



Review

# Graph Neural Networks in Point Clouds: A Survey

Dilong Li Chenghui Lu, Ziyi Chen \* Jianlong Guan, Jing Zhao and Jixiang Du

Fujian Key Laboratory of Big Data Intelligence and Security, Xiamen Key Laboratory of Computer Vision and Pattern Recognition, Xiamen Key Laboratory of Data Security and Blockchain Technology, College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China; scholar.dll@hqu.edu.cn (D.L.); 23013083012@stu.hqu.edu.cn (C.L.); 23013083003@stu.hqu.edu.cn (J.G.); 22014083071@stu.hqu.edu.cn (J.Z.); jxdu@stu.hqu.edu.cn (J.D.)

\* Correspondence: chenziyihq@hqu.edu.cn

**Abstract:** With the advancement of 3D sensing technologies, point clouds are gradually becoming the main type of data representation in applications such as autonomous driving, robotics, and augmented reality. Nevertheless, the irregularity inherent in point clouds presents numerous challenges for traditional deep learning frameworks. Graph neural networks (GNNs) have demonstrated their tremendous potential in processing graph-structured data and are widely applied in various domains including social media data analysis, molecular structure calculation, and computer vision. GNNs, with their capability to handle non-Euclidean data, offer a novel approach for addressing these challenges. Additionally, drawing inspiration from the achievements of transformers in natural language processing, graph transformers have propelled models towards global awareness, overcoming the limitations of local aggregation mechanisms inherent in early GNN architectures. This paper provides a comprehensive review of GNNs and graph-based methods in point cloud applications, adopting a task-oriented perspective to analyze this field. We categorize GNN methods for point clouds based on fundamental tasks, such as segmentation, classification, object detection, registration, and other related tasks. For each category, we summarize the existing mainstream methods, conduct a comprehensive analysis of their performance on various datasets, and discuss the development trends and future prospects of graph-based methods.



**Citation:** Li, D.; Lu, C.; Chen, Z.; Guan, J.; Zhao, J.; Du, J. Graph Neural Networks in Point Clouds: A Survey. *Remote Sens.* **2024**, *16*, 2518. <https://doi.org/10.3390/rs16142518>

Academic Editor: Sander Oude Elberink

Received: 30 April 2024

Revised: 21 June 2024

Accepted: 27 June 2024

Published: 9 July 2024



**Copyright:** © 2024 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:** graph neural networks; point cloud processing; graph convolutional neural network; graph attention; graph-based methods; classification; segmentation; registration; object detection

## 1. Introduction

With the rapid advancements in 3D data acquisition technologies, the applications and prevalence of 3D sensor technology have significantly expanded, and the cost of acquiring 3D data has significantly reduced. Three-dimensional sensors encompass a variety of technologies, including various 3D scanners like the Artec Eva, the FARO Focus series, and the GOM ATOS, which is notably used in industrial applications for generating dense point clouds. Other technologies include infrared sensors, RGB-D cameras such as the Microsoft Kinect and Intel RealSense, structured light sensors akin to those featured in the ASUS Xtion series, and LiDAR systems like the Velodyne VLP-16 and the Leica Geosystems BLK360. The 3D data captured by these sensors provide abundant geometric information, morphological features, and dimensional details. In terms of data representation, 3D data can be expressed in various formats, including meshes, depth images, volumetric grids, and point clouds. Among these, point clouds are widely used for their ability to intuitively retain the original geometric information of the 3D space without any form of discretization, making point clouds the preferred data representation format for applications in scene understanding.

In recent years, deep learning technologies have achieved significant advancements across various domains such as image recognition, natural language processing, and 3D

computer vision, also catalyzing rapid advancements in the deep learning approach applied to 3D point cloud analysis. Numerous studies have developed innovative methods for 3D shape classification, semantic segmentation, object detection and place recognition, registration, reconstruction, and completion of point clouds. However, in comparison to traditional structured data, the application of deep learning to point clouds presents unique challenges, primarily owing to their unstructured nature, computational and storage demands, and the sparsity and irregularity that complicate processing. These factors require models capable of managing large volumes of unfilled space and extracting useful features from irregular data formats. Additionally, the high computational costs associated with large-scale scenes pose significant barriers to deployment. Existing models often lack generalization capabilities, performing well on specific datasets but poorly in variable real-world environments. This underscores the gap between laboratory settings and practical applications, highlighting the need for robust models adaptable to diverse scenarios. To address these issues, researchers are exploring new architectures, learning strategies, and efficient data preprocessing methods. GNNs have emerged as a prominent approach in this area, leveraging the geometric structure of point clouds to maintain data integrity while enhancing feature extraction, learning efficiency, and model generalizability.

