticscholar.org](https://semanticscholar.org), and [scholar.google.com](https://scholar.google.com), by using a combination of keywords including “3D”, “dataset”, “point cloud”, “semantic segmentation”, “instance segmentation”, and “scene understanding”. The criterion for inclusion in our list was that the datasets must be freely accessible to all and are relatable to civil engineering in a broader sense. The exploration resulted in 52 datasets that met our criteria and would be further considered. First, we evaluated all datasets by the covered scene type, the utilized sensors, the data representation, the intended use case, the available features, and the dataset size. Second, we looked into the class imbalance problem in 3D data and how this is dealt with in the literature. Based on this, we collected solutions presented in the past to handle class imbalance. Third, we summarized the evaluation metrics that are commonly used by authors to evaluate and compare the performance of scene understanding tasks in 3D space. Lastly, we summarized and compared a selection of significant deep learning methods developed in the past for solving the 3D PCSS problem. We followed up on the work of [25,26], who had both previously conducted surveys on the general topic of deep learning on point clouds. We deepened this review to cover specific topics of the industry and collected the advantages and disadvantages of applying certain methods. As a measure of relevance in this survey, we quantified PCSS methods by its benchmark results (viz. mIoU) on different datasets, originally sourced from the [paperswithcode.com](https://paperswithcode.com) [30] database and cross-validated with its associated papers. Representative benchmark results on the S3DIS dataset [31] are given in Section 5 and Table A1. This survey only considers publications published up until the end of December 2022.

## 3. Public Datasets for Scene Understanding Methods

Efficient training in machine learning and the applicability of deep neural networks heavily depends on the available training data. While the performance of most shallow machine learning algorithms converges at some point, deep learning can increase performance relative to the amount of training data [32]. Creating huge datasets for the individual is not always feasible. Most protagonists in the machine learning community, therefore, necessarily rely on the use of freely available data. However, famous achievements demonstrate that the need for public datasets and the need for people to work together does not have to be a burden but offers great opportunities for accelerated progress.

The MNIST dataset, which features 70,000 gray-scale images of handwritten digits, was first published in 1999 [33] under a free-to-use, copy, share, and adapt license (Creative Commons license: CC BY-SA 3.0) and is still widely used as a benchmark for evaluating the performance of convolutional neural network (CNN) models [34]. The ImageNet database, which was introduced in 2012 (Russakovsky et al., 2015), has been instrumental in driving the development of deep learning algorithms in the community to groundbreaking innovation [35]. Three-dimensional point cloud semantic segmentation (PCSS) is a fast-growing field of research, and the amount and variety of open-source datasets to train, validate, and compare new model architectures are almost exclusively credited to the car industry and autonomous driving [36–40].

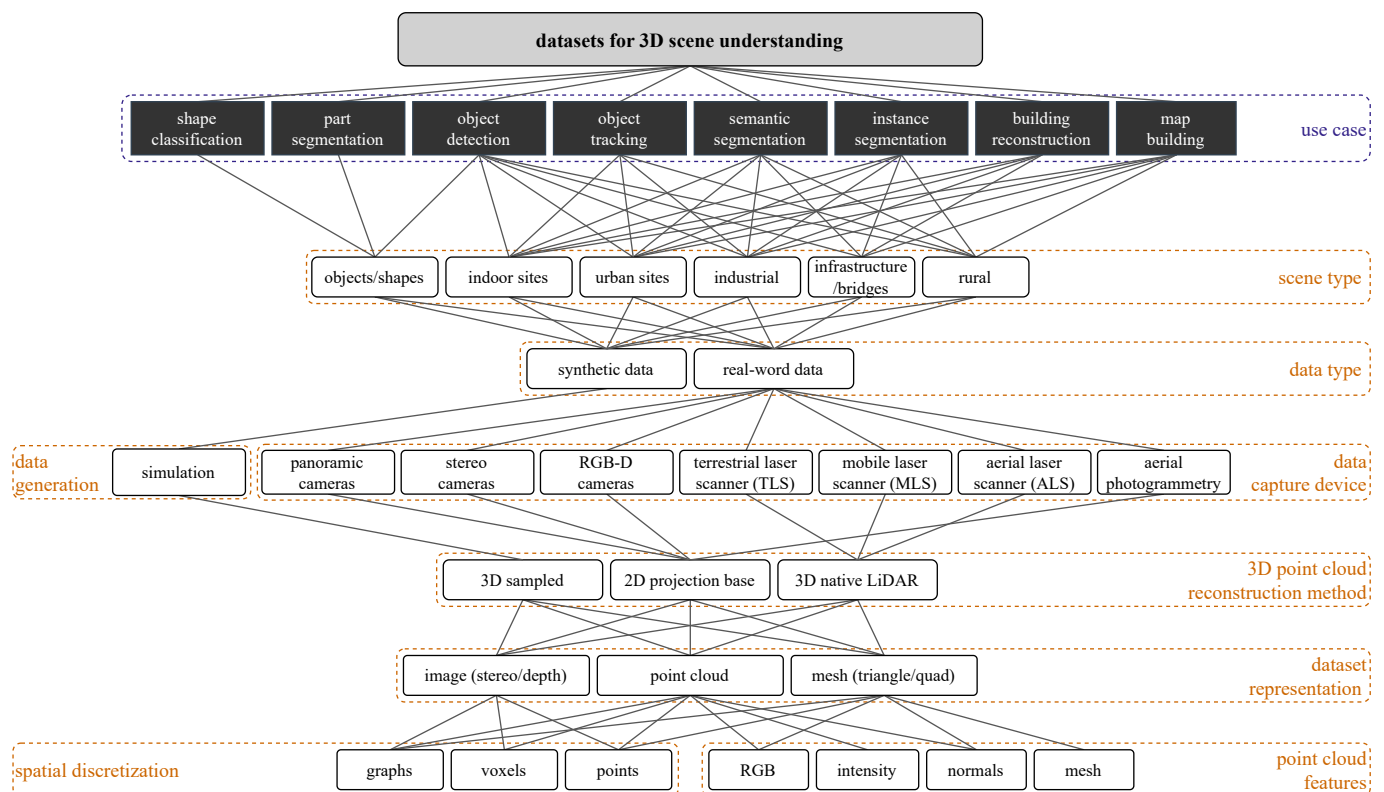
Three-dimensional datasets designed for computer vision scene understanding applications in the construction industry are rare and deviate in pure volume from their automotive counterparts. A literature review was conducted to summarize existing civil engineering-related open-source datasets and make them more visible for future research. The review was performed with a combination of search items including “3D”, “dataset”, “point cloud”, “scene understanding”, “semantic segmentation”, and “instance segmentation”, which revealed a range of different datasets. In total, 52 datasets were found that met the authors’ criteria. To the best of the authors’ knowledge, this list is the most complete database of construction-related 3D datasets for computer vision and scene understanding. A tree structure is presented in Figure 2 to visualize the extent of combinations in the wider scope of 3D scene understanding. The tree structure shows how certain sensors, data types, and data formats are commonly paired for given use cases in the literature. The main scene understanding tasks are arranged as use cases on top of the tree. The layers represent certain objectives, attributes, and methods that we extracted from the reviewed datasets. The connecting edges show the configurations found within the papers in the following Table 1. This table includes a comprehensive compilation of a wide range of freely available datasets that are relevant for different 3D point cloud scene understanding tasks in the scope of the construction industry. We categorized the available data by its binary data type of acquisition {*real-word*; *synthetic*}, the content of the dataset {*scene type*}, utilized hardware {*sensors*}, the dataset file format {*representation*}, the intended use case {*application task*}, and the available point cloud features {*features*}. In addition, the number of used semantic annotation classes is given.



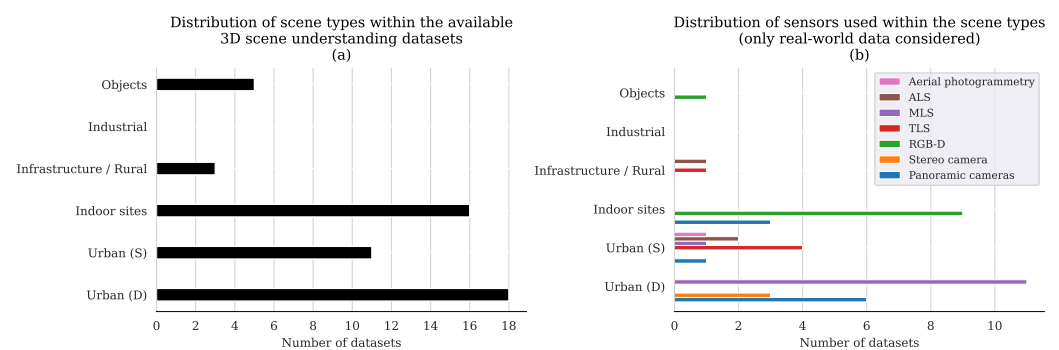
Scene Type	Sensors	Representation	Application Task	Features	Nr
------------	---------	----------------	------------------	----------	----

Year	Name	Reference	Data Type	Objects	Indoor	Urban (static)	Urban (driving)	Industrial	Infrastructure	Panorama camera	Stereo camera	RGB-D	TLS	MLS	ALS	Aerial photogrammetry	IMU	GPS	RGB sequence	Depth sequence	Point cloud	3D model	Object detection	Pose estimation	Shape classification	Object tracking	Semantic segmentation	Instance sem. segment.	Point cloud registration	Scene reconstruction	Surface reconstruction	Volume reconstruction	Normal	Mesh	Intensity	RGB	SLAM	# Sem. classes	
2009	Oakland 3-D	[41]	R			X							X								X						X											5	
2011	Ford Campus Vision and Lidar Data Set	[42]	R				X			X				X			X	X			X						X											N/A	
2012	KITTI stereo evaluation 2012	[43]	R				X				X			X				X								X		X										8	
2013	NYUv2	[44]	R		X							X					X	X	X	X						X		X										14	
2013	SUN3D	[45]	R		X						X								X	X							X											N/A	
2013	Sydney Urban Objects	[46]	R	X								X		X							X				X		X											14	
2014	Paris-rue-Madame database	[47]	R				X							X							X					X		X							X			17	
2015	iQmulus	[48]	R				X							X							X					X		X										8	
2015	NCTL Dataset	[49]	R		X	X				X				X			X	X	X		X					X		X										N/A	
2015	SceneNet	[50]	S		X									X								X			X													270	
2015	ShapeNet	[51]	S	X										X								X			X		X		X									53	
2015	SUN RGB-D	[52]	R		X							X							X	X				X	X		X		X									100	
2016	ObjectNet3D	[53]	S	X																				X		X		X											
2016	Oxford RobotCar	[54]	R				X							X			X	X			X							X											
2016	SceneNet RGB-D	[55]	S		X							X							X	X		X				X		X										N/A	
2016	SceneNN	[56]	R		X							X								X	X					X		X										40	
2017	2D-3D-S Dataset	[31]	R		X					X		X							X	X	X					X		X										13	
2017	Matterport3D	[57]	R		X					X		X							X	X	X					X		X										40	
2017	Redwood	[58]	R		X							X							X	X	X					X		X											
2017	S3DIS	[59]	R		X					X		X								X	X						X		X									13	
2017	Semantic3D	[60]	R			X							X								X					X		X										9	
2018	Paris-Lille-3D	[61]	R			X							X					X		X	X					X		X										50	
2018	ScanNet V2	[62]	R		X							X						X		X	X				X		X		X									21	
2018	WHU-TLS dataset	[63]	R			X			X				X								X								X									7	
2019	A*3D	[64]	R				X			X				X					X	X	X																	17	
2019	Agroverse1	[65]	R				X			X	X			X				X	X	X	X					X		X										25	
2019	ApolloScape	[66]	R				X			X	X			X				X	X	X	X					X		X										3	
2019	BLVD	[67]	R				X				X			X				X	X	X	X					X		X										8	
2019	H3D	[68]	R				X			X				X			X	X	X	X	X					X		X										3	
2019	Lyft Level 5	[69]	R							X				X					X	X	X					X		X										24	
2019	PartNet	[70]	S	X										X					X	X	X					X		X										2	
2019	PreSIL	[71]	S				X							X					X	X						X		X										88	
2019	Replica	[72]	R		X									X					X	X	X					X		X										15	
2019	ScanObjectNN	[73]	R	X										X							X					X		X										28	
2019	SemanticKITTI	[74]	R				X							X							X					X		X										40	
2019	Structured3D	[75]	S		X					X		X		X					X	X						X		X										9	
2019	SynthCity	[76]	S				X							X							X					X		X										24	
2020	Campus3D	[77]	R			X										X					X						X		X									8	
2020	DALES	[78]	R						X												X						X		X									32	
2020	nuScenes-lidarseg	[37]	R				X			X				X				X	X		X					X		X										8	
2020	Toronto-3D	[40]	R				X			X				X				X	X		X					X		X										8	
2020	Waymo Open	[39]	R				X			X				X							X						X		X										23
2021	3D-FRONT	[79]	S		X																X					X		X											
2021	Agroverse2	[36]	R							X	X			X				X	X		X					X												30	
2021	BuildingNet	[80]	S				X							X							X						X											31	
2021	ONCE	[38]	R				X							X				X	X		X																	5	
2021	Paris-CARLA-3D	[81]	R			X				X				X			X	X			X					X		X											23
2021	Paris-CARLA-3D	[81]	S			X								X				X	X		X					X		X											23
2021	PSNet5	[19]	R						X												X					X		X											5
2022	LiSurveying	[82]	R			X								X		X					X					X		X											54
2022	VASAD	[22]	S		X									X							X					X		X											11

The presented datasets consider different scene types and were recorded with different sensors to capture 3D spaces. A quantitative evaluation of the look-up table for scene types and sensors is given in Figure 3. Sub-figure (a) shows the occurrence frequency of each scene type in our database. Sub-figure (b) breaks down which sensors were used to capture the scene types. As shown, some included datasets feature dynamic urban scenes from autonomous driving (D). We considered these datasets for two reasons. On the one hand, urban driving data contain road infrastructure and building facades in the peripheral field of view of the car's sensors. On the other hand, autonomous driving is a leading force for scene understanding and Simultaneous Localization And Mapping (SLAM) methods, which implies that the latest algorithms are often developed and benchmarked with urban driving data. The SemanticKITTY dataset contains semantic classes of structural elements: roads, sidewalks, buildings, poles, road signs, and other structures besides the standard traffic unit classes, and became a benchmark for autonomous driving and outdoor SLAM algorithms [74]. The NCTL dataset consists of omnidirectional imagery, 3D LiDAR, planar LiDAR, GPS, and odometry of outdoor and indoor scenes from the University of Michigan campus, captured by an autonomous Segway robot [49]. Unfortunately, the point data are not annotated; therefore, the application for supervised machine learning is not possible without considerable effort to prepare the data.



**Figure 2.** A tree structure to summarize the variety of common dataset configurations for 3D scene understanding tasks. Digital zoom is recommended.



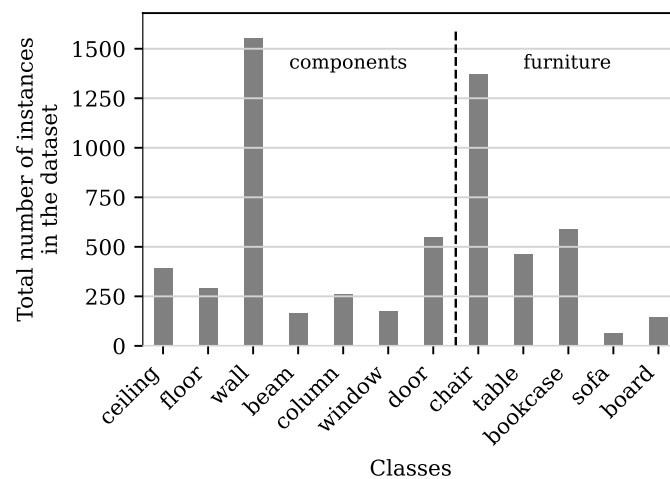
**Figure 3.** Distribution of scene-type variety and sensor hardware used to capture 52 famous 3D datasets for scene understanding from Table 1. Declaration: static sensor (S) position, dynamic sensor (D) position from a vehicle-mounted device, aerial laser scanning (ALS), mobile laser scanning (MLS), terrestrial laser scanning (TLS).

Besides autonomous driving, the second most covered spaces in the included datasets are residential indoor scenes. Most real-world indoor datasets were captured with RGB-D devices and were released between 2015 and 2017. This trend can be explained by the price drop of 3D imaging hardware in the mid-2010s, which caused a leap in the development of many successful semantic segmentation methods that use both RGB and depth features [56,83]. Consumer RGB-D cameras, like the Microsoft Kinect [84], are easy to use but not necessarily best suited for capturing large-scale spaces. Their small field of view requires stitching together many frames per scene in postprocessing, which often leads to registration errors and non-complete room geometry, as shown in Figure 4. Almost all indoor datasets have in common that they present fully furnished rooms of residential buildings and office spaces. In most cases, the class annotations focus on furniture besides the main building components (wall, floor, ceiling, door, and window). One of the first large indoor scene understanding datasets was the SUN RGB-D benchmark suite [52], which holds 10K RGB and depth images of residential rooms. Objects and room geometry is annotated in 2D by polygons and in 3D by its boundary boxes and orientation. SceneNN [56] holds comparable content, but the scenes are reconstructed into triangle meshes and possess per-vertex and per-pixel annotation. One of the most impactful 3D datasets in the indoor domain with point-wise semantic class annotations available is ScanNetV2 [62], now in its second generation. The dataset offers RGB-D video streams and 3D camera poses of 1.5K residential scenes captured with low-cost sensor setups and crowd-sourced instance-level semantic annotation. VASAD [22] is a synthetic volume and semantic architectural dataset composed of six buildings. The focus of VASAD is to improve semantic segmentation and volumetric reconstruction for BIM modeling. VASAD's semantic classes only consist of building components, and its authors introduce a method to automatically simulate terrestrial laser scanning (TLS) behavior by raytracing virtual scanlines from iteratively added viewpoints to sample point clouds from mesh-based CAD models. The Stanford 3D Indoor Scene Dataset S3DIS [59] stands out as the biggest fully annotated indoor point cloud dataset and is heavily used for training and benchmarking in semantic segmentation tasks [85]. The S3DIS sources from joint 2D–3D semantic data [31] and contains 6 large-scale indoor areas with 271 rooms. These areas show diverse architectural styles and appearances and mainly include office areas, educational and exhibition spaces, restrooms, open spaces, lobbies, stairways, and hallways. Each point in the scene point cloud is annotated with one of the 13 semantic classes shown in Figure 5. Besides five furniture classes, S3DIS contains seven classes (ceiling, floor, wall, column, beam, window, and door) relevant to the construction industry [59]. Because S3DIS is the only real-world dataset of this size and holds a reasonable number of classes to classify structures of buildings, we adopted the S3DIS dataset as our reference dataset in the following sections.