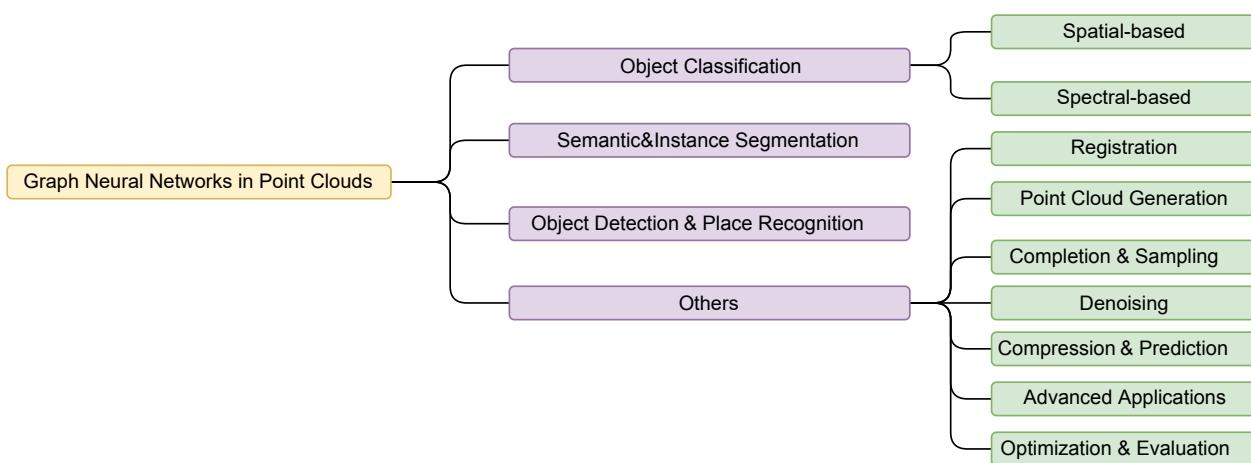
Graphs are data structures that model complex systems by representing nodes and their interconnections. They are crucial across various domains such as social network analysis, physical system simulations, and knowledge graph construction, offering powerful tools for tasks like node classification and link prediction. With advancements in machine learning, graph neural networks (GNNs) have been developed to tackle the complexities of graph data, such as the irregular and unordered nature of nodes. Graph convolutional networks (GCNs), a key subset of GNNs, are designed to address these challenges. Although there is no universally defined convolution operation for graphs, GCNs have introduced new concepts that enhance graph data analysis and advance graph analysis methodologies.

Given the inherently unstructured nature of point clouds, it is intuitive to process these data using graph structures. The rapid advancements in deep learning have spurred diverse approaches for point cloud processing. These methods include voxelization of the space, however, this incurs significant additional memory usage and computational overhead. Alternatively, PointNet [1] was a pioneering approach in the direct processing of point clouds, employing a shared multilayer perceptron (MLP) to extract features from each point individually and then aggregating these features uniformly using a symmetric function. However, PointNet struggles to capture local features as it processes each point independently, without inter-point message passing. PointNet++ [2] attempts to address this by introducing sampling and grouping operations hierarchically, but it still fundamentally lacks in learning local geometric structures of point clouds. Leveraging the successes of graphs in deep learning, the introduction of GNNs and GCNs addresses the deficiencies previously identified in point cloud data processing. By constructing edge relations among points within local neighborhoods, GNNs and GCNs enhance the capability to handle unstructured data, thus providing a novel perspective for the analysis and interpretation of point cloud data. The following points elaborate on why GNNs and GCNs are particularly well suited for processing point clouds:

- Unstructured data processing: GNNs and GCNs are naturally suited to handle unstructured data. The unstructured nature of point cloud data, where data points lack a fixed spatial arrangement, aligns with the free connection pattern of nodes in graph data. GNNs and GCNs can effectively learn feature representations of nodes in this unordered environment.
- Capturing local geometric structures: GNNs and GCNs capture local structural information through convolution operations defined on graphs. By transforming point clouds into a graph form—where points serve as nodes and spatial relationships between points define the edges—these networks effectively learn the local geometric characteristics of each point based on its relations with neighboring points.

- Permutation invariance: A fundamental characteristic of point cloud data is their permutation invariance, which stipulates that the 3D representation of a shape is unaffected by variations in the ordering of the data points. This attribute is critical for ensuring consistent interpretation and processing of point cloud data. GNNs and GCNs intrinsically support this permutation invariance, as the representation of a graph is independent of the ordering of its nodes. This inherent characteristic allows these networks to extract feature representations agnostic to point arrangement.
- Scalability and flexibility: GNNs and GCNs offer scalability in handling point clouds, allowing adaptation to various application needs through different graph construction strategies, such as k-nearest neighbors (KNN) or random walk. Moreover, these networks can flexibly adjust the weight on local and global information by tweaking parameters used for graph construction, such as the number of neighbors and different methods of constructing the graph.

We organized the methods related to the application of graph neural networks in 3D point clouds in recent years, forming this review. Review articles, such as [3], summarize the applications of GNNs in computer vision. Although the review also covers the use of GNNs in point clouds, it primarily provides an overview of GNN applications across the entire field of computer vision rather than focusing specifically on the point cloud domain. The review by Guo et al. [4] provides a comprehensive overview of recent advancements in deep learning methods for point clouds. It covers three main tasks, including 3D shape classification, 3D object detection and tracking, and 3D point cloud segmentation. For an in-depth and detailed analysis of the advantages, development, and future directions of GNN methods in various downstream tasks of point clouds, to our best knowledge, there are currently no such review articles available, which is also the motivation for writing this review. We systematically searched multiple databases, including Google Scholar, Web of Science, Scopus, arXiv, and IEEE Xplore, to ensure a comprehensive collection of the relevant literature. We categorized the collected literature by year of publication, focusing primarily on publications from the past seven years. The organization of this review is structured as follows: Section 2 introduces the fundamental theories of graph neural networks, along with some commonly used datasets and evaluation metrics associated with point cloud processing. In Section 3, we detail the applications of graph neural networks in various point cloud tasks, including classification, segmentation, object detection and place recognition, and other applications. Finally, in Section 4, we summarize and look forward to the prospects of the application of graph neural networks in point cloud tasks. The structure of this review is shown in Figure 1.



**Figure 1.** A taxonomy of a graph neural network in a point cloud.

## 2. Theoretical Background and Datasets

### 2.1. Related Theoretical Foundations

#### 2.1.1. Spectral-Based GCN

Spectral approaches define graph convolution using the Laplacian spectrum. For an undirected graph  $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$ , the adjacency matrix is  $\mathbf{A}$ , and the diagonal degree matrix is  $\mathbf{D}$ , with  $\mathbf{D}_{ii} = \sum_j^N \mathbf{A}_{ij}$ . The normalized Laplacian matrix of  $\mathbf{G}$  is  $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ , which can be decomposed as  $\mathbf{L} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T$ . Here,  $\mathbf{U}$  is the matrix of eigenvectors and  $\boldsymbol{\Lambda} = \text{diag}[\lambda_1, \dots, \lambda_N]$  is the diagonal matrix of eigenvalues.

Let  $\mathbf{Z} \in \mathbb{R}^{N \times d}$  ( $N = |\mathbf{V}|$ ) be the feature matrix of  $\mathbf{G}$  and  $\mathbf{z} \in \mathbb{R}^N$  be one of its columns ( $d = 1$ ). The graph Fourier transform of  $\mathbf{z}$  is  $\mathbf{F}(\mathbf{z}) = \mathbf{U}^T \mathbf{z}$ , and the inverse is  $\mathbf{F}^{-1}(\hat{\mathbf{z}}) = \mathbf{U} \hat{\mathbf{z}}$ , where  $\hat{\mathbf{z}} = \mathbf{F}(\mathbf{z})$ .

The convolution of  $\mathbf{z}$  with a filter  $\mathbf{g} \in \mathbb{R}^N$  is defined as

$$\mathbf{z} *_{\mathbf{G}} \mathbf{g} = \mathbf{F}^{-1}(\mathbf{F}(\mathbf{z}) \odot \mathbf{F}(\mathbf{g})) = \mathbf{U}((\mathbf{U}^T \mathbf{z}) \odot (\mathbf{U}^T \mathbf{g})), \quad (1)$$

where  $*_{\mathbf{G}}$  is the graph convolution operator and  $\odot$  is the Hadamard product. By defining  $\mathbf{g}_\theta = \text{diag}(\mathbf{U}^T \mathbf{g})$ , which depends on  $\boldsymbol{\Lambda}$ , we obtain

$$\mathbf{z} *_{\mathbf{G}} \mathbf{g} = \mathbf{U} \text{diag}(\mathbf{U}^T \mathbf{g}) \mathbf{U}^T \mathbf{z} = \mathbf{U} \mathbf{g}_\theta \mathbf{U}^T \mathbf{z}. \quad (2)$$

The Chebyshev spectral CNN (ChebNet) [5] uses Chebyshev polynomials to approximate the filtering operation  $\mathbf{g}_\theta$ . This can be expressed as  $\mathbf{g}_\theta \approx \sum_{i=0}^K \theta_i T_i(\tilde{\mathbf{L}})$ , where  $\tilde{\mathbf{L}} = \frac{2\mathbf{L}}{\lambda_{\max}} - \mathbf{I}$  is the scaled Laplacian matrix,  $\lambda_{\max}$  is the largest eigenvalue of  $\mathbf{L}$ , and  $\theta_i$  are learnable parameters. The Chebyshev polynomials  $T_i(\mathbf{z})$  is defined recursively:  $T_0(\mathbf{z}) = 1$  and  $T_1(\mathbf{z}) = \mathbf{z}$ , with  $T_i(\mathbf{z}) = 2\mathbf{z}T_{i-1}(\mathbf{z}) - T_{i-2}(\mathbf{z})$ . Consequently, the filtering operation is formulated as

$$\mathbf{z} *_{\mathbf{G}} \mathbf{g} \approx \mathbf{U} \left( \sum_{i=0}^K \theta_i T_i(\tilde{\mathbf{L}}) \right) \mathbf{U}^T \mathbf{z} \approx \sum_{i=0}^K \theta_i T_i(\tilde{\mathbf{L}}) \mathbf{z}. \quad (3)$$

Graph convolutional networks (GCNs) [6] introduced a first-order approximation of ChebNet ( $K = 1$ ). A GCN iteratively aggregates information from neighbors, and the feed-forward propagation for node  $v_i$  is defined as

$$\mathbf{h}_i^{(l+1)} = \sigma \left( \sum_{v_j \in \mathbf{N}(v_i) \cup \{v_i\}} \hat{a}(v_i, v_j) \mathbf{W}^{(l)} \mathbf{h}_j^{(l)} \right), \quad (4)$$

where  $\sigma(\cdot)$  is a nonlinear activation function,  $\hat{a}(v_i, v_j)$  denotes the renormalized adjacency matrix  $\hat{\mathbf{A}}$ , and  $\mathbf{W}^{(l)}$  is a learnable transformation matrix in the  $l$ -th layer.

#### 2.1.2. Spatial-Based GCN

GraphSAGE [7] is a general inductive framework that updates node states by sampling and aggregating hidden states from a fixed number of local neighbors. It performs graph convolutions in the spatial domain. Formally, this is expressed as

$$\mathbf{h}_{\mathbf{N}_s(v_i)}^{(l+1)} = \text{Aggregator}_{l+1} \left( \left\{ \mathbf{h}_j^{(l)}, \forall v_j \in \mathbf{N}_s(v_i) \right\} \right), \quad (5)$$

$$\mathbf{h}_i^{(l+1)} = \sigma \left( \mathbf{W}^{(l+1)} \cdot \left[ \mathbf{h}_i^{(l)} \oplus \mathbf{h}_{\mathbf{N}_s(v_i)}^{(l+1)} \right] \right), \quad (6)$$

where  $\mathbf{N}_s(v_i)$  is a subset of nodes sampled from the full neighborhood  $\mathbf{N}(v_i)$ , and  $\oplus$  denotes the concatenation operator. The aggregation function  $\text{Aggregator}_l$  can be a mean, LSTM, or pooling aggregator.