**Figure 4.** Examples of poor registration in 3D reconstruction from depth frames in the SceneNN dataset [56]. (a) Misalignment of the ceiling in two point clouds, highlighted by the red circle. (b) Deviation from the ground plane and large parts missing from the wall behind the desk. Highlighted by the two red lines and the opening angle, the left part of the room tilts, and the floor is not level.



**Figure 5.** Illustrative class–instance distribution in the S3DIS semantics dataset considering all six areas [59]. The pictured distribution is representative of institutional, educational, and office areas. Other indoor datasets may deviate depending on the use of space.

In the presented datasets, no content of the scene type industrial can be found and only four datasets include scenes showing infrastructure objects. A few publications exist about deep learning approaches for semantic segmentation of bridge components and industrial environments [20,23,24], but at the time of this publication, only RC Bridges [86], which contains ten reinforced concrete bridges in the area around Cambridge, is freely available. This dataset is intended to cluster point clouds into the listed bridge parts *slab*, *pier*, *pier cap*, and *girder*, but Ruodan LU et al. [86] did not publish ground truth semantic annotations. The only two sources featuring semantically annotated point clouds of infrastructure content have low quality, representing urban areas captured from an aerial view with sparse point density and low Level of Detail (LoD) [19,78].

Object detection on 3D data is becoming increasingly relevant for the construction industry in terms of detecting building parts (e.g., doors, railings, pipes, rebars, and built-in parts) in the context of BIM but is also crucial for autonomous robots on construction sites to detect interactive objects and obstacles. The available datasets predominantly feature models of furniture and decoration from residential buildings [70,73]. ShapeNet, the most famous 3D-object dataset, holds a collection of human-made objects divided into 270 categories, including vehicles, clothing, weapons, and household equipment, but only a few construction elements, like towers [51].

### 3.1. Dataset Quality

A backlog in data quality (Figure 4), caused by poor sensor technologies used in the process of capturing indoor/building datasets, is a relevant issue for industrial application and research. Most indoor datasets originate between 2015 and 2017, while most large-scale outdoor datasets originate from the period between 2019 and today. This observation correlates with the finding in Figure 3b, where indoor datasets tend to be captured on RGB-D devices, while most urban datasets rely on laser scanning technology. Sensor arrays and stationary rotating cameras help with registration accuracy and scene completeness as they produce 360° panoramic depth images like in the S3DIS dataset [59], but they still suffer from higher measuring inaccuracy if compared to Light Detection And Ranging (LiDAR) (manufacturer's information Matterport: precision of  $\pm 1\%$  within max. 4.5 m range [87]). Most outdoor datasets use costly mobile or terrestrial laser scanning technologies in combination with synchronized panoramic cameras to color the point clouds, Global Positioning System (GPS) for georeferencing, and inertial measurement units (IMUs) for motion correction. The industry standard rotating LiDAR sensors for mobile laser scanning (MLS) applications from companies like Velodyne Inc. and Ouster Inc. claim range accuracy down to  $\pm 0.5$  cm within 35 m (down to  $\pm 2.0$  cm within max. 200 m) range [88]. Laser sensors for TLS applications from companies like Leica Geosystems AG sell laser scanners with 3D accuracy down to  $\pm 3.2$  mm within 35 m ( $\pm 5.6$  mm within 100 m) range [89]. By utilizing laser scanners and multi-sensor-fusion, the latest urban datasets allow significantly higher data quality [37,82].

### 3.2. Why Class Imbalance Is a Problem for Deep Learning

The availability of datasets that are designed for semantic segmentation tasks, applicable in the construction industry, and acquired from real-world scenes are limited and often characterized by a lack of object class balance. To achieve optimal performance and avoid overfitting on individual classes, deep learning models must be trained on training sets that are as balanced as possible. The evaluation of the well-adopted S3DIS dataset [31] in Figure 5 shows that, in the real-world example, wall instances are heavily over-represented in common indoor scenes. In the case of educational and office areas, a predominant amount of site-specific furniture can be found. In the S3DIS example, chairs dominate the object classes. A deeper examination of the per-point level in Figure 6 reveals that chair instances often appear in the dataset but occupy only a small amount of data points. The situation is opposite for both the structural elements ceiling and floor, as these elements are single connected large segments with a wide surface area. Comparing the number of scan points by their semantic labels shows that only three classes (*wall*, *ceiling*, and *floor*) occupy about 63% of the training data (85% of the building component objects), while beam and column, but also window and door classes, are marginalized.

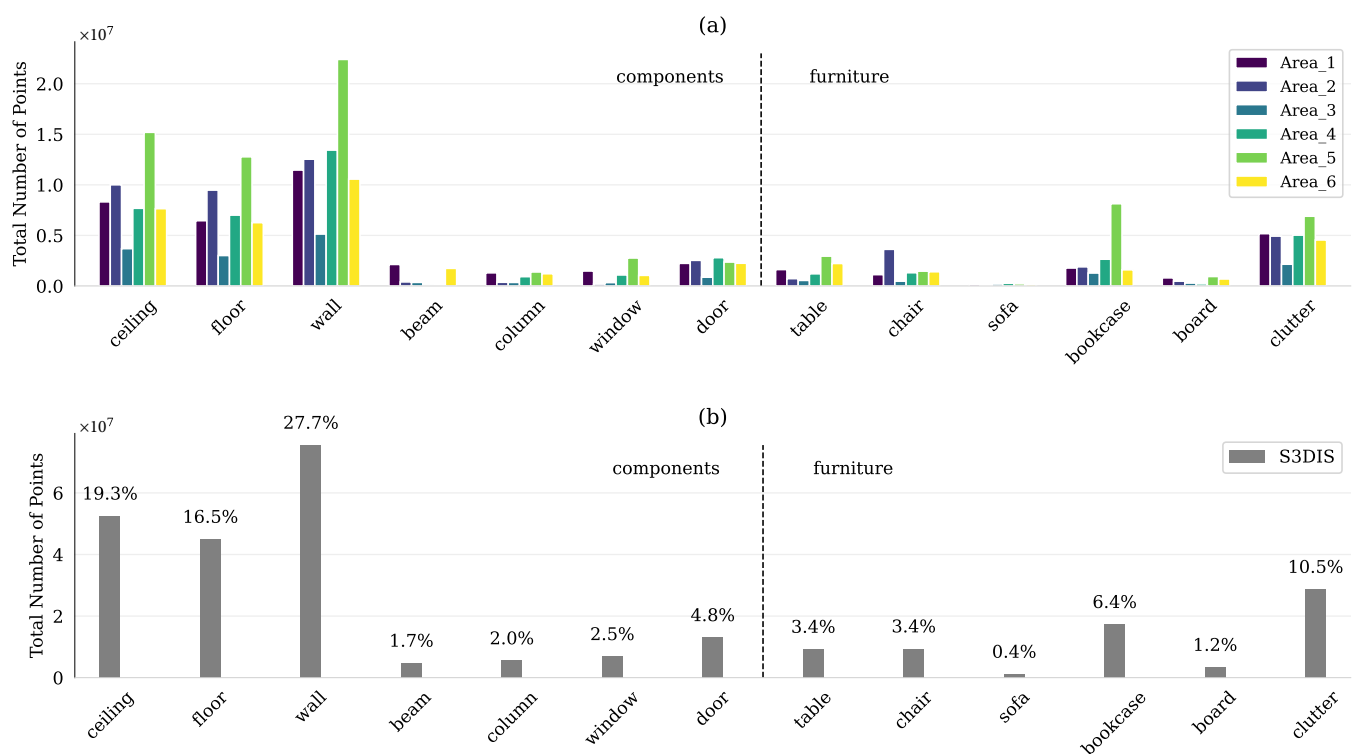
If a class imbalance exists within training data, learners typically over-classify the majority group due to its increased prior probability. As a result, the instances belonging to the minority group are misclassified more often than those belonging to the majority group [90]. Dataset imbalances of this form are commonly referred to [91] as follows:

*Intrinsic imbalance*, which is caused by the naturally occurring frequency of data, e.g., rooms usually have four walls but only one door and measurable damage in sensor data streams only occur in very rare catastrophic events.

*Extrinsic imbalance*, which is caused by external factors, e.g., ceilings did not get scanned due to camera viewing angles, and most buildings' architecture is biased by a country's certain style.

The mechanism behind the poor network adoption for subordinate classes can be explained by Anand et al. [92]. Within the training process of feedforward networks, they could show that the backpropagation (BP) gradient vector lengths are proportional to the sizes of the training sets. In standard BP, the sum of all gradient vectors corresponds to the direction to follow to change perceptron weights. In imbalanced scenarios, the

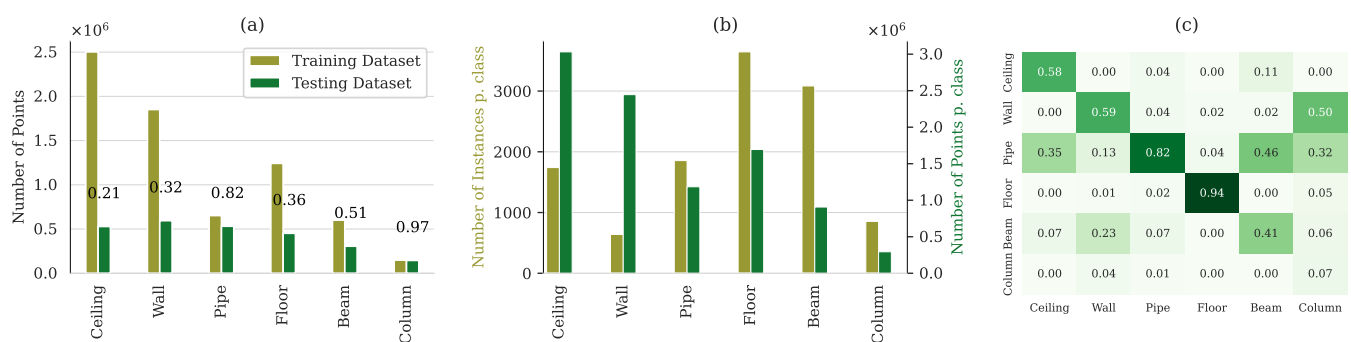
majority class gradients dominate the net gradient vector, which can result in a moving direction pointing upwards for minority classes. Consequently, standard BP will make major improvements in reducing the net error in the first steps and will likely get stuck in a slow mode of error reduction. Besides poor convergence behavior, class imbalance can lead to further problems. In the case of strong object variation, it can happen that the model does not see enough data on the underrepresented class to classify this class correctly, even if the dataset seems extensive and detailed. For performance evaluation, some metrics, such as accuracy, might mislead the analyst with high scores that incorrectly indicate good performance [90]. Finally, class imbalance makes it difficult to create appropriate validation sets if the sample pool to choose from is small.



**Figure 6.** Quantitative evaluation of the per-point semantic class distribution of the S3DIS (2D–3D semantic dataset) [31]. (a) Per-point per-scene class distribution. (b) Classes allocation of the total S3DIS dataset. The S3DIS dataset is partitioned into six subsets with diverse properties in architectural style and appearance. Area\_1 to Area\_6 represent the six subsets.

Nevertheless, creating well-balanced datasets is not just the challenge of balancing the numbers. Japkowicz [93] concluded, through an experiment with an artificial dataset, that a system's sensitivity to training data imbalance increases with the degree of complexity of the system and that non-complex linear separable domains do not appear sensitive to any amount of imbalance. This entails for the use case of semantic segmenting building elements that distinguishing between a vertical wall and a horizontal floor is easier for a neural network than distinguishing between a vertical cylindrical *pipe* and a cylindrical *column* because more analogous classes add extra complexity to the model. To illustrate this, we refer to the segmentation results published by Yeritza Perez-Perez et al. [20], where point cloud data of industrial areas containing plenty of mechanical/electrical/plumbing (MEP) building systems were segmented with the help of a support vector machine (SVM) model. The confusion matrix in Figure 7c reveals that *floors* and *pipes* get classified with high precision, while *columns* and *beams* suffer from poor classification accuracy. These results correlate with the available data from the used dataset represented in Figure 7a, where a strong discrepancy in the training/test split is recognizable due to poor class distribution in

real-world scenes, which, in turn, forces the analyst to an imbalanced data split. In practice, this effect is further amplified by the observation that the degree of complexity is inverse to the likelihood of occurrence in the real-world dataset. Scan points of object classes with a low level of complexity (e.g., *floors*) appear more frequently than object classes with a high level of complexity (e.g., *columns*). The confusion matrix representations show the extent to which misclassification impacts the prediction result. In the example by Yeritza Perez-Perez et al. [20], the model prefers to classify vertical *columns* as geometrically similar objects, of which it has seen more data in training, like *walls* and *pipes*. This dataset holds a large number of *beam* instances, but with a low number of points assigned, so that the model's classification decision becomes polluted by similar vertical objects, like *pipes*. In general, learners will typically over-classify the majority group due to its increased prior probability. As a consequence, the instances belonging to the minority group are misclassified more often [90].

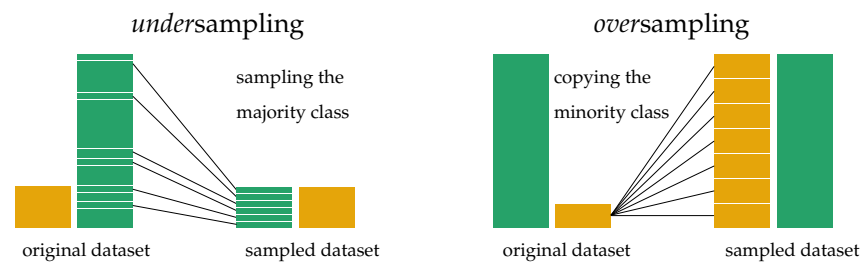


**Figure 7.** Quantitative evaluation of the Perez-Perez dataset configuration [20]. (a) Test and training data split partitioned by semantic classes. Numbers show the ratio between the testing and training split. (b) Class distribution available in the dataset comparing instances vs. raw per-point segments. (c) Exemplary confusion matrix to represent the average precision of semantic labeling per segment from a support vector machine (SVM) model. Axis flipped for the unified convention. Published by Yeritza Perez-Perez et al. [20].

### 3.3. Solutions to Handle Dataset Imbalance

With the knowledge that class imbalance affects learning performance, application-oriented solutions can be discussed. Different techniques for classical machine learning have been published to compensate for the immense hindering effects that class imbalance has on standard learning algorithms. These techniques generally approach the problem from one of two sides. They either try to change the condition of the class composition on the dataset side (active) or keep the dataset unchanged and adjust training or inference on the algorithm side (passive). Techniques that combine data-based and algorithm-based adjustments can be summarized as hybrid approaches [90].

Active techniques (basic concept depicted in Figure 8), which change the class distribution, include oversampling the minority classes, undersampling the majority classes, and a combination of the two methods. A collection of frequently used sampling techniques and their approach to the imbalance problem is given in Table 2. These techniques alter the dataset to make standard training algorithms work. Active sampling techniques are well developed. A detailed review of various oversampling and undersampling techniques is given by [91,94]. Lemaitre et al. [95] published an open-source Python toolbox including a wide range of state-of-the-art methods to cope with the problem of imbalanced datasets.



**Figure 8.** Graphical representation of the two principles of under- and oversampling for imbalanced training data handling in machine learning applications.

**Table 2.** Most commonly used undersampling and oversampling methods in standard classification tasks [90,91,95,96].

	Name	Approach
Undersampling	Random Undersampling	Randomly selects a subset of the majority class and discards the rest.
	Tomek Links	Removes samples from the majority class that are close to samples in the minority class.
	Near Miss	Selects a subset of the majority class such that the samples are close to the decision boundary between the minority and majority classes.
	One-Sided Selection	Selects a subset of the majority class such that the samples are more dissimilar to the minority class than the majority class samples that are not selected.
	Cluster Centroids	Creates clusters of the majority class and replaces each cluster with its centroid.
	Condensed Nearest Neighbor	Selects a subset of the majority class such that each sample in the subset has at least one majority class sample that is not its nearest neighbor.
	Edited Nearest Neighbor	Iteratively removes samples from the majority class such that the remaining samples are all "agreeable" with the minority class. Agreeable examples are those that are not misclassified by their nearest neighbor classifier.
Oversampling	Random Oversampling	Randomly selects samples from the minority class and duplicates them in the dataset.
	Synthetic Minority Oversampling Technique (SMOTE)	Generates synthetic samples for the minority class by interpolating between existing samples.
	Adaptive Synthetic Sampling (ADASYN)	Similar to SMOTE, but generates more synthetic samples for the minority class in regions where the classifier is less accurate.
	Borderline Oversampling	Generates synthetic samples for the minority class that are close to the decision boundary between the minority and majority classes.
	Backpropagation-Based Oversampling	Generates synthetic samples for the minority class using back-propagation to learn a transformation from the majority class to the minority class.

**Undersampling** the majority class is a common first choice as this can help reduce the overall storage size and training time, which can be beneficial for large datasets and slow classifiers. However, undersampling carries the inherent risk of losing potentially useful information from the majority class and the risk of creating biased data selection. The sample might not accurately represent the real world, causing the model to output



inaccurate results. Random undersampling of the majority class (Figure 8, left) is easy to implement and fast to execute. At the same time, it is associated with an uncontrollable risk of information loss. To address this limitation, more advanced techniques have been developed that leverage meta-information, like the distance between two samples, neighborhood knowledge, clustering, and even evolutionary algorithms, like simulating bird flocking behavior [96]. Generating more instances of the underrepresented class through **oversampling** (Figure 8, right) does not lead to information loss, which makes it a popular technique for balancing imbalanced datasets [97]. Applying oversampling techniques to the minority class can enhance the performance of a classifier on that particular class. The downside of oversampling is opaque. By replicating synthetic instances of the original dataset, the likelihood of creating random noise and causing class-overfitting increases. Overfitting, in particular, occurs when classifiers produce multiple clauses in a rule for multiple copies of the same example, which causes the rule to become overly specific. Although the training accuracy will be high in this scenario, the classification performance on the unseen testing data is generally far worse [91,96]. Oversampling for multi-class datasets potentially produces large amounts of data, which is storage-intensive and time-consuming to train with. It is important to note that oversampling before splitting the data into training/testing sets can lead to identical copies in both sets. This causes overfitting and poor generalization. Therefore, splitting the data before applying oversampling techniques is mandatory to ensure accurate evaluation and generalization of the model. **Combined over- and undersampling** can be a solution to the overfitting problem. Gustavo E. A. P. A. Batista et al. [98] combined SMOTE oversampling with Tomek links undersampling as a cleaning technique by removing overlapping samples in the training set to improve classification in a binary case. However, active sampling may not be the most effective approach to deal with imbalanced 3D point cloud data. Sampling techniques have proven to be effective for binary datasets [97] and datasets with a small number of features [99], but point clouds typically hold multiple features (e.g., X, Y, and Z-coordinates and r, g, and b colors for each point), which can make it difficult to pick meaningful samples or to determine which samples to discard.

Passive methods, which keep the dataset unchanged, include penalizing misclassification through **cost-sensitive learning**, adjusting the **decision threshold** of the classifier, and changing the **performance metric** to help the learner deal with the imbalance without the need to change the training dataset configuration. Accuracy and error rates are easy to implement metrics to evaluate the performance of multi-class classifiers and are frequently used in point cloud classification [17,100,101]. However, using singular assessment criteria does not provide adequate information, particularly in the context of imbalanced datasets [102], where the classifier prefers to ignore minority classes in favor of high accuracy on the majority classes. An exemplary case can be found in the situation when the majority class represents 99% of all cases, and the classifier simply assigns the label of the majority class to all test cases. An excellent but misleading accuracy of 99% will be assigned to a classifier that classified no minor cases right [103].

More informative assessment metrics are necessary for conclusive evaluations of performance in the presence of imbalanced data. In general, the machine learning literature often proposes the following evaluation metrics: receiver operating characteristics (ROC) curves, the area under the ROC curve (AUC), precision–recall (PR) curves, and cost curves, to evaluate the performance of imbalanced data classification from the confusion matrix [91]. However, these graphical methods are designed for binary classification and cannot be applied to multi-class classification problems. In the case of multi-class classification, the confusion matrix must be binarized. This can be conducted in two different ways: the *One-vs-Rest scheme* compares each class against all the others (assumed as one) and the *One-vs-One scheme* compares every unique pairwise combination of all classes [104]. State-of-the-art benchmarks use multi-class-suited metrics to compare performance in 3D point cloud tasks in practice. The most relevant ones are listed in Table 3 and application-oriented

performance metrics will be discussed in Section 4. Finally, a confusion matrix (binary case in Figure 9 and multi-class in Figure 7c) is a powerful tool to visualize classification results.

**Table 3.** Summary of frequently used evaluation metrics for 3D point cloud scene understanding tasks. Overall accuracy (OA), mean overall accuracy (mAcc), mean Intersection over Union (mIoU), mean average precision (mAP), average precision scores with IoU thresholds set to 25% and 50% (mAp@25 and mAP@50, respectively), mean precision (mPrec) and mean recall (mRec).

Task	Metric	Ref. Example
3D Point Cloud Classification	OA, mAcc	[105,106]
3D Semantic Segmentation	OA, mAcc, mIoU	[107,108]
3D Instance Segmentation	mAP, mAP@25, mAP@50, mPrec, mRec	[109,110]

**Cost-sensitive learning** considers the cost of prediction errors and assigns penalties to each class through a cost matrix. Increasing the cost of the minority group is equivalent to increasing its importance and decreases the likelihood that the learner will misclassify instances from this group [90,111]. A compilation of cost-sensitive learning approaches for neural networks is given by Buda et al. [103]. While implementing cost-sensitive learning into the training algorithm is a comparatively easy task and most common algorithms are modified to take a class penalty or weight into consideration, one of the biggest challenges in cost-sensitive learning is the assignment of an effective cost matrix [90]. The cost matrix can be defined empirically based on past experiences or by a domain expert with knowledge of the problem. However, the structure of multi-class datasets is not definite, and the relations among classes are not always obvious. For example, one class might be a majority compared to one other class, but a minority or well balanced for the rest of them [111]. Another way is to set the false negative prediction cost to a fixed value and vary only the false positive cost. The ideal cost matrix is identified using a validation set. This has the advantage of exploring a range of costs but can be expensive and even impractical if the dataset size or number of features is too large [90].

		Real (Ground Truth)		
		Positive	Negative	
Prediction	Positive	True Positive (TP)	False Positive (FP)	PP
	Negative	False Negative (FN)	True Negative (TN)	PN
		RP	RN	N

**Figure 9.** Confusion matrix for binary testing. Note: Other references may use a different convention for axes. Declaration: real positive (RP), real negative (RN), predicted positive (PP), predicted negative (PN).

**Thresholding methods** adjust the decision threshold of a classifier to convert a predicted probability or scoring into a singular class label. In the context of a binary classification problem with class labels 0 and 1, a common approach is to utilize normalized predicted probabilities and apply a threshold of 0.5. Values below the threshold are attributed to class 0, while values equal to or above the threshold are assigned to class 1. The presence of imbalanced data or low acceptance of *false* predictions poses a challenge, as the default threshold may not accurately reflect the optimal interpretation of predicted probabilities. Adjusting the threshold through hyperparameter tuning can enhance the classifier recall in a binary classification scenario. Nonetheless, multi-class classification, as typically encountered in point cloud classification, where each scan point can carry a single class label, cannot rely extensively on thresholding methods. Instead, class labels are

typically assigned based on the highest predicted probability. In conclusion, it is noteworthy that the solutions discussed for addressing dataset imbalance are representative of a broad range of techniques in machine learning. Johnson and Khoshgoftaar [90] suggest that the current research on the use of deep learning to tackle class imbalance in non-image data is limited and should be further explored. Since point cloud data are a novel research domain, they pose a unique challenge for deep learning methods, and therefore, there is still limited progress in developing effective solutions.

#### 4. Evaluation Metrics

Understanding common evaluation schemes and the metrics to measure performance is necessary for developing and applying deep learning methods. Although the overall accuracy (OA) is a significant index on a minimum unit basis (e.g., pixel, point, voxel, and mesh), it is not a well-defined metric to evaluate model predictions. Accuracy assigns equal cost to false positive and false negative cases, which, in reality, is rare. For example, engineering is known for high safety requirements where false negative predictions can have a crucial impact, while false positives may be neglected. Binary test results can be best visualized in a confusion matrix, as shown in Figure 9. Within the confusion matrix, the capital letters TP/FP refer to the number of positive predictions that are true/false-classified, and similarly, FN/TN refer to the number of negative predictions that are false/true-classified. The sum,  $N$ , of all four cells is equal to the total number of classified points. The capital letters PP/PN correspond to the horizontal sum of positive/negative predictions, and vice versa RP/RN correspond to the vertical sum of positive/negative real units. By convention, the lowercase letters,  $tp$ ,  $fp$ ,  $fn$ ,  $tn$  and  $rp$ ,  $rn$  and  $pp$ ,  $pn$ , refer to the joint and marginal probabilities, and the four contingency cells and the two pairs of marginal probabilities ( $rp + rn = 1$  and  $pp + pn = 1$ ) each must sum to 1 [112]. Figure 7c shows an example of a confusion matrix for multi-class classification. In the multi-class case, FP, FN, and TN cannot be obtained directly from the confusion matrix as in the binary case; instead, they must be determined individually for each class according to the summation scheme in Figure 10.

		Real (Ground Truth)					
Prediction	Column	Ceiling	Wall	Pipe	Floor	Beam	Column
		TN	FN	0.35	0.00	TN 0.07	0.00
		FP	TP	0.13	False Positives		0.04
		0.04	0.04	0.82	0.02	0.07	0.01
		0.00	0.00	0.04	0.94	0.00	0.00
		TN	FN	0.11	0.46	0.00	0.41
Column	Beam	0.00	0.00	0.32	0.05	0.06	0.07
		0.00	0.00	0.32	0.05	0.06	0.07

**Figure 10.** Multi-class confusion matrix summation scheme to determine the following: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). An example is shown for the class *wall*. Note: other references may use a different convention for axes.

Common statistical performance measures can be calculated from the evaluation of a binary classification problem and the per-class discretization. Precision (also known as confidence) denotes the proportion of predicted positive cases that are correctly real positives. Recall (also known as sensitivity) aims to identify all real positive (often called relevant) cases. The F1-score denotes the weighted average (harmonic mean) of precision and recall and is a measure of a test's fidelity, which is usually more useful than accuracy since it requires both precision and recall to have a higher value at the same time for the f1-score to rise, especially if dealing with uneven class distributions [113,114]. The formulas

to calculate precision, recall, and the F1-Score are reported in Equations (1), (2), and (3), respectively:

$$\text{Precision} = \frac{TP}{(TP + FP)} = \frac{TP}{PP} \quad (1)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} = \frac{TP}{RP} \quad (2)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Based on the confusion matrix kernel, different evaluation metrics were proposed to test the performance of various point cloud understanding tasks, not only in the binary scenario but also in multi-class cases. We assume a set of  $K + 1$  classes, where  $p_{ij}$  denotes the smallest instance (e.g., pixel, voxel, mesh, and point) of class  $i$  implied to belong to class  $j$  ( $i = j$  represents *true* and  $i \neq j$  represents *false* classifications). To generalize the implementation, matrix indexing is applied so that  $p_{ii}$  and  $p_{jj}$  represent *true positives* and *true negatives*, while  $p_{ij}$  and  $p_{ji}$  represent *false positives* and *false negatives*, respectively [26]. The formulas to calculate the overall accuracy (OA) and the mean class accuracy (mAcc) are reported in Equations (4) and (5), respectively.

OA (sometimes abbreviated oAcc) measures the overall effectiveness of a classifier by computing the ratio between the number of truly classified samples and the total number of samples.

$$\text{OA} = \frac{\sum_{i=0}^K p_{ii}}{\sum_{i=0}^K \sum_{j=0}^K p_{ij}} \quad (4)$$

mAcc measures the average per-class effectiveness by computing the OA per class and averaging by the total amount of  $K$  [115].


$$\text{mAcc} = \frac{1}{K+1} \sum_{i=0}^K \frac{p_{ii}}{\sum_{j=0}^K p_{ij}} \quad (5)$$

OA and mAcc are commonly used as performance metrics for point cloud classification [17,100,101]. However, for reasons we already explained in the last section (Section 3.3), both are hardly meaningful in the presence of imbalanced classes and to evaluate segmentation results. Accounting for the area of segment overlap improves the informative value. The formula to calculate the mean Intersection over Union (mIoU) is reported in Equation (6).

mIoU, as shown in Figure 11, computes the intersection ratio between ground truth and predicted values per class and averages the sum over the total number of classes  $K$  [116].

$$\text{mIoU} = \frac{1}{K+1} \sum_{i=0}^K \frac{p_{ii}}{(\sum_{j=0}^K p_{ij} + p_{ji}) - p_{ii}} \quad (6)$$

mIoU and mAcc are the most frequently used performance metrics for 3D point cloud semantic segmentation [40,59,74], assuming  $L_I, I \in [0, K]$  is the number of instances in every class and  $c_{ij}$  is the number of points of instance  $i$  inferred to belong to instance  $j$  ( $i = j$  represents correct and  $i \neq j$  represents incorrect segmentation) [26]. The formulas to calculate the average precision (AP) and the mean class average precision (mAP) are reported in Equations (7) and (8), respectively.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


**Figure 11.** Visualization of the Intersection over Union (IoU).

AP is often used as the metric for 3D object detection [117]. It is calculated class-wise as the area under the precision–recall curve.

$$AP = \sum_{I=0}^K \sum_{i=0}^{L_I} \frac{c_{ii}}{c_{ii} + \sum_{j=0}^{L_I} c_{ij}} \quad (7)$$

mAP is frequently used for 3D object detection and 3D instance segmentation if discriminating between multiple instances within one class. It is an extension of AP by averaging the per-class precision over the total number of  $K$  [26,109].

$$mAP = \frac{1}{K+1} \sum_{I=0}^K \sum_{i=0}^{L_I} \frac{c_{ii}}{c_{ii} + \sum_{j=0}^{L_I} c_{ij}} \quad (8)$$

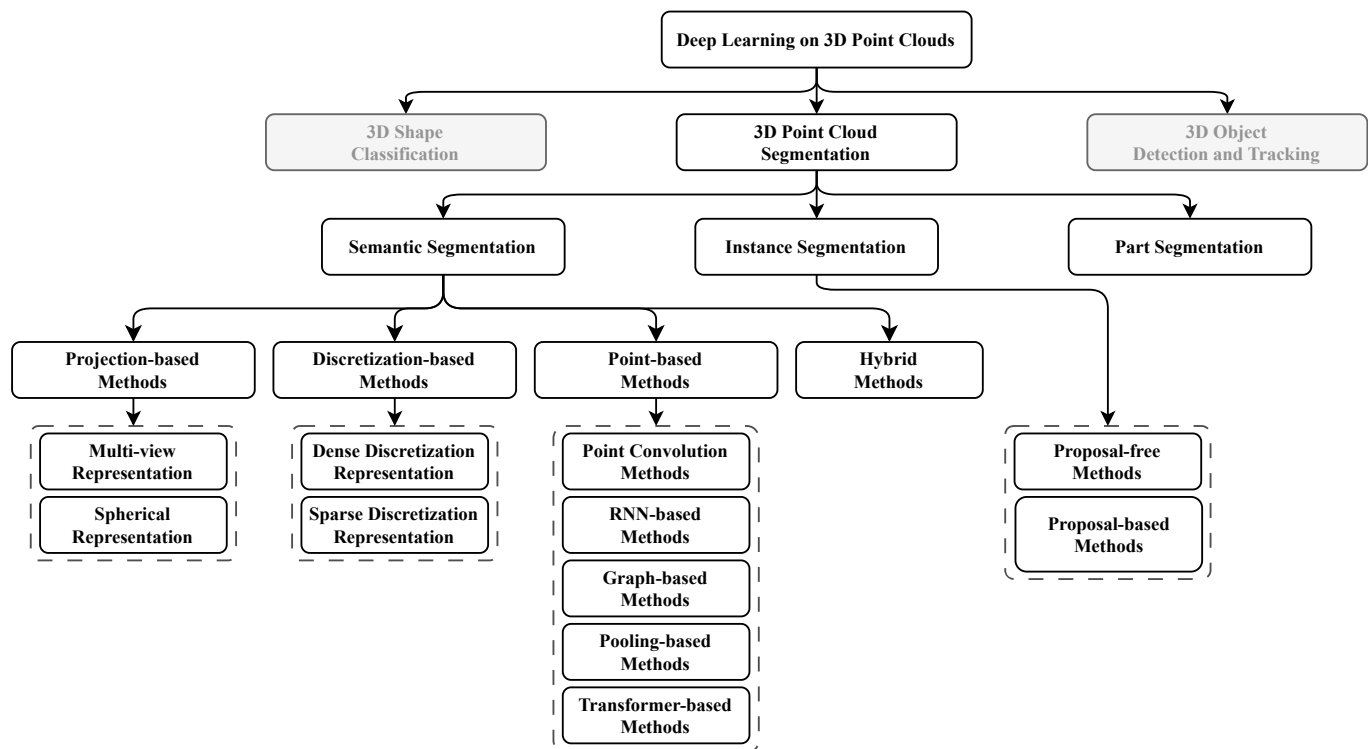
Apart from these commonly used metrics in machine learning, other problem-specific metrics are worth mentioning: Overall average category Intersection over Union (Cat.mIoU) and overall average instance Intersection over Union (Ins.mIoU) can be used for 3D part segmentation [26]. Precision and success are commonly used to evaluate the overall performance of 3D single-object tracking. Average multi-object tracking accuracy (AMOTA) and average multi-object tracking precision are the most frequently used metrics for 3D multi-object tracking [25,37].