### 2.1.3. Graph Attention Networks

Graph attention networks (GATs) [8] introduced a self-attention mechanism to learn dynamic weights between connected nodes, updating the hidden state of a node by attending to its neighbors:

$$\mathbf{h}_i^{(l+1)} = \sigma \left( \sum_{v_j \in \mathbf{N}(v_i) \cup \{v_i\}} \alpha_{ij} \mathbf{W}^{(l)} \mathbf{h}_j^{(l)} \right), \quad (7)$$

$$\alpha_{ij} = \frac{\exp \left( \text{LeakyReLU} \left( \mathbf{a}_l^T \left[ \mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \oplus \mathbf{W}^{(l)} \mathbf{h}_j^{(l)} \right] \right) \right)}{\sum_{v_k \in \mathbf{N}(v_i)} \exp \left( \text{LeakyReLU} \left( \mathbf{a}_l^T \left[ \mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \oplus \mathbf{W}^{(l)} \mathbf{h}_k^{(l)} \right] \right) \right)}, \quad (8)$$

where  $\alpha_{ij}$  is the attention weight and  $\mathbf{a}$  is a learnable parameter vector. To enhance model capacity and stabilize self-attention, GAT employs multihead self-attention.

### 2.1.4. Graph Transformers for Point Cloud

The point transformer [9] employs a local vector self-attention mechanism tailored for point cloud analysis. In contrast, the point cloud transformer [10] implements a global attention scheme. Defined as follows, the local vector self-attention block of the point transformer combines the elements

$$\mathbf{h}_i^{(l+1)} = \sum_{v_j \in \mathbf{N}_s(v_i)} \rho \left( \gamma \left( \mathbf{W}_{\mathbf{Q}}^{(l+1)} \mathbf{h}_i^{(l)} - \mathbf{W}_{\mathbf{K}}^{(l+1)} \mathbf{h}_j^{(l)} + \delta \right) \odot \left( \mathbf{W}_{\mathbf{V}}^{(l+1)} \mathbf{h}_j^{(l)} + \delta \right) \right), \quad (9)$$

where  $\mathbf{W}_{\mathbf{Q}}$ ,  $\mathbf{W}_{\mathbf{K}}$ , and  $\mathbf{W}_{\mathbf{V}}$  are shared parameter matrices for computing the query, key, and value in the attention-based aggregation. The operator  $\odot$  signifies element-wise multiplication, and  $\delta$  denotes a position-encoding function. The functions  $\gamma$  and  $\rho$  represent a nonlinear mapping (e.g., MLP) and a normalization function (e.g., softmax), respectively.

Recent developments include the fast point transformer [11], which introduces a lightweight local self-attention architecture using voxel hashing to enhance efficiency significantly. The stratified transformer [12] samples distant points as additional keys to expand the receptive field, thereby effectively modeling long-range dependencies. The point transformer v2 [13] proposed group vector attention, strengthening the position information for attention by an additional position-encoding multiplier and designing novel and lightweight partition-based pooling methods. The point transformer v3 [14] replaces the precise neighbor search by KNN with an efficient serialized neighbor mapping, organizing point clouds according to specific patterns.

## 2.2. Datasets and Evaluation Metrics

### 2.2.1. Datasets

A comprehensive collection of datasets was compiled to evaluate the performance of deep learning algorithms in various 3D point cloud applications, as outlined in Table 1. The datasets are categorized into classification, segmentation, object detection, and place recognition domains. Classification datasets are divided into synthetic (complete objects, no occlusions) and real-world (partial occlusions, background noise). Object detection datasets differentiate between indoor scenes and outdoor urban environments. Segmentation datasets, gathered using mobile, aerial, and terrestrial laser scanners and RGB-D cameras, address challenges such as environmental interference, shape integrity, and class imbalance. In the past five years, deep learning applications in point cloud data processing have flourished, prompting the release of numerous public datasets that have advanced research significantly. Notable datasets include ModelNet [15] for 3D object classification and detection with CAD models, ScanObjectNN [16] for real-world 3D object recognition with noise and occlusions, and ShapeNet [17] for classification, segmentation, and retrieval of 3D shapes. PartNet [17] offers detailed annotations for fine-grained segmentation and recognition. S3DIS [18] and ScanNet [19] support semantic segmentation and scene understanding

from indoor RGB-D videos, while Semantic3D [20] and KITTI [21] focus on large-scale outdoor scenes and autonomous driving benchmarks, respectively. SemanticKITTI [22] and Toronto-3D [23] extend capabilities with detailed LiDAR data and multiattribute point clouds for semantic segmentation. These datasets provide a standardized foundation for developing and evaluating point cloud deep learning techniques, fueling advancements in 3D computer vision.

### 2.2.2. Evaluation Metrics

To thoroughly evaluate the effectiveness of different methods across a range of point cloud understanding tasks, researchers have proposed a series of evaluation metrics [18–20,22–24].