## 5. Deep-Learning-Based Methods for Point Cloud Understanding

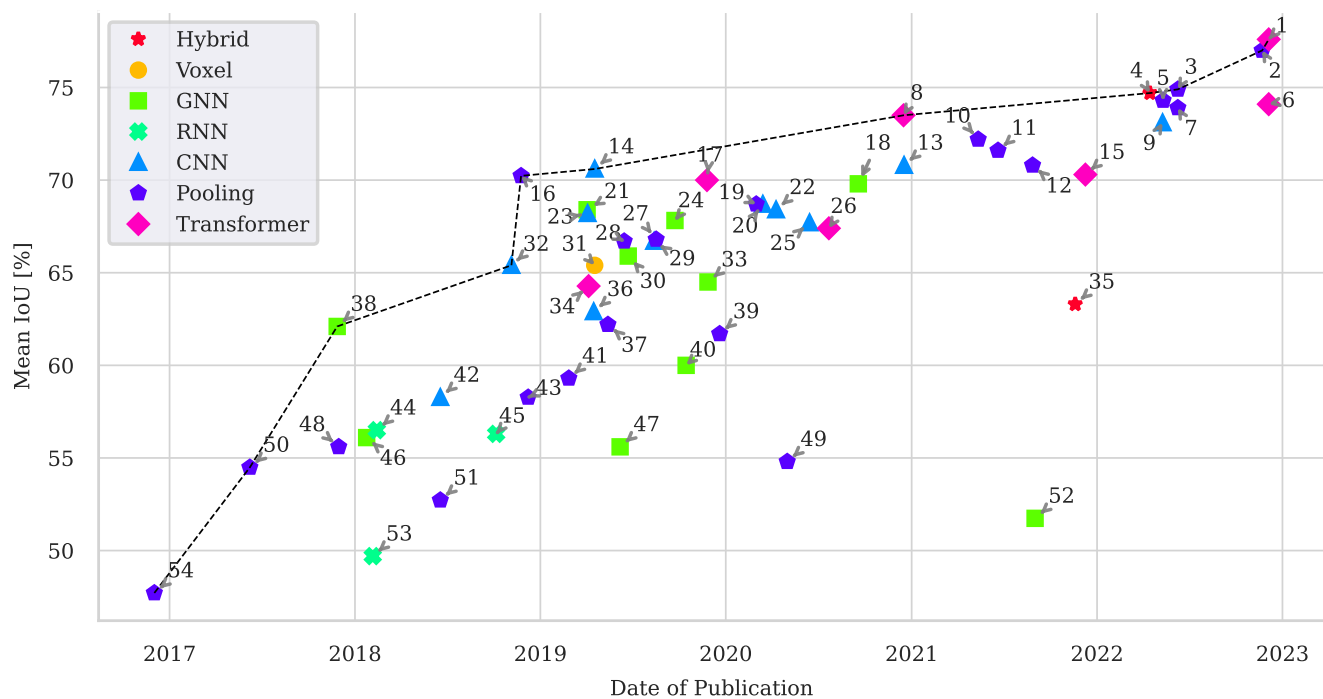
Deep learning for computer vision in the 3D domain has garnered significant attention and interest in recent years, particularly since 2015, when various approaches emerged to extend the application of 2D convolutions for image object recognition to the 3D model space [118–120]. However, the transition from 2D raster images to 3D point clouds presents unique challenges due to the unordered and unstructured nature of point cloud data [121]. Different preprocessing steps are required to transform the raw point cloud data into a structured format that can be processed by a neural network. According to Guo et al. [25], common learning-based PCSS approaches can be divided into three baseline categories: projection-based, discretization-based, and point-based methods. We adopted the taxonomy for 3D point cloud segmentation in Figure 12 and expanded them with the latest achievements in point-based methods. The choice to apply a particular type or a hybrid method depends on the problem to solve. In the following, we provide a concise overview to comprehend the basic characteristics of each method, followed by a discussion of the advantages and disadvantages of applying certain models in the construction industry, considering customary point density and cloud size for point clouds of building objects.

Projection-based and discretization-based methods share a common initial step. Both transform the unstructured point cloud into a regular intermediate representation to enable data handling. This regular representation is usually a  $d$ -dimensional grid  $d \in \mathbb{N}^+$  (e.g., image with  $d = 2$  and voxel with  $d = 3$ ), in which points can be referenced by indices, and the neighborhood relationship is defined by contiguity. Semantic segmentation is performed on the structured data in the intermediate space, and the result is subsequently projected back to the original, unstructured point cloud. In contrast to projection and discretization-based methods, point-based methods operate directly on irregular point clouds without using an intermediate representation [25]. We evaluate the publicly available semantic segmentation results of 54 methods on the S3DIS benchmark in Figure 13 on a timeline. For a performance comparison of the state of the art, we selected the mIoU as the most meaningful metric in the presence of class imbalance and the most adopted metric in public databases for PCSS tasks. The data for this comparison, secondary performance metrics (mIoU, mAcc, and oAcc), and an allocation of the deployed network architecture can be found in Table A1.





**Figure 12.** Taxonomy of deep learning methods for 3D point cloud segmentation. Adopted from [25] and extended with the latest transformer-based methods.

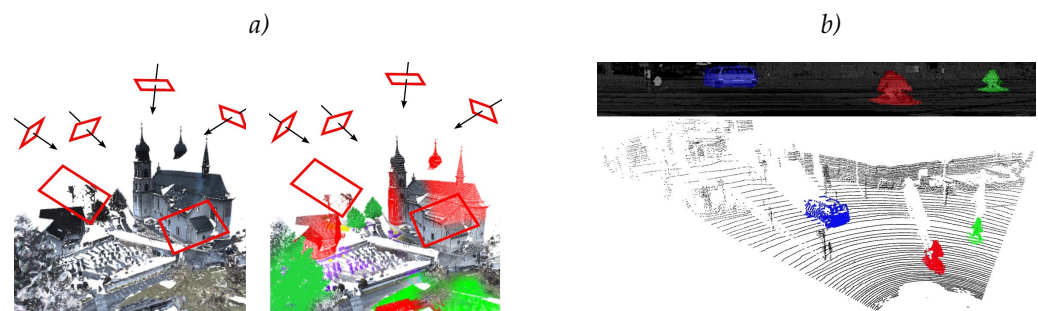


**Figure 13.** Reported results for semantic segmentation task on the large-scale indoor S3DIS benchmark (including all 6 areas, 6-fold cross-validation) [59]. Performance is expressed in terms of mean Intersection over Union (mIoU). Item numbers represent the ranking within this database according to mIoU performance. The data corresponds to the evaluation in Table A1.

### 5.1. Projection-Based Methods

Projection-based methods render raster image intermediate data from spatial 3D point clouds to enable the use of powerful 2D convolutional networks for semantic segmentation [3]. Several approaches have been proposed to project the 3D space onto 2D surfaces to cover large landscapes.

**Multi-view representation** approaches are among the earliest approaches for 3D data semantic segmentation. The limitations of 3D unstructured points can be avoided by taking snapshots of point cloud scenes with predefined image sizes, as shown in Figure 14a. These uniform, synthetic images can serve as input to any 2D CNN intended for semantic segmentation. The synthetic raster images may encompass features like trichromatic color values, XYZ-surface normals, and scalar field depth information [3]. One significant advantage of projection-based PCSS is the high availability of annotated training data since developers can revert to the knowledge base of standard image semantic segmentation resources and adopt pre-trained models through transfer learning [122,123]. Nevertheless, only a few studies have used multi-view-based deep learning approaches for PCSS in construction because of fundamental limitations. First, the performance of multi-view PCSS methods is sensitive to viewpoint selection and occlusions. For large scenes, which are common in the construction environment, it is challenging to select adequate viewpoints and camera angles to cover the whole scene; hence, the projection step inevitably introduces information loss [25]. Besides their limitations and the intense competition among the different approaches (Figure 13), the latest DeepViewAgg model [124] produces competitive state-of-the-art semantic segmentation results by leveraging multi-view aggregation to merge features from images taken at arbitrary positions. This method has the advantage of not being dependent on colorized point clouds, facilitating a more general sensor choice and higher recording speed.



**Figure 14.** Illustration of two projection-based methods. (a) Multi-view representation, snapshot planes; originally shown in [125]. (b) Spherical representation; originally shown in [126].

**Spherical representation** methods project the whole point cloud, captured by a single scan, onto a sphere with the origin in place of the capturing device's position [127]. Several LiDAR sensors already represent the raw input data in a range-image-like fashion [128]. Because of its synergy with 360-LiDAR sensors, spherical projection representation, as shown in Figure 14b, has proven to be extremely valuable in autonomous driving applications [127–129], where the real-time and robust perception of the environment is indispensable, while point cloud density and a low level of detail are secondary. Because of its runtime advantage, Xu et al. [130] called projection methods the “de facto” method for large-scale point cloud segmentation. Compared to single-view projection, spherical projection retains more information. However, this intermediate representation inevitably introduces several problems, such as discretization errors and blurred CNN outputs, due to subsampling to the resolution of the synthetic image [128]. If the center of the projection is not equal to the initial sensor's position, the method suffers from the same problems as multi-view representation, i.e., occlusion and translucency. Due to the dependence on the sample locations, projection methods are only suitable to a limited extent for practical

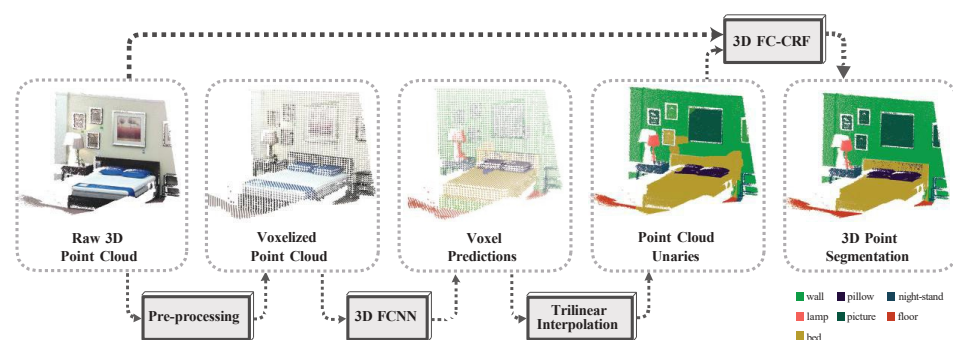
application in the construction industry. No published benchmark results of projection methods are known for the labyrinthine S3DIS indoor dataset to compare in Figure 13.

## 5.2. Discretization-Based Methods

Discretization methods aim to transform raw, unstructured point clouds into a grid structure for enabling 3D indexing and storing agent neighborhood information. Discretization-based approaches can be classified into two types: dense volumetric discretization and sparse permutohedral discretization.

**Dense discretization** representation methods first compute the boundary box surrounding the total point cloud and then subdivide the box space into a user-defined grid. Grid instances are called voxels and can be imagined as pixels in 3D. The position of a voxel is inferred from its position to other voxels. Neighboring voxels can be retrieved by alternating the index. The point cloud is subdivided by the grid, and every voxel is checked for the occupancy of points. In the simplest case, a voxel is labeled as *true* if occupied by points and *false* if not. In a more sophisticated version, the algorithm calculates a density value by counting how many points lie within each cell [131]. This regular data can be fed to a 3D fully convolutional neural network for voxel-wise semantic segmentation. Finally, all points within a voxel are assigned the same predicted semantic label as the voxel. Among deep learning on 3D data, volumetric CNNs are the pioneers in applying 3D CNNs on voxelized shapes [17].

Adopted from Tchapmi et al. [5], Figure 15 represents the basic steps of voxel discretization, the prediction of class labels per voxel, and the mapping back to the unordered point cloud. The stepwise snapshots illustrate one of the main drawbacks of voxelized data representation: voxelization involves point cloud downsampling, which introduces data loss and causes classification artifacts from back-projection. An inherent limitation of this technique is its sensitivity to the granularity of the voxels. Increasing the voxel resolution can alleviate some of these issues. However, the voxel size is a complex trade-off between the level of detail (LOD) of the output, memory usage, and computational complexity. Thus, identifying an optimal voxel resolution is a challenging task [25]. Furthermore, it should be noted that dense discretization strategies with fixed-size voxel structures entail the storage of not only occupied spaces but also free or inner spaces [121]. As a result, this approach may lead to memory overhead, particularly in the case of large scenes common for construction.



**Figure 15.** Process steps for 3D semantic segmentation on voxelized point clouds. The steps show the voxel discretization, including point cloud downsampling, the per-voxel class prediction, and the label back-projection onto the unstructured point cloud. Prediction artifacts can be optimized to produce final results. Originally shown in [5].

**Sparse discretization** representation methods decrease inefficiencies by omitting unoccupied space. Earlier approaches like OctNet [132] hierarchically partition the sparse point cloud, using a set of unbalanced octrees. Tree structures allow memory allocation and computation to focus on relevant dense voxels without sacrificing resolution. However, empty space still imposes computational and memory burdens in OctNet. Graham et al. [133] proposed a novel submanifold sparse convolutional network (SSCN) based on indexing,

which does not perform computations in empty regions, overcoming the drawback of OctNet. Based on these findings, MinkowskiNet [134] enables the direct processing of 3D sequences (e.g., LiDAR streams) with 4D spatiotemporal ConvNets with generalized sparse convolutions. Contrary to cubic discretization, approaches like LatticeNet [135] tessellate the scene space with  $d$ -dimensional permutohedral lattices into  $d$ -dimensional simplices (simplices are triangles for  $d = 2$  and tetrahedrons for  $d = 3$ ). The vertices of the permutohedral lattice store only the simplices, which contain non-empty regions. This sparse allocation allows for efficient implementation of all typical operations in CNNs. Spatial discretization is favorable in the construction industry, where true volumetric scene reconstruction is required and the level of reconstruction detail is secondary. This is the case with navigation and localization applications through the creation of building occupancy maps [136,137].

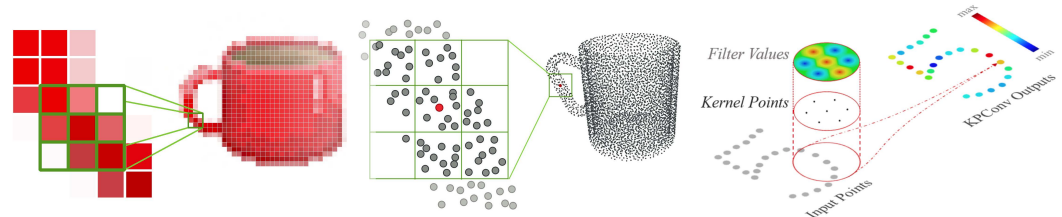
### 5.3. Point-Based Methods

Point-based methods do not rely on an intermediate representation. However, the lack of canonical order and permutation invariance makes raw point data infeasible for convolutional network architectures that require regular input data. This was first solved in 2017 with *PointNet* [17]. Unlike the previously mentioned projection and discretization methods, points are processed through a fully connected multi-layer perceptron (MLP) network to learn per-point features. The key to this approach was a single max-pooling layer that was trained to select a subset of informative points. The output is a global signature of the input set, which can be used for shape classification tasks. However, pointwise semantic segmentation requires local and global knowledge. Qi et al. [17] achieved global awareness with an additional segmentation network, which feeds the global point cloud feature vector back to the local per-point features by simple matrix–vector concatenation. Per-point class scores can be predicted from the combination of local and global point features. Based on the success of PointNet, various segmentation methods have been proposed: point-based convolution networks, graph-based networks, RNN-based networks, and point-based MLP networks.

**Point-based convolutional neural networks** (CNNs) apply convolving kernels at each point in the point cloud to learn pointwise features. However, directly convolving kernels against features associated with the points results in losing shape information and variance to point ordering [106]. Many approaches have been proposed to address these issues. Hua et al. [138] invented a convolution operator that bins nearest neighbors into kernel cells before convolving with kernel weights, which can be applied at each point of a point cloud (Figure 16 center). Wang et al. [139] proposed continuous convolution, a new learnable operator that operates over non-grid structured data and uses kernel functions defined for arbitrary points in the continuous support domain. Similarly, Boulch [140] replaced discrete kernels with continuous ones to generalize convolution for point clouds. This formulation allows arbitrary point cloud sizes and can easily be used for designing neural networks similar to 2D CNNs. Li et al. [106] proposed a method to learn a  $\chi$ -transformation that weights and permutes input points and features before convolving them. The jointly learned  $\chi$ -operator is explicitly dependent on the input order, as  $\chi$  is trained to permute the feature vector to ensure permutation invariance. According to Li et al. [106], this only works to a limited extent but is still significantly better than the direct application of typical convolutions on point clouds.

Real-world data are typically associated with inhomogeneous point density. Methods based on a fixed number of samples deteriorate when point density fluctuates [18,141]. Hermosilla et al. [142] used Monte Carlo convolution for non-uniformly sampled point clouds by phrasing the convolution integral as a Monte Carlo approximation and, thus, providing a new form of robust sampling invariance. Similarly, Wu et al. [143] extended the Monte Carlo approximation method with an MLP in each filter to approximate the weight functions and a density scale to re-weight the learned weight function. They call this permutation-invariant convolution operations PointConv. However, the “naive” implemen-

tation of PointConv is memory-consuming and inefficient. A reformulation by reducing PointConv to two standard operations (matrix multiplication and 2D convolution) takes advantage of GPU parallel computing and allows easy implementation with mainstream deep learning frameworks. Thomas et al. [100] presented kernel point convolution (KPConv) for flexible and deformable convolutions on sparse point clouds, as shown on the right of Figure 16. This method is inspired by image-based convolution, but instead of kernel pixels, it uses a set of kernel points to define the volume, where a linear correlation function applies each kernel weight. For greater flexibility, the number of kernel points is not fixed, and the positions of the kernel points are formulated as an optimization problem of the best coverage in a sphere space and trained to fit the point cloud geometry.



**Figure 16.** Left: pixel-wise convolution with kernel size  $3 \times 3$ . Center: Pointwise convolutions: For each point, nearest neighbors are searched and binned into kernel cells before convolving with kernel weights. Graphic originally shown in [138]. Right: Kernel point convolution 2D example. Input points with a constant scalar feature (in grey) are convolved through a KPConv that is defined by a set of kernel points (in black) with filter weights on each point. The graphic was originally shown in [100].

Liu et al. [144] (FG-Net) published a general deep learning framework for large-scale point clouds to tackle issues (noise, outliers, and dynamically changing environments) associated with methods, designed for small point clouds. They introduce a deformable convolution for modeling local object structures with kernels that dynamically adapt to geometries. Pointwise attention aggregation is used to capture the distributed contextual information in spatial locations and semantic features across a long spatial range.

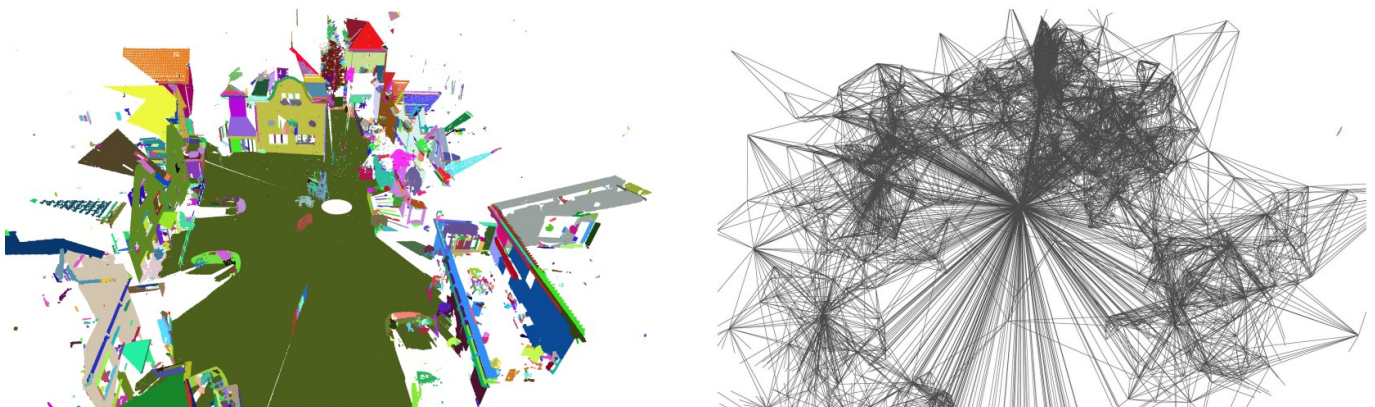
Tang et al. [145] showed how to improve the segmentation results of a well-known ConvNet baseline model [140] by applying contrastive boundary learning (CBL) to enhance feature discrimination between points across boundaries. Experiments demonstrate that CBL can help to improve predictions, especially in cluttered regions, and reduce classification artifacts. The CBL framework is not limited to ConvNet architecture but can be coupled to any other multi-stage backbone, as results with the RandLA-Net backbone [146] have shown.

**Point-based graph neural networks (GNNs)** allow a natural representation of data (point cloud and mesh) within non-Euclidean space. The nature of these data does not imply familiar properties, such as an orthonormal coordinate system, vector space structure, or shift invariance. Consequently, basic operations such as convolution, which are taken for granted in the Euclidean case, are not even well defined in non-Euclidean domains [147]. In recent decades, researchers have been working on how to conduct convolutional operations on graphs. Graph convolutional network (GCN) models are neural networks that can leverage the graph structure and aggregate node information from the neighborhoods in a convolutional fashion [148].

Landrieu and Simonovsky [149] suggested a superpoint graph (SPG) (Figure 17) constructed from simple yet meaningful shapes, which partition from the global point cloud by local geometric features (linearity, planarity and scattering, and verticality). Superedges interconnect superpoints in a global graph to capture the spatial context information, which is then exploited by a GCN. Wang et al. [150] proposed a dynamic graph CNN (DGCNN), an EdgeConv-based model to group points in Euclidean and semantic space over potentially long distances. The EdgeConv module applies to the dynamically updated graphs in each network layer. Zhiheng and Ning [151] published the PyramNet architecture



as a combination of a graph embedding module (GEM) and a pyramid attention network (PAN). To improve local feature expression ability, GEM utilizes a covariance matrix to replace the Euclidean distance. The PAN combines features of different resolutions and different semantic strengths from four convolution kernels to improve segmentation. Wang et al. [152] used graph attention convolution (GAC) with learnable kernel shapes to adapt to the structure of the objects. This is achieved by dynamically assigning attention weights to different neighboring points and feature channels based on their spatial positions and feature differences. Ma et al. [153] tested a plug-and-play point global context reasoning (PointGCR) module, which can help to capture context information along the channel dimension by using an undirected graph representation and self-attention mechanism. The authors show a significant boost in performance with the PointGCR module appended to several famous encoder–decoder networks for point cloud segmentation on outdoor and indoor scenes. Xie et al. [154] published the multi-resolution graph neural network (MuGNet) architecture, which translates large-scale point clouds into directed connectivity graphs and efficiently segments the point clouds with a bidirectional graph convolution network. Using a multi-resolution feature fusion network reduces memory consumption but conserves rich contextual relationships. The success of this algorithm was its ability to train deep networks reliably. However, training very deep GCNs comes with certain shortcomings. First, stacking multiple GCN layers leads to the vanishing gradient problem, the same as in CNNs. Second, the over-smoothing problem can occur when repeatedly applying many GCN layers [155]. Thus, most state-of-the-art GCNs are limited to shallow network architectures, usually no deeper than four layers [156,157].



**Figure 17.** Visualization of individual steps in the superpoint graph pipeline. **Left:** geometric partitions from superpoints. **Right:** visualization of the interconnecting superpoint graph (SPG); originally shown in Landrieu and Simonovsky [149].

**Point-based recurrent neural network (RNN)** methods aim to improve scene understanding by treating the point input as a sequence of features. RNNs have proven to be highly effective for processing sequence data with variable lengths, like sensor data streams, human speech, and written text [158]. Few attempts exist to leverage the power of RNNs and the ability to memorize the prior input for point cloud semantic segmentation. Engelmann et al. [159] improved on the first PointNet [17], which is bound to subdivide larger point clouds into a grid of blocks and process each block individually. In the proposed network, one recurrent consolidation unit (RCU) inputs a sequence of block features originating from four spatially nearby blocks and returns a sequence of corresponding updated block features. The RCU is configured as a gated recurrent unit (GRU) where updated block features are returned only after the GRU has seen the whole input sequence. The GRU retains relevant information about the scene in their internal memory and updates it according to new observations. Huang et al. [160] (RSNet) suggested a lightweight local dependency module that uses three slice pooling layers along the X, Y, and Z axis to convert unordered point feature sets into an ordered sequence of feature vectors. By modeling

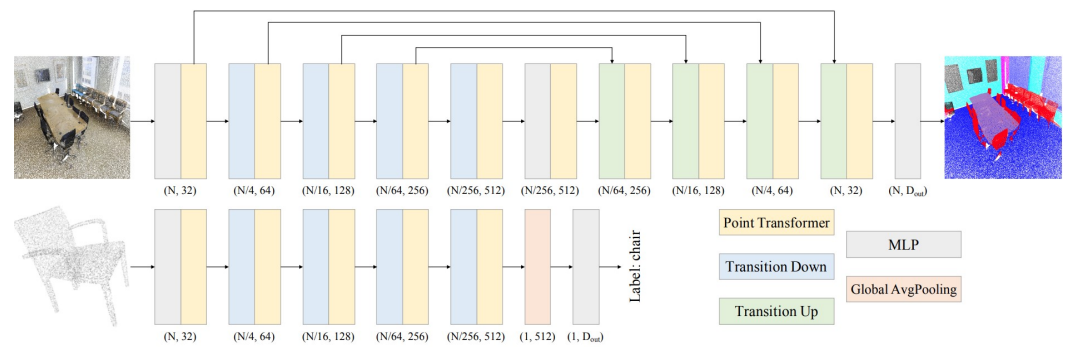
one slice as one timestamp, the information interacts with other slices as the information flows through the RNN internal state (memory). Comparably, Ye et al. [161] (3P-RNN) utilized a combination of pointwise pyramid pooling to capture local geometries and a two-directional sliding window hierarchical RNN to explore long-range spatial dependencies in the X and Y direction. To this extent, the above RNN-based methods rely on static pooling to aggregate features from the local neighborhood, without adapting to density changes [162]. Similar to the aforementioned dense voxel discretization methods, this leads to computational inefficiencies and poor segmentation results. At this point, the latest state-of-the-art benchmark results, like S3DIS's large-scale indoor scene benchmark [59] (Figure 13), show that RNN-based and GCN-based methods lately cannot keep up with modern point-based convolution, MLP-pooling, and transformer-based methods.

**Point-based multi-layer perceptron networks** (MLPs) usually process each point individually through shared MLP to perform local feature aggregation. MLPs are highly efficient but often face difficulties in grasping the global context and spatial features of point clouds [17,163]. Among the point-based MLP networks, state-of-the-art methods can be subdivided into pooling-based and attention-based transformer approaches, to capture a wider context for each point and learn richer local structures.

*Pooling-based methods* learn to consolidate features in the local neighborhood of a point group. The hierarchical neural network PointNet++ [18] can recognize fine-grained patterns and generalizes complex scenes by recursively applying PointNet on a nested partitioning of the input point set. The information from different neighborhood sizes is aggregated for each point of interest with neighboring feature pooling. Thus, MLP-based methods can improve general feature learning for non-uniform point density in different areas [18]. In contrast to PointNet++, which defines the neighborhood from the metric world space, Engelmann et al. [164] employed two grouping mechanisms to define neighborhoods in the world space by clustering points using K-means, but also in the feature space, by computing the  $k$  nearest neighbors (KNNs). Assuming that points from the same semantic class are likely to be nearby in the feature space, they defined a pairwise similarity loss function and found with [165] that semantic similarity can be measured as a distance in the feature space. Qiu et al. [166] (BAAF-Net) resolved three major drawbacks of existing works that affect segmentation quality: ambiguity in close points, redundant features, and inadequate global representations. While most architectures are designed to solve a broad range of scene understanding tasks (including object detection and shape classification), the BAAF-Net architecture entirely focuses on large-scale point cloud semantic segmentation. This is conducted by the concept of augmenting the local context bilaterally and fusing multi-resolution features for each point adaptively. Lin et al. [85] proposed a unified framework called PointMeta. Its building blocks can be summarized in terms of four meta functions: a neighbor update function, a neighbor aggregation function, a point update function, and a position embedding function (implicit or explicit). By modifying the existing approaches, Lin et al. [85] derived a basic building block (PointMetaBase). PointMetaBase-XXL surpasses previous methods in several PCSS benchmarks (Figure 13) in the configuration with multiple stacked PointMetaBase blocks, max-pooling as neighborhood aggregation function, and Explicit Position Embedding (EPE) adopted from Point Transformer [108]).

*Transformer-based methods*, first introduced to the field of natural language processing (NLP) [167], are designed to process sequential input data. The Transformer is a decoder–encoder structure with three main modules for input (word) embedding, positional (order) encoding, and self-attention. The self-attention module is the core component that relates different positions of a single sequence to compute a representation of the sequence. Following the success of Transformer network architectures in the language and image domains, where competitive performance was achieved compared to their CNN counterparts [168–170], Transformers have lately been applied to process unordered 3D point clouds. Since point clouds essentially are sets of points with positional attributes, the self-attention mechanism seems particularly suitable for these data type [108,167].

The simplest approach to modify the Transformer [167] for point cloud processing implies treating the entire cloud as a sentence and each point as a word. Guo et al. [171] propose a naïve point cloud Transformer (PCT) by implementing a coordinate-based point embedding, which ignores interactions between points and instantiates the attention layer with the self-attention proposed in the original Transformer [167]. Without further modification and purely based on coordinate features, the naïve PCT was capable to outperform state-of-the-art methods. Yang et al. [172] presented a point attention Transformer (PAT) with two core operations: group shuffle attention (GSA) for mining relations between elements in the feature set and gumbel subset sampling (GSS), to select a representative subset of input points. To improve frame rates in real-time applications, Hu et al. [146] published RandLA-Net, a lightweight architecture that directly infers per-point semantics for large-scale point clouds. It uses random point sampling instead of more complex and computationally heavy point-selection approaches. RandLA-Net introduces a novel local feature aggregation module that increases the receptive field for each 3D point and leverages attentive pooling to preserve local features and overcome the loss of key features expected from random sampling. Besides the significant downsampling, the method can retain features necessary for accurate segmentation. Engel et al. [173] designed a Point Transformer architecture that utilizes a SortNet module and global feature generation to extract local and global features and relates both representations by introducing the local–global attention mechanism. The output of Point Transformer is a sorted and permutation-invariant feature list that can directly be incorporated into common computer vision applications. With the same name, Zhao et al. [108] proposed another Point Transformer architecture, entirely based on vector self-attention and pointwise operations, as a general backbone for several 3D scene understanding tasks. These encoder and decoder structures consist of modular point transformer layers, stacked in multiple stages with transition layers for down- and upsampling. For dense prediction tasks such as semantic segmentation, Point Transformer adopts a U-Net [174] design, as shown in Figure 18. The number of stages and the sampling rates can be varied depending on the application, e.g., to construct a lightweight backbone for fast processing. To overcome its early limitations, Wu et al. [175] proposed group vector attention, a revised Point Transformer V2 architecture with improved position encoding and with an efficient partition-based pooling scheme. Lai et al. [176] proposed a Stratified Transformer to compensate for the drawbacks of the Point Transformer (V1) [108] and improve on capturing long-range context in point clouds by using standard multi-head self-attention [167]. Partitioning a point cloud into non-overlapping cubic windows limits the effective receptive field (EFR) [177] to the local region, which causes false predictions. Enlarging the window size helps to increase the respected area, but comes with a higher memory cost. To extend the attention beyond the limited local region, Lai et al. [176] used a stratified strategy to ensure that the subgroup (strata) is adequately represented. Points within a small window next to a query point are sampled densely. Points within a second large window are sampled sparsely as a trade-off between EFR and memory consumption. Wang et al. [107] built upon Stratified Transformers and achieved state-of-the-art point cloud semantic segmentation results on the S3DIS large-scale indoor scene benchmark, as shown in Table A1, with leading accuracy scores on structural component classification (ceiling, floor, wall, and column). Improvements are gained from the proposed window normalization with prior knowledge to account for variant local neighborhood point cloud density in the downsampling step.



**Figure 18.** Point Transformer network architecture for semantic segmentation (**top**) and classification (**bottom**); originally shown in [108]. The encode–decoder structure is a U-Net [174] shape.

## 6. Discussion

With this survey, we have shown that point cloud semantic segmentation has made great strides through recent developments in machine learning. It has become a significant research area in computer vision, with broad applications in fields such as robotics, autonomous driving, and geodesy. The construction industry can greatly benefit from that. However, several challenges must be addressed before these methods can be fully integrated into industrial applications. In Section 3, we showed that there are currently no significant datasets available for construction-related point cloud learning. The state-of-the-art publications we surveyed used one of two workarounds to circumvent this shortage. Most publications focus on one of the few well-established datasets [59,62,131] for indoor scenes, which are, however, polluted by furniture and often sourced from outdated RGB-D sensors. Fewer publications established their private task-specific datasets [19,24,86] but avoid sharing this with the public. Not sharing the deployed data with the public makes the reproduction of the results impossible and complicates the further use of the results. This study highlights a significant challenge in the training of neural networks through supervised learning, wherein the availability of substantial human-labeled data remains crucial. Specifically, the creation of extensive, well-documented, and open-source datasets comprising point clouds combined with technical meta-data (such as component labels, object properties, and damage classification) emerges as a major hurdle. Despite conducting extensive research, viable alternatives to large annotated datasets for effectively training machine learning models have not yet been identified.

Transfer learning [122,123] is a powerful tool that can lower the amount of required training data; yet, context-specific data to fine tune the model does not become obsolete. The opposite is the case. Fine tuning a pre-trained model raises the demand for high-quality and versatile data [178,179]. The need for data applies also to the required number of different datasets, designed for different sectors and applications in the construction industry. Sectors include building structures, civil engineering (over- and underground), infrastructure, and pre-cast fabrication. Applications include object detection, shape classification, semantic and instance segmentation, 3D reconstruction, and SLAM.

The preparation of such datasets is a non-trivial task. First of all, it requires expensive hardware to capture the data. Secondly, human resources are needed to annotate data, an activity that is even more challenging in the 3D domain than in the 2D domain. Furthermore, domain experts are needed to create and review the dataset. Datasets must be extrinsically balanced (regarding architecture and style, location, and environmental changes) and intrinsically (in terms of their beholding classes). The huge efforts needed to create qualitative datasets have led companies to keep their elaborately compiled data under lock when data have become a valuable commodity in all industries. However, open-sourcing these data is necessary for interoperability (organizations or systems to work together), saving economic costs (avoiding collecting new data) and improving data quality (crowd-sourced debugging) and verification (reproducibility). Comprehensive and



insightful databases, like the one we proposed in Table A1, help make such data more visible and accessible and, thus, boost development.

Balancing datasets retroactively is only possible to a limited extent, as we explain in Section 3. One lesson we learned from reviewing multiple datasets within this survey was to keep track of the object distribution already in the process of creating a new dataset. The statistics about class distribution, as proposed in Section 2, in terms of points per class as well as instances per class, help to keep an overview and to counteract iteratively in the case of disequilibrium. From the findings in the first half of this paper, we advocate for the community to put more effort into the creation of industry-standard datasets and for the recipients to honor the hard work of developing datasets.