In the realm of classification, the commonly employed metrics include overall accuracy (OA) and mean class accuracy (mAcc). OA quantifies the proportion of accurately classified test instances, defined as

$$OA = \sum_{i=0}^M \frac{c_{ii}}{\sum_{j=0}^M c_{ij}}, \quad (10)$$

mAcc reflects average accuracy across different shape categories, together assessing the model's generalization ability, given by

$$mAcc = \frac{1}{M+1} \sum_{i=0}^M \frac{c_{ii}}{\sum_{j=0}^M c_{ij}}. \quad (11)$$

In the field of 3D object detection, average precision (AP) stands out as a pivotal metric, derived from the area under the precision–recall curve. This metric provides a thorough evaluation of algorithms across various thresholds, represented by

$$AP = \int_0^1 p(r) dr. \quad (12)$$

where  $p(r)$  is the precision at recall  $r$ .

In the context of segmentation tasks, metrics such as overall accuracy (OA), mean intersection over union (mIoU), and mean class accuracy (mAcc) are employed to assess segmentation outcomes. mIoU measures the overlap between predicted and actual segments, indicating fine-grained performance, defined as

$$mIoU = \frac{1}{M+1} \sum_{i=0}^M \left( \frac{c_{ii}}{\sum_{j=0}^M (c_{ij} + c_{ji} - c_{ii})} \right), \quad (13)$$

For instance segmentation, mean average precision (mAP) assesses efficiency in identifying and localizing individual entities, calculated as follows:

$$mAP = \text{Average over classes} \left( \int_0^1 p(r) dr \right). \quad (14)$$

**Table 1.** This table surveys datasets for classification, 3D object detection, and segmentation. It includes the publication year, training and testing samples, number of classes, data types, sensors used, and other application scenarios. Cls: classification, SemSeg: semantic segmentation, InsSeg: instance segmentation, PanSeg: panoramic segmentation, PartSeg: part segmentation, Det: object detection, Gen: point cloud generation, Recon: point cloud reconstruction, Tracking: object tracking.

Datasets Mainly Used in 3D Shape Classification							
Name and Reference	Year	Training	Test	Sample	Classes	Type	Tasks
McGill Benchmark [25]	2008	304	152	456	19	Synthetic	Cls
Sydney Urban Objects [26]	2013	-	-	588	14	Real world	Cls
ModelNet10 [15]	2015	3991	605	4899	10	Synthetic	Cls
ModelNet40 [15]	2015	9843	2468	12,311	40	Synthetic	Cls
ShapeNet [17]	2015	-	-	51,190	55	Synthetic	Cls/SemSeg/PartSeg
ScanNet [19]	2017	9677	2606	12,283	17	Real world	SemSeg
ScanObjectNN [16]	2019	2321	581	2902	15	Real world	Cls
Datasets Mainly Used in Point Cloud Semantic Segmentation							
Name and Reference	Year	Points	Classes	Scans	Spatial Size	Sensors	Tasks
ISPRS [27]	2012	1.6 M	5(44)	17	-	ALS	Cls/SemSeg
Paris-rue-Madame [28]	2014	20 M	17	2	-	MLS	Cls/SemSeg/Det
IQmulus [29]	2015	300 M	8(22)	10	-	MLS	Cls/SemSeg
ScanNet [19]	2017	-	20(20)	1513	8 × 4 × 4	RGB-D	Cls/SemSeg/InsSeg/Det
S3DIS [18]	2017	273 M	13(13)	272	10 × 5 × 5	Matterport	SemSeg/InsSeg/Det
Semantic3D [20]	2017	4000 M	8(9)	15/15	250 × 260 × 80	TLS	SemSeg
Paris-Lille-3D [24]	2018	143 M	9(50)	3	200 × 280 × 30	MLS	SemSeg
SemanticKITTI [22]	2019	4549 M	25(28)	23,201/20,351	150 × 100 × 10	MLS	SemSeg/Compl
Toronto-3D [23]	2020	78.3 M	8(9)	4	260 × 350 × 40	MLS	SemSeg
DALES [30]	2020	505 M	8(9)	40	500 × 500 × 65	ALS	SemSeg/PanSeg
SemanticPOSS [31]	2020	216 M	14	-	-	MLS	SemSeg/PanSeg
OpenGF [31]	2021	542.1 M	-	-	-	ALS	SemSeg
STPLS3D [32]	2022	216 M	18/14	-	-	ALS	SemSeg/PanSeg
HRHD-HK [33]	2023	273 M	7	-	-	ALS	SemSeg
ARCH2S [34]	2024	5 M × 5	14	-	-	-	SemSeg/Gen/Recon
FRACTAL [35]	2024	9621 M	7	-	-	ALS	SemSeg
Datasets Mainly Used in 3D Object Detection and Place Recognition							
Name and Reference	Year	Scenes	Classes	Frames	3D Boxes	Scene Type	Tasks
KITTI [21]	2012	22	8	15 K	200 K	Urban (driving)	Det
SUN RGB-D [36]	2015	47	37	5 K	65 K	Indoor	SemSeg/Det
ScanNetV2 [19]	2018	1.5 K	18	-	-	Indoor	Cls/SemSeg/InsSeg/Det
H3D [37]	2019	160	8	27 K	1.1 M	Urban (driving)	Det
Argoverse [38]	2019	113	15	44 K	993 K	Urban (driving)	Det/Tracking
A * 3D [39]	2019	-	7	39K	230 K	Urban (driving)	Det
Waymo Open [40]	2020	1 K	4	200 K	12 M	Urban (driving)	Det/Tracking
nuScenes [41]	2020	1 K	23	40 K	1.4 M	Urban (driving)	SemSeg/Det/Tracking
RadarScence [42]	2021	-	-	832 K	1.4 M	Urban (driving)	Det/Tracking