Synthetic data are considered a complementary solution against dataset scarcity because synthetic data annotation is essentially free [180]. Table 1 features synthetic datasets and singular attempts that utilize synthetic training data to improve real-world performance. An outstanding example is the VASAD dataset [22] for building reconstruction. Researchers argue that the transfer of knowledge can improve the ability to perform complex tasks when initially performed in simulation [181,182]. Furthermore, the class imbalance problem becomes obsolete if class–object appearance in scenes is user-defined. However, experience with synthetic data to improve point cloud semantic segmentation is still limited, and studying their effects is currently under investigation. For example, the effect known as the “Sim2Real (sim-to-real) gap” [182,183] describes the discrepancy between simulated and real environment data, a phenomenon that can result in bad performance if training the network with synthetic data but applying the model to the real world [184].

Unlike training data, which are content and context-locked, general machine learning models can be transferred independently on their initial application as this survey shows. Algorithms initially developed to segment, e.g., cars and pedestrians, can be successfully deployed to segment buildings, if trained with the appropriate data. One objective of this survey has been to identify the future trends and most promising methods for PCSS to channel our future research and share this with everyone with the same intentions. Several authors described their methods as the most suitable for PCSS and scene understanding, but this review leads to the assumption that, still, no method proves to be dominant. Convolutional neural networks (CNNs) have traditionally been preferred for processing image data, while recurrent neural networks (RNNs) have been for sequential data. However, we find no consensus on the most appropriate method to analyze the 3D domain. Projection-based methods have legitimation in autonomous driving and mapping applications because of fast processing and their synergy with directed stereo cameras and spherical LiDAR if the demand for resolution is low. Discretization-based learning methods for PCSS appear to be not competitive due to memory overhead with fully connected CNNs and the lack of canonical order for permutation invariance. However, voxel discretization remains important for all kinds of point cloud processing [185] and map building in SLAM. The reviewed RNN methods for point cloud understanding struggle to adapt to changing point cloud density and rely on extensive partitioning procedures. Only a few approaches of this kind could be found to deal with large-scale PCSS. Like in natural language processing, transformers recently look more promising for the sequence datatypes [167]. Graph neural networks (GNNs) and graph convolution offer a natural representation of point clouds and claim to solve the issue of capturing long-range spatial context information. However, stacking multiple GCN layers leads to a vanishing gradient and over-smoothing. Autonomous vehicles and a network of sensors on our roads potentially soon produce a huge 3D roadmap dataset [186]. There is no expertise in this field for now, but GNNs might be a valuable approach for scene understanding in infrastructure civil engineering with low requirements for detail but the need to process huge spatial datasets. We find that point-based networks, among all the mainstream methods, look the most promising for PCSS and support this statement by the leaderboard in Table A1 and additional public benchmarks [30,62]. Pooling-based MLPs, attention-based transform-



ers, and point-based CNNs perform on par, with small deviations depending on which benchmark is investigated. Yet, multiple sources lately comment on transformers being the most versatile general-purpose architecture for different types of data and computer vision tasks [168,187,188]. Even though, many of the new transformer models still incorporate the best parts of convolutions. That means future models are more likely to use both than to abandon CNNs entirely.

There are, of course, some drawbacks associated with the topic of deep learning in general and PCSS for the construction industry, which we want to discuss. We find in Section 3.2 that PCSS results are sensitive to geometric similar shapes and models easily get confused in the presence of class imbalance. This problem can be illustrated with the practical example of bridge's part segmentation. Even though most bridges have similar components with the same function, these components look very different across several bridge types. If models can not generalize geometries, the industry faces the risk that the needed number of specialized datasets adds up fast, eventually becoming impossible to cover. Further, the costs associated with deep learning are the high demand for computing power, particularly in the training phase, and also for inference. This is true for all deep learning, but transformers are particularly affected [189,190]. Training the models requires expensive GPU server clusters, which are not common in conventional civil engineering offices. Cloud services can be a workaround for the training phase, but real-time application on-site will require equipping robots with powerful hardware. Finally, point cloud semantic segmentation is only one first step of machine learning applications in civil engineering. Instance segmentation is becoming increasingly important, as the construction industry continues to integrate digital technologies into the building process and construction surveillance [191–193]. Instance segmentation can be considered a refined version of semantic segmentation. Where semantic segmentation assigns all segments (points) of the same recognized class into one global group, instance segmentation can differentiate between different objects of the same group. With the ability to automatically identify and label objects in a 3D point cloud, instance segmentation can provide valuable information for various construction-related tasks, including fully autonomous robot operation, as-built reconstruction, automated building information modeling (BIM), progress tracking, and quality control.

## 7. Conclusions

Point cloud learning has gained strong attention due to its numerous applications in various fields of computer vision, but the visual understanding of construction sites by deep learning, such as semantic segmentation, is hardly mentioned in the literature. Fortunately, general learning-based approaches can also be transformed into solving construction-related tasks, with the right training data at hand. This paper presented a contemporary survey of the state-of-the-art algorithms for learning-based PCSS methods tailored toward the construction industry and its unique demands. We conducted a comprehensive literature review of the available datasets and well-established model architectures. An unprecedented database for scene understanding training datasets was evaluated, and the evaluation of the indoor scene's largest benchmark for PCSS methods was presented. From the revision of the latest state-of-the-art publications, we found a strong future trend toward transformer-based model architecture in the presence of dense point clouds with a high level of detail. Applications with requisitions of very fast inference times still profit from point projections to leverage the well-adopted 2D convolutional image segmentation. However, this comes at the expense of a lower level of detail. Very large sparse point clouds seem to profit from graph-based methods. When considering that, in the recent past, the majority of research in the field of deep learning has been focused on Transformers as a versatile tool for various applications, we expect that this trend will also impact the construction industry. The initial indications of this development were evident in our benchmark comparison. Falling prices for sensor hardware and increasing success in all areas suggest that the number of applications will only increase. One challenge that should

not go unmentioned is the question of how to meet the increasing demand for computing power. This question remains unanswered today, but should be solvable in the future with the development of more efficient tensor processors and cloud computing.

The extensive review of public datasets revealed a significant data scarcity to train, test, and validate supervised learning, which must be treated by domain experts from inside the industry itself. This challenge can be best tackled together with a large community. We hope to contribute to the success with this work and accelerate future developments by summarizing and providing an extensive database of the state-of-the-art. Future research is encouraged to close this gap but also needs societies that provide the necessary funding.

**Author Contributions:** Conceptualization, L.R.; methodology, L.R.; formal analysis, L.R.; investigation, L.R.; resources, L.R.; data curation, L.R.; writing—original draft preparation, L.R.; writing—review and editing, L.R. and T.B.; visualization, L.R.; supervision, T.B.; funding acquisition, L.R.; All authors have read and agreed to the published version of the manuscript.

**Funding:** We acknowledge financial support by the Universität der Bundeswehr München for covering the article processing costs, which enabled us to publish this article in open-access.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are openly available at [https://github.com/RauchLukas/Article-PCSS\\_for\\_Construction-A\\_survey](https://github.com/RauchLukas/Article-PCSS_for_Construction-A_survey) (accessed on 31 July 2023).

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

ADASYN	Adaptive Synthetic Sampling
AFA	Adaptive Feature adjustment
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AP	Average Precision
ARM	Autonomous Mobile Robot
AUC	Area Under the ROC Curve
BIM	Building Information Model
BP	Back-Propagation
CBL	Contrastive Boundary Learning
CNN	Convolutional Neural Network
DL	Deep Learning
FN	False Negative
FP	False Positive
GCN	Graph Convolutional Network
GEM	Graph Embedding Module
GNN	Graph Neural Network
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
GSA	Graph Shuffle Attention
GSS	Gumbel Subset Sampling
IMU	Inertial Measurement Unit
IoU	Intersection over Union
KNN	K-Nearest Neighbors
LiDAR	Light Detection and Ranging
LoD	Level of Detail
mAP	Mean Average Precision

MEP	Mechanical, Electrical, and Plumbing
mIoU	Mean Intersection over Union
MLP	Multi-Layer Perceptron
MLS	Mobile Laser Scanning
mPrec	Mean Precision
mRec	Mean Recall
NLP	Natural Language Processing
NN	Neural Networks
OA	Overall Accuracy
PAT	Point Attention Transformer
PCS	Point Cloud Segmentation
PCSS	Point Cloud Semantic Segmentation
PCT	Point Cloud Transformer
PR	Precision–Recall
RCU	Recurrent Consolution Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
S3DIS	Stanford Large-Scale 3D Indoor Spaces Dataset
SLAM	Simultaneous Localization and Mapping
SMOTE	Synthetic Minority Over-sampling Technique
SPG	Superpoint Graph
SVM	Support Vector Machine
TLS	Terrestrial Laser Scanning
TN	True Negative
TP	True Positive

## Appendix A

### Appendix A.1. Published Benchmark Results on the S3DIS Dataset

**Table A1.** Reported results for semantic segmentation task on the large-scale indoor S3DIS benchmark (including all 6 areas, 6-fold cross validation). Ranked in descending order based on mIoU performance. Declaration: convolution-based (C), graph-based (G), hybrid (H), pooling-based (P), RNN-based (R), Transformer-based (T), voxel-based (V).

Rank	Year	Model Name	Ref.	Method	mIoU	mAcc	oAcc
1	2022	WindowNorm + StratifiedTransformer	[107]	T	77.60	85.8	91.7
2	2022	PointMetaBase-XXL	[85]	P	77.00	-	91.3
3	2022	PointNeXt-XL	[194]	P	74.90	83.0	90.3
4	2022	DeepViewAgg	[124]	H	74.70	83.8	90.1
5	2022	RepSurf-U	[195]	P	74.30	82.6	90.8
6	2022	WindowNorm +PointTransformer	[107]	T	74.10	82.5	90.2
7	2022	PointNeXt-L	[194]	P	73.90	82.2	89.9
8	2020	PointTransformer	[108]	T	73.50	81.9	90.2
9	2022	CBL	[145]	C	73.10	79.4	89.6
10	2021	BAAF-Net	[166]	P	72.20	83.1	88.9
11	2021	SCF-Net	[196]	P	71.60	82.7	88.4
13	2020	FG-Net	[144]	C	70.80	82.9	88.2
12	2021	RPNet-D27	[197]	P	70.80	-	-
14	2019	KPConv	[100]	C	70.60	79.1	-
15	2021	FastPointTrans. (small)	[198]	T	70.30	-	-
16	2018	PointSIFT	[199]	P	70.23	-	88.72
17	2019	RandLA-Net	[146]	T	70.00	81.5	87.1
18	2020	MuGNet	[154]	G	69.80	-	88.5
19	2020	PointASNL	[200]	P	68.70	79.0	88.8
20	2020	FPConv	[201]	C	68.70	-	-
21	2019	SSP+SPG	[202]	G	68.40	78.3	87.9
22	2020	FKACConv	[203]	C	68.40	-	-
23	2019	ConvPoint	[140]	C	68.20	-	88.8
24	2019	HPEIN	[204]	G	67.82	76.26	88.2
25	2020	JSENet	[205]	C	67.70	-	-

Table A1. Cont.

Rank	Year	Model Name	Ref.	Method	mIoU	mAcc	oAcc
26	2020	CT2	[206]	T	67.40	-	-
27	2019	ShellNet	[207]	P	66.80	-	-
29	2019	InterpCNN	[208]	C	66.70	-	88.7
28	2019	PointWeb	[209]	P	66.70	76.2	87.3
30	2019	PAG	[210]	G	65.90	-	88.1
31	2019	MinkowskiNet	[134]	V	65.40	-	-
32	2018	PointCNN	[106]	C	65.40	75.6	88.1
33	2019	DPAM	[211]	G	64.50	-	87.6
34	2019	PAT	[172]	T	64.28	-	-
35	2021	DSPoint	[212]	H	63.30	70.9	-
36	2019	A-CNN	[213]	C	62.90	-	87.3
37	2019	LSANet	[214]	P	62.20	-	86.8
38	2017	SPG	[149]	G	62.10	73.0	85.5
39	2019	JSNet	[215]	P	61.70	71.7	88.7
40	2019	DeepGCN	[157]	G	60.00	-	85.9
41	2019	ASIS	[216]	P	59.30	70.1	86.2
42	2018	PCNN	[139]	C	58.27	67.01	-
43	2018	Engelmann	[164]	P	58.27	67.77	83.95
44	2018	RSNet	[160]	R	56.50	66.5	-
45	2018	3P-RNN	[161]	R	56.30	73.6	86.9
46	2018	DGCNN	[150]	G	56.10	-	84.1
48	2017	3DContextNet	[217]	P	55.60	74.5	84.9
47	2019	PyramNet	[151]	G	55.60	-	86.6
49	2020	Point-PlaneNet	[218]	P	54.80	-	83.9
50	2017	PointNet++	[18]	P	54.49	67.05	81.03
51	2018	A-SCN	[219]	P	52.72	-	81.59
52	2021	SMS	[220]	G	51.74	-	-
53	2018	G+RCU	[159]	R	49.70	66.4	81.1
54	2016	PointNet	[17]	P	47.71	66.2	78.62

## References

- Arbeláez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [\[CrossRef\]](#) [\[PubMed\]](#)
- Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1915–1929. [\[CrossRef\]](#)
- Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015. [\[CrossRef\]](#)
- Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. 2012. Available online: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/> (accessed on 31 July 2023).
- Tchapmi, L.P.; Choy, C.B.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic Segmentation of 3D Point Clouds. In Proceedings of the International Conference of 3D Vision, Qingdao, China, 10–12 October 2017. [\[CrossRef\]](#)
- Xu, S.; Wang, J.; Shou, W.; Ngo, T.; Sadick, A.M.; Wang, X. Computer Vision Techniques in Construction: A Critical Review. *Arch. Comput. Methods Eng.* **2021**, *28*, 3383–3397. [\[CrossRef\]](#)
- Kim, M.-K.; Cheng, J.-C.; Sohn, H.; Chang, C.-C. A framework for dimensional and surface quality assessment of precast concrete elements using BIM and 3D laser scanning. *Autom. Constr.* **2015**, *49*, 225–238. [\[CrossRef\]](#)
- Jiang, Y.; Huang, Y.; Liu, J.; Li, D.; Li, S.; Nie, W.; Chung, I.-H. Automatic Volume Calculation and Mapping of Construction and Demolition Debris Using Drones Deep Learning and GIS. *Drones* **2022**, *6*, 279. [\[CrossRef\]](#)
- Han, S.; Jiang, Y.; Bai, Y. Fast-PGMED: Fast and Dense Elevation Determination for Earthwork Using Drone and Deep Learning. *J. Constr. Div. Manag.* **2022**, *148*, 04022008. [\[CrossRef\]](#)
- Xiong, X.; Adan, A.; Akinci, B.; Huber, D. Automatic creation of semantically rich 3D building models from laser scanner data. *Autom. Constr.* **2013**, *31*, 325–337. [\[CrossRef\]](#)
- Adán, A.; Ramón, A.; Vivancos, J.L.; Vilar, A.; Aparicio-Fernández, C. Automatic generation of as-is BEM models of buildings. *J. Build. Eng.* **2023**, *73*, 106865. [\[CrossRef\]](#)
- Turkan, Y.; Bosche, F.; Haas, C.T.; Haas, R. Automated progress tracking using 4D schedule and 3D sensing technologies. *Autom. Constr.* **2012**, *22*, 414–421. [\[CrossRef\]](#)
- Son, H.; Kim, C. 3D structural component recognition and modeling method using color and 3D data for construction progress monitoring. *Autom. Constr.* **2010**, *19*, 844–854. [\[CrossRef\]](#)
- Wang, Q.; Cheng, J.C.; Sohn, H. Automated Estimation of Reinforced Precast Concrete Rebar Positions Using Colored Laser Scan Data. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 787–802. [\[CrossRef\]](#)
- Chen, J.; Fang, Y.; Cho, Y.K.; Kim, C. Principal Axes Descriptor for Automated Construction-Equipment Classification from Point Clouds. *J. Comput. Civ. Eng.* **2017**, *31*, 0401605. [\[CrossRef\]](#)
- Ray, S.J.; Teizer, J. Computing 3D blind spots of construction equipment: Implementation and evaluation of an automated measurement and visualization method utilizing range point cloud data. *Autom. Constr.* **2013**, *36*, 95–107. [\[CrossRef\]](#)

17. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [\[CrossRef\]](#)
18. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
19. Yin, C.; Wang, B.; Gan, V.J.; Wang, M.; Cheng, J.C. Automated semantic segmentation of industrial point clouds using ResPointNet++. *Autom. Constr.* **2021**, *130*, 103874. [\[CrossRef\]](#)
20. Perez-Perez, Y.; Golparvar-Fard, M.; El-Rayes, K. Segmentation of point clouds via joint semantic and geometric features for 3D modeling of the built environment. *Autom. Constr.* **2021**, *125*, 103584. [\[CrossRef\]](#)
21. Su, Y.; Liu, W.; Yuan, Z.; Cheng, M.; Zhang, Z.; Shen, X.; Wang, C. DLA-Net: Learning Dual Local Attention Features for Semantic Segmentation of Large-Scale Building Facade Point Clouds. *Pattern Recognit.* **2022**, *123*, 108372. [\[CrossRef\]](#)
22. Langlois, P.A.; Xiao, Y.; Boulch, A.; Marlet, R. VASAD: A Volume and Semantic dataset for Building Reconstruction from Point Clouds. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August, 2022; pp. 4008–4015. [\[CrossRef\]](#)
23. Xia, T.; Yang, J.; Chen, L. Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning. *Autom. Constr.* **2022**, *133*, 103992. [\[CrossRef\]](#)
24. Yang, X.; Del Rey Castillo, E.; Zou, Y.; Wotherspoon, L.; Tan, Y. Automated semantic segmentation of bridge components from large-scale point clouds using a weighted superpoint graph. *Autom. Constr.* **2022**, *142*, 104519. [\[CrossRef\]](#)
25. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [\[CrossRef\]](#)
26. He, Y.; Yu, H.; Liu, X.; Yang, Z.; Sun, W.; Wang, Y.; Fu, Q.; Zou, Y.; Mian, A. Deep Learning based 3D Segmentation: A Survey. *arXiv* **2021**, arXiv:2103.05423.
27. Jacobsen, E.L.; Teizer, J. Deep Learning in Construction: Review of Applications and Potential Avenues. *J. Comput. Civ. Eng.* **2022**, *36*, 03121001. [\[CrossRef\]](#)
28. Akinosho, T.D.; Oyedele, L.O.; Bilal, M.; Ajayi, A.O.; Delgado, M.D.; Akinade, O.O.; Ahmed, A.A. Deep learning in the construction industry: A review of present status and future innovations. *J. Build. Eng.* **2020**, *32*, 101827. [\[CrossRef\]](#)
29. Khallaf, R.; Khallaf, M. Classification and analysis of deep learning applications in construction: A systematic literature review. *Autom. Constr.* **2021**, *129*, 103760. [\[CrossRef\]](#)
30. Database for Machine Learning Datasets. *Meta AI Research* **2023**. Available online: <https://paperswithcode.com/datasets> (accessed on 27 February 2023).
31. Armeni, I.; Sax, S.; Zamir, A.R.; Savarese, S. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv* **2017**, arXiv:1702.01105.
32. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [\[CrossRef\]](#)
33. LeCun, Y.; Cortes, C.; Burges, C.J. MNIST Handwritten Digit Database. ATT Labs. 2010; Volume 2. Available online: <http://yann.lecun.com/exdb/mnist> (accessed on 15 July 2023).
34. Sanghyeon, A.; Lee, M.; Park, S.; Yang, H.; So, J. An Ensemble of Simple Convolutional Neural Network Models for MNIST Digit Recognition. *arXiv* **2020**, arXiv:2008.10400.
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
36. Wilson, B.; Qi, W.; Agarwal, T.; Lambert, J.; Singh, J.; Khandelwal, S.; Pan, B.; Kumar, R.; Hartnett, A.; Pontes, J.K.; et al. Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting. In Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), Virtual-only Online Conference, 6–14 December 2021.
37. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
38. Mao, J.; Niu, M.; Jiang, C.; Liang, H.; Chen, J.; Liang, X.; Li, Y.; Ye, C.; Zhang, W.; Li, Z.; et al. One Million Scenes for Autonomous Driving: ONCE Dataset. In Proceedings of the Thirty-Fifth Neural Information Processing Systems, Virtual-only Online Conference, 6–14 December 2021.
39. Sun, P.; Kretschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020. [\[CrossRef\]](#)
40. Tan, W.; Qin, N.; Ma, L.; Li, Y.; Du, J.; Cai, G.; Yang, K.; Li, J. Toronto-3D: A Large-scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; Volume 38, pp. 797–806. [\[CrossRef\]](#)
41. Munoz, D.; Bagnell, J.A.; Vandapel, N.; Hebert, M. Contextual classification with functional Max-Margin Markov Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 975–982. [\[CrossRef\]](#)
42. Pandey, G.; McBride, J.R.; Eustice, R.M. Ford Campus vision and lidar data set. *Int. J. Robot. Res.* **2011**, *30*, 1543–1552. [\[CrossRef\]](#)



43. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. [\[CrossRef\]](#)
44. Couprie, C.; Farabet, C.; Najman, L.; LeCun, Y. Indoor Semantic Segmentation using depth information. *arXiv* **2013**, arXiv:1301.3572.
45. Xiao, J.; Owens, A.; Torralba, A. SUN3D: A Database of Big Spaces Reconstructed using SfM and Object Labels. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013. [\[CrossRef\]](#)
46. de Deuge, M.; Quadros, A.; Hung, C.; Douillard, B. Unsupervised Feature Learning for Classification of Outdoor 3D Scans. In Proceedings of the Australasian Conference on Robotics and Automation; University of New South Wales Kensington, Kensington, Australia, 2–4 December 2013.
47. Serna, A.; Marcotegui, B.; Goulette, F.; Deschaud, J.E. Paris-rue-Madame database: A 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods, Angers, France, 6–8 March 2014; Volume 1, pp. 819–824. [\[CrossRef\]](#)
48. Vallet, B.; Brédif, M.; Serna, A.; Marcotegui, B.; Paparoditis, N. TerraMobilita/iQmulus urban point cloud analysis benchmark. *Comput. Graph.* **2015**, *49*, 126–133. [\[CrossRef\]](#)
49. Carlevaris-Bianco, N.; Ushani, A.K.; Eustice, R.M. University of Michigan North Campus long-term vision and lidar dataset. *Int. J. Robot. Res.* **2016**, *35*, 1023–1035. [\[CrossRef\]](#)
50. Handa, A.; Patraucean, V.; Badrinarayanan, V.; Stent, S.; Cipolla, R. SceneNet: Understanding Real World Indoor Scenes With Synthetic Data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015. [\[CrossRef\]](#)
51. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012v1.
52. Song, S.; Lichtenberg, S.P.; Xiao, J. SUN RGB-D: A RGB-D scene understanding benchmark suite. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 567–576. [\[CrossRef\]](#)
53. Xiang, Y.; Kim, W.; Chen, W.; Ji, J.; Choy, C.; Su, H.; Mottaghi, R.; Guibas, L.; Savarese, S. ObjectNet3D: A Large Scale Database for 3D Object Recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 June 2016; Volume 9912, pp. 160–176. [\[CrossRef\]](#)
54. Maddern, W.; Pascoe, G.; Linegar, C.; Newman, P. 1 year, 1000 km: The Oxford RobotCar dataset. *Int. J. Robot. Res.* **2017**, *36*, 3–15. [\[CrossRef\]](#)
55. McCormac, J.; Handa, A.; Leutenegger, S.; Davison, A.J. SceneNet RGB-D: 5M Photorealistic Images of Synthetic Indoor Trajectories with Ground Truth. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
56. Hua, B.S.; Pham, Q.H.; Nguyen, D.T.; Tran, M.K.; Yu, L.F.; Yeung, S.K. SceneNN: A Scene Meshes Dataset with aNNotations. In Proceedings of the 2016 IEEE International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 92–101. [\[CrossRef\]](#)
57. Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Nießner, M.; Savva, M.; Song, S.; Zeng, A.; Zhang, Y. Matterport3D: Learning from RGB-D Data in Indoor Environments. In Proceedings of the 2017 IEEE/CVF International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017.
58. Park, J.; Zhou, Q.Y.; Koltun, V. Colored Point Cloud Registration Revisited. In Proceedings of the 2017 IEEE/CVF International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 143–152. [\[CrossRef\]](#)
59. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D Semantic Parsing of Large-Scale Indoor Spaces. In Proceedings of the 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543. [\[CrossRef\]](#)
60. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. Semantic3d. net: A new large-scale point cloud classification benchmark. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **2017**, *IV-1/W1*, 91–98. [\[CrossRef\]](#)
61. Royndard, X.; Deschaud, J.E.; Goulette, F. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int. J. Robot. Res.* **2018**, *37*, 545–557. [\[CrossRef\]](#)
62. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Niessner, M. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In Proceedings of the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2432–2443. [\[CrossRef\]](#)
63. Dong, Z.; Liang, F.; Yang, B.; Xu, Y.; Zang, Y.; Li, J.; Wang, Y.; Dai, W.; Fan, H.; Hyypä, J.; et al. Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *163*, 327–342. [\[CrossRef\]](#)
64. Pham, Q.H.; Sevestre, P.; Pahwa, R.S.; Zhan, H.; Pang, C.H.; Chen, Y.; Mustafa, A.; Chandrasekhar, V.; Lin, J. A\*3D Dataset: Towards Autonomous Driving in Challenging Environments. *arXiv* **2019**, arXiv:1909.07541v1.
65. Chang, M.F.; Ramanan, D.; Hays, J.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; et al. Argoverse: 3D Tracking and Forecasting With Rich Maps. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8740–8749. [\[CrossRef\]](#)
66. Huang, X.; Wang, P.; Cheng, X.; Zhou, D.; Geng, Q.; Yang, R. The ApolloScape Open Dataset for Autonomous Driving and its Application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2702–2719. [\[CrossRef\]](#) [\[PubMed\]](#)

67. Xue, J.; Fang, J.; Li, T.; Zhang, B.; Zhang, P.; Ye, Z.; Dou, J. BLVD: Building A Large-scale 5D Semantics Benchmark for Autonomous Driving. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [\[CrossRef\]](#)
68. Patil, A.; Malla, S.; Gang, H.; Chen, Y.T. The H3D Dataset for Full-Surround 3D Multi-Object Detection and Tracking in Crowded Urban Scenes. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [\[CrossRef\]](#)
69. Houston, J.; Zuidhof, G.; Bergamini, L.; Ye, Y.; Chen, L.; Jain, A.; Omari, S.; Iglovikov, V.; Ondruska, P. One Thousand and One Hours: Self-driving Motion Prediction Dataset. *arXiv* **2020**, arXiv:2006.14480.
70. Mo, K.; Zhu, S.; Chang, A.X.; Yi, L.; Tripathi, S.; Guibas, L.J.; Su, H. PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 909–918. [\[CrossRef\]](#)
71. Hurl, B.; Czarnecki, K.; Waslander, S. Precise Synthetic Image and LiDAR (PreSIL) Dataset for Autonomous Vehicle Perception. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019. [\[CrossRef\]](#)
72. Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J.J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. The Replica Dataset: A Digital Replica of Indoor Spaces. *arXiv* **2019**, arXiv:1906.05797. [\[CrossRef\]](#)
73. Uy, M.A.; Pham, Q.H.; Hua, B.S.; Nguyen, D.T.; Yeung, S.K. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [\[CrossRef\]](#)
74. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
75. Zheng, J.; Zhang, J.; Li, J.; Tang, R.; Gao, S.; Zhou, Z. Structured3D: A Large Photo-Realistic Dataset for Structured 3D Modeling. In Proceedings of the 16th European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 519–535. [\[CrossRef\]](#)
76. Griffiths, D.; Boehm, J. SynthCity: A large scale synthetic point cloud. *arXiv* **2019**, arXiv:1907.04758.
77. Li, X.; Li, C.; Tong, Z.; Lim, A.; Yuan, J.; Wu, Y.; Tang, J.; Huang, R. Campus3D: A Photogrammetry Point Cloud Benchmark for Hierarchical Understanding of Outdoor Scene. Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; Volume 4, pp. 238–246. [\[CrossRef\]](#)
78. Varney, N.; Asari, V.K.; Graehling, Q. DALES: A Large-scale Aerial LiDAR Data Set for Semantic Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 717–726. [\[CrossRef\]](#)
79. Fu, H.; Cai, B.; Gao, L.; Zhang, L.X.; Wang, J.; Li, C.; Zeng, Q.; Sun, C.; Jia, R.; Zhao, B.; et al. 3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10913–10922. [\[CrossRef\]](#)
80. Selvaraju, P.; Nabail, M.; Loizou, M.; Maslioukova, M.; Averkiou, M.; Andreou, A.; Chaudhuri, S.; Kalogerakis, E. BuildingNet: Learning to Label 3D Buildings. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021. [\[CrossRef\]](#)
81. Deschaut, J.E.; Duque, D.; Richa, J.P.; Velasco-Forero, S.; Marcotegui, B.; Goulette, F. Paris-CARLA-3D: A Real and Synthetic Outdoor Point Cloud Dataset for Challenging Tasks in 3D Mapping. *Remote Sens.* **2021**, *13*, 4713. [\[CrossRef\]](#)
82. Lugo, G.; Li, R.; Chauhan, R.; Wang, Z.; Tiwary, P.; Pandey, U.; Patel, A.; Rombough, S.; Schatz, R.; Cheng, I. LiSurveying: A high-resolution TLS-LiDAR benchmark. *Comput. Graph.* **2022**, *107*, 116–130. [\[CrossRef\]](#)
83. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor Segmentation and Support Inference from RGBD Images. In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 746–760. [\[CrossRef\]](#)
84. Zhang, Z. Microsoft Kinect Sensor and Its Effect. *IEEE Multimed.* **2012**, *19*, 4–10. [\[CrossRef\]](#)
85. Lin, H.; Zheng, X.; Li, L.; Chao, F.; Wang, S.; Wang, Y.; Tian, Y.; Ji, R. Meta Architecture for Point Cloud Analysis. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–23 June 2022.
86. Lu, R.; Brilakis, I.; Middleton, C.R. Detection of Structural Components In Point Clouds of Existing Rc Bridges. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *34*, 191–212. [\[CrossRef\]](#)
87. Matterport. Pro2 3D Scanning Camera for High-Precision Imaging. Available online: <https://matterport.com/cameras/pro2-3d-camera> (accessed on 7 December 2022).
88. Ouster Inc. 3D LiDAR Sensors. Available online: <https://ouster.com/products/rev7/> (accessed on 9 December 2022).
89. Leica. Hochauflösende 3D-Laserscanner-Lösung. Available online: <https://leica-geosystems.com/de-de/products/laser-scanners/scanners/leica-scanstation-p40--p30> (accessed on 9 December 2022).
90. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [\[CrossRef\]](#)
91. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [\[CrossRef\]](#)
92. Anand, R.; Mehrotra, K.G.; Mohan, C.K.; Ranka, S. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Trans. Neural Netw.* **1993**, *4*, 962–969. [\[CrossRef\]](#)
93. Japkowicz, N. The Class Imbalance Problem: Significance and Strategies. In Proceedings of the 2000 International Conference on Artificial Intelligence IC-AI, Las Vegas, NV, USA, 26–29 June 2000.

94. Mahani, A.; Riad Baba Ali, A. Classification Problem in Imbalanced Datasets. In *Recent Trends in Computational Intelligence*; Sadollah, A., Sinha, T., Eds.; IntechOpen, Rijeka, Croatia, 2019. [CrossRef]
95. Lemaitre, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.* **2017**, *18*, 559–563.
96. Susan, S.; Kumar, A. The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art. *Eng. Rep.* **2021**, *3*, e12298. [CrossRef]
97. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 463–484. [CrossRef]
98. Batista, G.E.; Bazzan, A.L.; Monard, M.C. Balancing Training Data for Automated Annotation of Keywords: A Case Study. In Proceedings of the II Brazilian Workshop on Bioinformatics, Macaé, RJ, Brazil, 3–5 December 2003; pp. 10–18.
99. Junsomboon, N.; Phienthrakul, T. Combining Over-Sampling and Under-Sampling Techniques for Imbalance Dataset. In Proceedings of the 9th International Conference on Machine Learning and Computing (ICMLC), Singapore, 24–26 February 2017; pp. 243–247. [CrossRef]
100. Thomas, H.; Qi, C.R.; Deschud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [CrossRef]
101. Ma, X.; Qin, C.; You, H.; Ran, H.; Fu, Y. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. In Proceedings of the 2022 IEEE International Conference on Learning Representations (ICLR), Virtual-only Online Conference, 25–29 April 2022.
102. Chawla, N.V. Data Mining for Imbalanced Datasets: An Overview. In *Data Mining and Knowledge Discovery Handbook*; Springer: Cham, Switzerland, 2005; pp. 853–867. [CrossRef]
103. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **2018**, *106*, 249–259. [CrossRef]
104. scikit-learn.org Online. Multiclass Receiver Operating Characteristic (ROC). 2023. [https://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_roc.html](https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html) (accessed on 14 October 2022).
105. Xue, L.; Gao, M.; Xing, C.; Martín-Martín, R.; Wu, J.; Xiong, C.; Xu, R.; Niebles, J.C.; Savarese, S. ULIP: Learning Unified Representation of Language, Image and Point Cloud for 3D Understanding. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022.
106. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution on X-Transformed Points. In Proceedings of the Thirty-Second International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 828–838.
107. Wang, Q.; Shi, S.; Li, J.; Jiang, W.; Zhang, X. Window Normalization: Enhancing Point Cloud Understanding by Unifying Inconsistent Point Densities. *arXiv* **2022**, arXiv:2212.02287.
108. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.; Koltun, V. Point Transformer. In Proceedings of the ICCV International Conference on Computer Vision, Virtual, 23–28 August 2020.
109. Schult, J.; Engelmann, F.; Hermans, A.; Litany, O.; Tang, S.; Leibe, B. Mask3D for 3D Semantic Instance Segmentation. *arXiv* **2022**, arXiv:2210.03105.
110. Liang, Z.; Li, Z.; Xu, S.; Tan, M.; Jia, K. Instance Segmentation in 3D Scenes using Semantic Superpoint Tree Networks. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Virtual-only Online Conference, 11–17 October 2021; pp. 16259–16268. [CrossRef]
111. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]
112. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Int. J. Mach. Learn. Technol.* **2011**, *2*, 37–63.
113. Sasaki, Y. The truth of the F-measure. *Teach Tutor Mater* **2007**, *1*, 1–5.
114. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 6. [CrossRef] [PubMed]
115. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
116. Wang, L.; Liu, Y.; Zhang, S.; Yan, J.; Tao, P. Structure-Aware Convolution for 3D Point Cloud Classification and Segmentation. *Remote Sens.* **2020**, *12*, 634. [CrossRef]
117. Yang, H.; Shi, C.; Chen, Y.; Wang, L. Boosting 3D Object Detection via Object-Focused Image Fusion. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022. [CrossRef]
118. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the 2015 IEEE/CVF International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 945–953.
119. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928. [CrossRef]



120. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the 2015 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 07–12 June 2015; pp. 1912–1920. [\[CrossRef\]](#)
121. Xie, Y.; Tian, J.; Zhu, X.X. Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote. Sens. Mag.* **2020**, *8*, 38–59. [\[CrossRef\]](#)
122. Imad, M.; Doukhi, O.; Lee, D.J. Transfer Learning Based Semantic Segmentation for 3D Object Detection from Point Cloud. *Sensors* **2021**, *21*, 3964. [\[CrossRef\]](#)
123. Sun, R.; Zhu, X.; Wu, C.; Huang, C.; Shi, J.; Ma, L. Not All Areas Are Equal: Transfer Learning for Semantic Segmentation via Hierarchical Region Selection. In Proceedings of the 1019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
124. Robert, D.; Vallet, B.; Landrieu, L. Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5575–5584. [\[CrossRef\]](#)
125. Boulch, A.; Le Saux, B.; Audebert, N. Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks. In Proceedings of the 2017 Workshop on 3D Object Retrieval, Lyon, France, 23–24 April 2017. [\[CrossRef\]](#)
126. Wang, Y.; Shi, T.; Yun, P.; Tai, L.; Liu, M. PointSeg: Real-Time Semantic Segmentation Based on 3D LiDAR Point Cloud. *arXiv* **2018**, arXiv:1807.06288.
127. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018. [\[CrossRef\]](#)
128. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4213–4220. [\[CrossRef\]](#)
129. Yan, X.; Gao, J.; Zheng, C.; Zheng, C.; Zhang, R.; Cui, S.; Li, Z. 2DPASS: 2D Priors Assisted Semantic Segmentation on LiDAR Point Clouds. In Proceedings of the CVF European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022. [\[CrossRef\]](#)
130. Xu, C.; Wu, B.; Wang, Z.; Zhan, W.; Vajda, P.; Keutzer, K.; Tomizuka, M. SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation. In Proceedings of the 16th European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020. [\[CrossRef\]](#)
131. Huang, J.; You, S. Point cloud labeling using 3D Convolutional Neural Network. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2670–2675. [\[CrossRef\]](#)
132. Riegler, G.; Ulusoy, A.O.; Geiger, A. OctNet: Learning Deep 3D Representations at High Resolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [\[CrossRef\]](#)
133. Graham, B.; Engelcke, M.; van der Maaten, L. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018. [\[CrossRef\]](#)
134. Choy, C.; Gwak, J.; Savarese, S. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
135. Rosu, R.A.; Schütt, P.; Quenzel, J.; Behnke, S. LatticeNet: Fast spatio-temporal point cloud segmentation using permutohedral lattices. *Auton. Robot.* **2022**, *46*, 45–60. [\[CrossRef\]](#)
136. Zhong, Y.; Peng, H. Real-time Semantic 3D Dense Occupancy Mapping with Efficient Free Space Representations. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022. [\[CrossRef\]](#)
137. Zhong, X.; Pan, Y.; Behley, J.; Stachniss, C. SHINE-Mapping: Large-Scale 3D Mapping Using Sparse Hierarchical Implicit Neural Representations. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 8371–8377. [\[CrossRef\]](#)
138. Hua, B.S.; Tran, M.K.; Yeung, S.K. Pointwise Convolutional Neural Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018. [\[CrossRef\]](#)
139. Wang, S.; Suo, S.; Ma, W.C.; Pokrovsky, A.; Urtasun, R. Deep Parametric Continuous Convolutional Neural Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2589–2597. [\[CrossRef\]](#)
140. Boulch, A. ConvPoint: Continuous Convolutions for Point Cloud Processing. *Comput. Graph.* **2020**, *88*, 24–34. [\[CrossRef\]](#)
141. Klovov, R.; Lempitsky, V. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [\[CrossRef\]](#)
142. Hermosilla, P.; Ritschel, T.; Vázquez, P.P.; Vinacua, À.; Ropinski, T. Monte Carlo Convolution for Learning on Non-Uniformly Sampled Point Clouds. *ACM Trans. Graph.* **2018**, *37*, 1–12. [\[CrossRef\]](#)

143. Wu, W.; Qi, Z.; Fuxin, L. PointConv: Deep Convolutional Networks on 3D Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9613–9622. [\[CrossRef\]](#)
144. Liu, K.; Gao, Z.; Lin, F.; Chen, B.M. FG-Net: Fast Large-Scale LiDAR Point Clouds Understanding Network Leveraging Correlated Feature Mining and Geometric-Aware Modelling. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021. [\[CrossRef\]](#)
145. Tang, L.; Zhan, Y.; Chen, Z.; Yu, B.; Tao, D. Contrastive Boundary Learning for Point Cloud Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022. [\[CrossRef\]](#)
146. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [\[CrossRef\]](#)
147. Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [\[CrossRef\]](#)
148. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: A comprehensive review. *Comput. Soc. Netw.* **2019**, *6*, 1–23. [\[CrossRef\]](#)
149. Landrieu, L.; Simonovsky, M. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018. [\[CrossRef\]](#)
150. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2018**, *38*, 1–12. [\[CrossRef\]](#)
151. Zhiheng, K.; Ning, L. PyramNet: Point Cloud Pyramid Attention Network and Graph Embedding Module for Classification and Segmentation. *arXiv* **2019**, arXiv:1906.03299.
152. Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; Shan, J. Graph Attention Convolution for Point Cloud Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 10288–10297. [\[CrossRef\]](#)
153. Ma, Y.; Guo, Y.; Liu, H.; Lei, Y.; Wen, G. Global Context Reasoning for Semantic Segmentation of 3D Point Clouds. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020; pp. 2920–2929. [\[CrossRef\]](#)
154. Xie, L.; Furuhashi, T.; Shimada, K. Multi-Resolution Graph Neural Network for Large-Scale Pointcloud Segmentation. In Proceedings of the 2020 Conference on Robot Learning (CoRL), Virtual, 18 November 2020.
155. Li, Q.; Han, Z.; Wu, X.M. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 3538–3545.
156. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [\[CrossRef\]](#)
157. Li, G.; Müller, M.; Qian, G.; Delgadillo, I.C.; Abualshour, A.; Thabet, A.; Ghanem, B. DeepGCNs: Making GCNs Go as Deep as CNNs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *45*, 6923–6939. [\[CrossRef\]](#) [\[PubMed\]](#)
158. Dai, A.M.; Le, V. Q. Semi-supervised Sequence Learning. In Proceedings of the 2015 Conference on Advances in Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015.
159. Engelmann, F.; Kontogianni, T.; Hermans, A.; Leibe, B. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; Volume 3, pp. 716–724. [\[CrossRef\]](#)
160. Huang, Q.; Wang, W.; Neumann, U. Recurrent Slice Networks for 3D Segmentation of Point Clouds. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018. [\[CrossRef\]](#)
161. Ye, X.; Li, J.; Huang, H.; Du, L.; Zhang, X. 3D Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 11211, pp. 415–430. [\[CrossRef\]](#)
162. Zhao, Z.; Liu, M.; Ramani, K. DAR-Net: Dynamic Aggregation Network for Semantic Scene Segmentation. *arXiv* **2019**, arXiv:1907.12022.
163. Yang, J.; Lee, C.; Ahn, P.; Lee, H.; Yi, E.; Kim, J. PBP-Net: Point Projection and Back-Projection Network for 3D Point Cloud Segmentation. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24–29 October 2020. [\[CrossRef\]](#)
164. Engelmann, F.; Kontogianni, T.; Schult, J.; Leibe, B. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 4–8 September 2018; Volume 11131, pp. 395–409. [\[CrossRef\]](#)



165. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; pp. 539–546. [\[CrossRef\]](#)
166. Qiu, S.; Anwar, S.; Barnes, N. Semantic Segmentation for Real Point Cloud Scenes via Bilateral Augmentation and Adaptive Fusion. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021. [\[CrossRef\]](#)
167. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–7 December 2017; pp. 6000–6010. [\[CrossRef\]](#)
168. Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; Shen, C. Twins: Revisiting the Design of Spatial Attention in Vision Transformers. *arXiv* **2021**, arXiv:2104.13840.
169. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022. [\[CrossRef\]](#)
170. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021. [\[CrossRef\]](#)
171. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S.M. PCT: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [\[CrossRef\]](#)
172. Yang, J.; Zhang, Q.; Ni, B.; Li, L.; Liu, J.; Zhou, M.; Tian, Q. Modeling Point Clouds with Self-Attention and Gumbel Subset Sampling. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
173. Engel, N.; Belagiannis, V.; Dietmayer, K. Point Transformer. *IEEE Access* **2021**, *9*, 134826–134840. [\[CrossRef\]](#)
174. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the 2015 International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241. [\[CrossRef\]](#)
175. Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; Zhao, H. Point Transformer V2: Grouped Vector Attention and Partition-based Pooling. In Proceedings of the 36 Conference on Neural Information Processing Systems (NIPS), New Orleans, LA, USA, 28 November–9 December 2022. [\[CrossRef\]](#)
176. Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; Jia, J. Stratified Transformer for 3D Point Cloud Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022. [\[CrossRef\]](#)
177. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In Proceedings of the Thirty-First Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017. [\[CrossRef\]](#)
178. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *27*. [\[CrossRef\]](#)
179. Radenović, F.; Tolias, G.; Chum, O. Fine-tuning CNN Image Retrieval with No Human Annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1655–1668. [\[CrossRef\]](#)
180. Prakash, A.; Boochoon, S.; Brophy, M.; Acuna, D.; Cameracci, E.; State, G.; Shapira, O.; Birchfield, S. Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [\[CrossRef\]](#)
181. Uhr, M.B.F.; Felix, D.; Williams, B.J.; Krueger, H. Transfer of Training in an Advanced Driving Simulator: Comparison between Real World Environment and Simulation in a Manoeuvring Driving Task. In Proceedings of the Driving Simulation Conference, Dearborn, MI, USA, 8–10 October 2003.
182. Bewley, A.; Rigley, J.; Liu, Y.; Hawke, J.; Shen, R.; Lam, V.D.; Kendall, A. Learning to Drive from Simulation without Real World Labels. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [\[CrossRef\]](#)
183. Vincze, M.; Patten, T.; Christensen, H.I.; Nalpantidis, L.; Liu, M. (Eds.) *Computer Vision Systems*; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2021. [\[CrossRef\]](#)
184. Zhao, W.; Queralta, J.P.; Westerlund, T. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 1–4 December 2020. [\[CrossRef\]](#)
185. Xu, Y.; Tong, X.; Stilla, U. Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry. *Autom. Constr.* **2021**, *126*, 103675. [\[CrossRef\]](#)
186. Balado, J.; Martínez-Sánchez, J.; Arias, P.; Novo, A. Road Environment Semantic Segmentation with Deep Learning from MLS Point Cloud Data. *Sensors* **2019**, *19*, 3466. [\[CrossRef\]](#)

187. Michiels, T. How Transformers are Changing the Direction of Deep Learning Architectures: 2022 Embedded Vision Summit Sessions. Available online: <https://www.edge-ai-vision.com/2022/08/how-transformers-are-changing-the-direction-of-deep-learning-architectures-a-presentation-from-synopsys/> (accessed on 18 May 2022).
188. Zhang, Y.; Gong, K.; Zhang, K.; Li, H.; Qiao, Y.; Ouyang, W.; Yue, X. Meta-Transformer: A Unified Framework for Multimodal Learning. *arXiv* **2023**, arXiv:2307.10802. [\[CrossRef\]](#)
189. Keles, F.D.; Wijewardena, P.M.; Hegde, C. On The Computational Complexity of Self-Attention. *arXiv* **2022**, arXiv:2209.04881.
190. Wang, P.; Panda, R.; Hennigen, L.T.; Greengard, P.; Karlinsky, L.; Feris, R.; Cox, D.D.; Wang, Z.; Kim, Y. Learning to Grow Pretrained Models for Efficient Transformer Training. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Ruanda, 1–5 May 2023. [\[CrossRef\]](#)
191. Halber, M.; Shi, Y.; Xu, K.; Funkhouser, T. Rescan: Inductive Instance Segmentation for Indoor RGBD Scans. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [\[CrossRef\]](#)
192. Xiao, B.; Xiao, H.; Wang, J.; Chen, Y. Vision-based method for tracking workers by integrating deep learning instance segmentation in off-site construction. *Autom. Constr.* **2022**, *136*, 104148. [\[CrossRef\]](#)
193. Kang, K.S.; Cho, Y.W.; Jin, K.H.; Kim, Y.B.; Ryu, H.G. Application of one-stage instance segmentation with weather conditions in surveillance cameras at construction sites. *Autom. Constr.* **2022**, *133*, 104034. [\[CrossRef\]](#)
194. Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.A.A.K.; Elhoseiny, M.; Ghanem, B. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In Proceedings of the Thirty-sixth Conference on Neural Information Processing Systems (NIPS), New Orleans, LA, USA, 28 November–9 December 2022. [\[CrossRef\]](#)
195. Ran, H.; Liu, J.; Wang, C. Surface Representation for Point Clouds. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022. [\[CrossRef\]](#)
196. Fan, S.; Dong, Q.; Zhu, F.; Lv, Y.; Ye, P.; Wang, F.Y. SCF-Net: Learning Spatial Contextual Features for Large-Scale Point Cloud Segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 14499–14508. [\[CrossRef\]](#)
197. Ran, H.; Zhuo, W.; Liu, J.; Lu, L. Learning Inner-Group Relations on Point Clouds. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021. [\[CrossRef\]](#)
198. Park, C.; Jeong, Y.; Cho, M.; Park, J. Fast Point Transformer. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022. [\[CrossRef\]](#)
199. Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; Lu, C. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation. *arXiv* **2018**, arXiv:1807.00652.
200. Yan, X.; Zheng, C.; Li, Z.; Wang, S.; Cui, S. PointASNL: Robust Point Clouds Processing using Nonlocal Neural Networks with Adaptive Sampling. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [\[CrossRef\]](#)
201. Lin, Y.; Yan, Z.; Huang, H.; Du, D.; Liu, L.; Cui, S.; Han, X. FPConv: Learning Local Flattening for Point Convolution. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020. [\[CrossRef\]](#)
202. Landrieu, L.; Boussaha, M. Point Cloud Oversegmentation with Graph-Structured Deep Metric Learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
203. Boulch, A.; Puy, G.; Marlet, R. FKConv: Feature-Kernel Alignment for Point Cloud Convolution. In Proceedings of the Asian Conference on Computer Vision (ACCV), Kyoto, Japan, 30 November–4 December 2020; pp. 381–399. [\[CrossRef\]](#)
204. Jiang, L.; Zhao, H.; Liu, S.; Shen, X.; Fu, C.W.; Jia, J. Hierarchical Point-Edge Interaction Network for Point Cloud Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [\[CrossRef\]](#)
205. Hu, Z.; Zhen, M.; Bai, X.; Fu, H.; Tai, C.I. JSENet: Joint Semantic Segmentation and Edge Detection Network for 3D Point Clouds. In Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Volume 12365, pp. 222–239. [\[CrossRef\]](#)
206. Mazur, K.; Lempitsky, V. Cloud Transformers: A Universal Approach to Point Cloud Processing Tasks. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021. [\[CrossRef\]](#)
207. Zhang, Z.; Hua, B.S.; Yeung, S.K. ShellNet: Efficient Point Cloud Convolutional Neural Networks using Concentric Shells Statistics. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [\[CrossRef\]](#)
208. Mao, J.; Wang, X.; Li, H. Interpolated Convolutional Networks for 3D Point Cloud Understanding. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019. [\[CrossRef\]](#)
209. Zhao, H.; Jiang, L.; Fu, C.W.; Jia, J. PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)

210. Pan, L.; Chew, C.M.; Lee, G.H. PointAtrousGraph: Deep Hierarchical Encoder-Decoder with Point Atrous Convolution for Unorganized 3D Points. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020. [\[CrossRef\]](#)
211. Liu, J.; Ni, B.; Li, C.; Yang, J.; Tian, Q. Dynamic Points Agglomeration for Hierarchical Point Sets Learning. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7545–7554. [\[CrossRef\]](#)
212. Zhang, R.; Zeng, Z.; Guo, Z.; Gao, X.; Fu, K.; Shi, J. DSPoint: Dual-scale Point Cloud Recognition with High-frequency Fusion. *arXiv* **2021**, arXiv:2111.10332. [\[CrossRef\]](#)
213. Komarichev, A.; Zhong, Z.; Hua, J. A-CNN: Annularly Convolutional Neural Networks on Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
214. Chen, L.Z.; Li, X.Y.; Fan, D.P.; Wang, K.; Lu, S.P.; Cheng, M.M. LSANet: Feature Learning on Point Sets by Local Spatial Aware Layer. *arXiv* **2019**, arXiv:1905.05442. [\[CrossRef\]](#)
215. Zhao, L.; Tao, W. JSNet: Joint Instance and Semantic Segmentation of 3D Point Clouds. In Proceedings of the IAAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34. [\[CrossRef\]](#)
216. Wang, X.; Liu, S.; Shen, X.; Shen, C.; Jia, J. Associatively Segmenting Instances and Semantics in Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
217. Zeng, W.; Gevers, T. 3DContextNet: K-d Tree Guided Hierarchical Learning of Point Clouds Using Local and Global Contextual Cues. In Proceedings of the 2018 European Conference on Computer Vision Workshop (ECCVW), Munich, Germany, 8–14 September 2018; pp. 314–330. [\[CrossRef\]](#)
218. Peyghambarzadeh, S.M.; Azizmalayeri, F.; Khotanlou, H.; Salarpour, A. Point-PlaneNet: Plane kernel based convolutional neural network for point clouds analysis. *Digit. Signal Process.* **2020**, *98*, 102633. [\[CrossRef\]](#)
219. Xie, S.; Liu, S.; Chen, Z.; Tu, Z. Attentional ShapeContextNet for Point Cloud Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4606–4615. [\[CrossRef\]](#)
220. Sun, C.; Zheng, Z.; Wang, X.; Xu, M.; Yang, Y. Self-supervised Point Cloud Representation Learning via Separating Mixed Shapes. *IEEE Trans. Multimed.* **2021**. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.