**Table 1.** *Cont.*

Datasets Mainly Used in 3D Object Detection and Place Recognition							
Name and Reference	Year	Scenes	Classes	Frames	3D Boxes	Scene Type	Tasks
aiMotive [43]	2023	176	14	26,583	-	Urban (driving)	Det
UT Campus Object [44]	2023	1 K	53	5 K	130 M	Urban	SemSeg/Det
Dual Radar [45]	2023	-	-	10 K	-	Urban (driving)	Det/Tracking

### 3. Graph Methods in Point Cloud Tasks

This section categorizes the applications of graph methods in the point cloud domain into several common tasks, including point cloud classification, segmentation, and object recognition, along with additional applications.

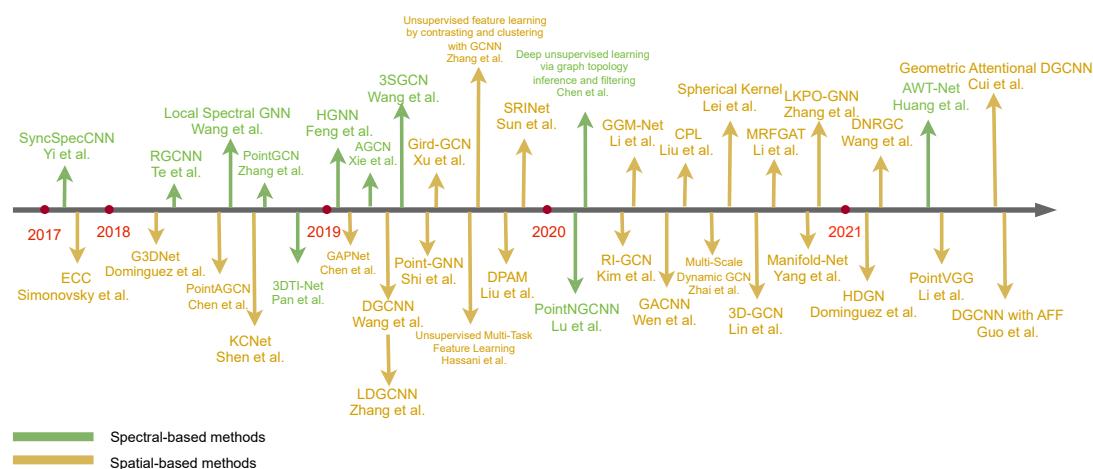
In point clouds, GNN methods primarily adopt spatial-based approaches, with only a few spectral-based methods employed in classification tasks. These spectral-based methods are briefly introduced in Section 3.1.

#### 3.1. Classification

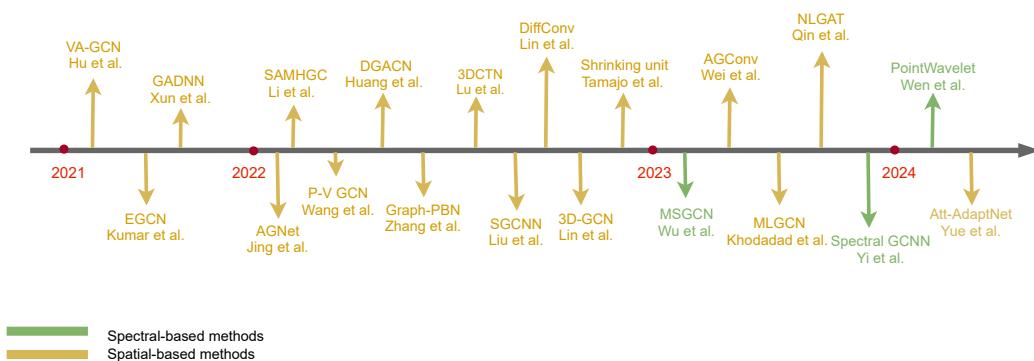
Methods for classification tasks typically begin by learning representations for individual points, followed by implementing an aggregation strategy to derive a comprehensive global embedding from the complete point cloud. This embedding is then used in several fully connected layers for classification. GNN methods fall into two main categories based on signal processing domains: spectral and spatial. In spectral domain approaches, graph convolutions are defined through spectral filtering, which is enabled by the eigen-decomposition of the graph's Laplacian matrix, utilizing spectral graph theory to project node features into the spectral space. After performing filtering operations, features are projected back to the original space, which is suitable for capturing global graph structural information. Spatial methods work directly on the graph's spatial structure by aggregating feature information from neighboring nodes and are divided into graph convolutional methods, graph attention networks, and dynamic graph updating techniques. A chronological overview of publications on GNNs in classification tasks over recent years is shown in Figures 2 and 3.

Graph-based spectral methods [6] have significantly advanced point cloud analysis. RGCNN [46] updates the graph's Laplacian matrix at each layer to maintain comprehensive connectivity and incorporates graph signal smoothing in the loss function to enhance feature homogeneity among adjacent vertices. AGCN [47] addresses diverse graph topologies by utilizing a learnable distance metric in its SGC-LL layer, facilitating adaptive topology adjustments. HGNN [48] extends this concept by applying spectral convolution on hypergraphs through hyperedge layers, enabling the capture of complex multidimensional relationships.

Local spectral graph convolutions [49] proposed by Wang et al. focus on KNN graphs, optimizing feature learning by avoiding the precomputation of Laplacian matrices. They introduced recursive clustering and pooling strategies, significantly enhancing local feature extraction capabilities. PointGCN [50] and 3DTI-Net [51] employ spectral convolution techniques using KNN graphs; PointGCN utilizes Gaussian kernel-weighted edges and Chebyshev polynomial filters for detailed feature mapping, while 3DTI-Net offers robustness against geometric transformations by learning from relative distances, thus enhancing classification accuracy.



**Figure 2.** Chronological overview of the most relevant spectral-based and spatial-based methods from 2017 to 2021.



**Figure 3.** Chronological overview of the most relevant spectral-based and spatial-based methods from 2021 to present.

### 3.1.1. Spectral-Based Methods

In 3SGCN [52], multispectral point clouds are modeled through distinct spectral and spatial graphs, combined to produce classification maps that are smooth and devoid of noise artifacts—essential for remote sensing applications. SyncSpecCNN [53] introduced an innovative spectral transformation network with a dilated convolutional kernel that enables effective weight sharing and spectral multiscale analysis across diverse shapes and sizes within a single framework. Lastly, PointNGCNN [54] by Lu et al. and the deep unsupervised learning model by Chen et al. illustrate the applications of neighborhood graph filters and graph topology inference, respectively. PointNGCNN employs Chebyshev polynomials for filtering to enhance object recognition and segmentation capabilities. Chen et al. [55] introduced a model that is incorporated within an autoencoder. It utilizes a graph topology inference module and a graph filtering module to optimize 3D point cloud reconstruction, demonstrating superior performance in tasks such as visualization and transfer classification. The authors of [56] present the adaptive wavelet transformer network (AWT-Net), which uses multiscale wavelet decomposition and transformers to enhance 3D shape features through recursive classification and feature integration, achieving competitive performance in 3D shape classification and segmentation. Furthermore, PointWavelet [57] employs a learnable graph wavelet transform for 3D point cloud data processing in the spectral domain, enhancing structural representation learning through multiscale spectral graph convolutions. Despite its efficiency in training and effectiveness across various datasets, it faces challenges with highly irregular data due to its dynamic local graph connection approach. Yi et al. [58] proposed a spectral graph convolutional

neural network (GCNN) approach for graph classification, integrating innovative edge feature schemes and additional layers between convolution layers, dynamically learning edge features and optimizing graph structures to address the limitations of traditional GCNNs. Wu et al. [59] presented an adaptive graph neural network method that effectively handles the irregularities of 3D point clouds by modeling spatial relationships at various scales. Using a self-adaptive convolution kernel based on Chebyshev polynomials, the approach enhances computational efficiency and outperforms existing methods in classification and retrieval tasks. PointAGCN [60] presents an adaptive spectral graph convolutional neural network; utilizes localized graph convolutions to autonomously extract local geometric features, circumventing the need for manually crafted filters and avoiding empirical design pitfalls, enabling dynamic updates to the graph topology across layers; and incorporates a novel graph pooling strategy that streamlines computations and enhances performance. The partial accuracies of the essential methods are presented in Table 2.

**Table 2.** Comparative analysis of 3D point cloud classification and segmentation method performance across ModelNet, ShapeNet, ScanNet, S3DIS, and other datasets.

Method	Datasets	Classification Results		Datasets	Segmentation Results		
		mA	OA		Cat mIoU	Ins mIoU	OA
RGCNN [46]	ModelNet40	87.3	90.5	ShapeNet	84.3	-	-
AGCN [47]	ModelNet40	90.7	92.6	ShapeNetPart	82.6	85.4	-
				S3DIS(6-fold)	56.63	-	84.13
HGNN [48]	ModelNet40	96.7	-	-	-	-	-
LSConv [49]	ModelNet40	-	91.5	ShapeNet	85.4	-	-
				ScanNet	85.4	-	84.8
PointGCN [50]	ModelNet40	86.19	89.27	-	-	-	-
3DTI-Net [51]	ModelNet40	86.	91.7	ShapeNet	84.9	-	-
SyncSpecCNN [53]	-	-	-	ShapeNet	84.74	-	-
PointNGCNN [54]	ModelNet40	-	92.8	ShapeNetPart	82.4	85.6	-
				S3DIS	-	-	87.3
				ScanNet	-	-	84.9
GTIF [55]	ModelNet40	-	89.55	-	-	-	-
AWT-Net [56]	ModelNet40	-	93.9	-	-	-	-
	ScanObjectNN	-	88.5	-	-	-	-
PointWavelet [57]	ModelNet40	91.1	94.3	ShapeNetPart	85.2	87.0	-
	ScanObjectNN	85.8	87.7	S3DIS(Area5)	71.3	-	-
MSGCN [59]	ModelNet40	-	92.5	-	-	-	-
	ShapeNetCore	-	86.6	-	-	-	-
	nuScenes	-	74.1	-	-	-	-
PointAGCN [60]	ModelNet40	-	91.4	ShapeNet	85.6	-	-
				S3DIS	52.3	-	-

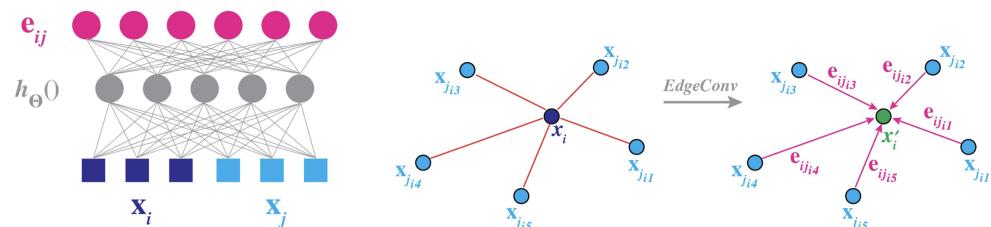
### 3.1.2. Spatial-Based Methods

These approaches execute operations within the spatial domain, including convolution and pooling, applied to spatial neighbors through a multilayer perceptron (MLP). Subsequently, they produce a coarsened graph by aggregating information from these neighbors. In the graph representation, each vertex typically holds attributes such as coordinates, laser intensity, or color, while each edge characteristically displays geometric properties associated with the connections between two vertices. As a pioneering effort, Simonovsky et al. connected each graph vertex to all its neighbors with directed edges and introduced edge-conditioned convolution (ECC) [61], which uses an MLP to generate filters. Graph coarsening is accomplished through the application of VoxelGrid, and max

pooling is utilized to aggregate information from neighborhoods. Considering that convolution on traditional structured data does not effectively apply to irregular data, Xu et al., proposed SpiderCNN [62], which adapts regular grid convolution for irregular point sets through SpiderConv units, parameterizing a set of convolutional filters to capture local geodesic information with a simple step function and ensure expressiveness with a Taylor polynomial. This model inherits the multiscale hierarchical architecture of classic CNNs, enabling deep semantic feature extraction.

Dominguez et al. proposed G3DNet [63], a method that transforms 3D point clouds into G3D feature vectors using graph convolution and clustering techniques adapted from traditional image CNNs. Shen et al. advanced this domain with KCNet [64], which enhances the semantic learning of 3D point clouds through kernel correlation and graph pooling operations, capturing finer local structural information compared to previous methods. Chen et al. introduced GAPNet [65], an architecture that embeds a graph attention mechanism within stacked MLP layers to more efficiently learn relationships between points and their neighborhoods. The core innovation of GAPNet, the GAPLayer, captures contextual attention features by allocating different attention weights to each point's neighborhood, with multihead attention mechanisms and attention pooling layers aggregating diverse feature representations to enhance network robustness.

In DGCNN [66], the graph is reconstructed in the feature space and dynamically updated following each network layer. The core module EdgeConv, shown in Figure 4, utilizes an MLP for feature learning on each edge, applying a channel-wise symmetric aggregation function (max pooling) to the edge features of each point's neighbors. LDGCNN [67], improving upon DGCNN [66], omits the transformation network and connects hierarchical features from different layers to enhance performance and reduce model size. It incorporates skip connections to aggregate dynamic graph features across layers, effectively learning valuable edge vector features and preventing vanishing gradients. Guo et al. [68] improved DGCNN [66] by integrating global and local features through an adaptive feature fusion module, employing multiscale transformations and residual blocks for enhanced performance. Geometric attentional dgcNN [69] adapts classical CNNs with a geometric attentional EdgeConv that extracts attributes and dynamically constructs graphs in both geometric and feature spaces.



**Figure 4.** Illustration of EdgeConv.

Inspired by DGCNN [66], Hassani and Haley introduced an unsupervised multi-task model [70] using a multiscale graph-based encoder to co-learn point and shape features through clustering, reconstruction, and self-supervised classification tasks. Similarly, Zhang et al. [71] utilized GCNNs for contrasting parts and clustering objects with the models ContrastNet and ClusterNet for feature verification and clustering prediction, respectively. Liu et al. proposed a graph convolution-based dynamic points agglomeration module (DPAM) [72] to simplify the point aggregation process into a single step by learning an aggregation matrix through graph convolution and applying matrix multiplication with the point feature matrix. Based on the PointNet architecture, this method creates a hierarchical learning structure by stacking multiple DPAMs, dynamically utilizing relationships between points to aggregate them in the semantic space. Sun et al. proposed SRINet [73], a method aimed at learning strictly rotation-invariant representations from point clouds. SRINet explores the local structure of point clouds through local graph convolution and two

types of graph downsampling operations, introducing point projection features to achieve rotation invariance. Utilizing a PointNet-based backbone to extract global features and perceiving local shape structures through graph aggregation operations, SRINet extracts strictly rotation-invariant representations. Additionally, the study introduced an efficient keypoint descriptor that assigns varying responses to each point, aiding in recognizing the overall geometric shape.

Bazazian and Nahat developed the dynamic capsule graph convolutional network (DCG-Net) [74] for point clouds, employing a dynamic capsule-based graph updated via dynamic routing at each convolutional layer to significantly enhance feature aggregation and point cloud analysis performance. Kim et al., introduced RI-GCN [75], designed to robustly handle rotations and other geometric transformations without explicit data augmentation, constructing a hierarchical rotation-invariant descriptor from multilevel graph convolutional abstractions and leveraging stochastic sampling of 3D points for effective feature representation. Wen et al. presented GACNN [76], integrating global and local attention mechanisms to assess global contextual information and local structural features through dynamic learning of spatial relationships and density variations among neighboring points, thus enhancing the accuracy of classification in airborne LiDAR data. Li et al. introduced GGM-Net [77], utilizing first and second geometric moments of local point sets to explicitly encode local geometric structures through GGM convolution operations in a directed graph, advancing the perception of surface geometric properties and increasing robustness to input variations.

Nezhadarya et al. proposed the critical points layer (CPL) [78] for adaptive downsampling, aiming to selectively preserve significant points in point clouds. This method implements a global downsampling mechanism that integrates seamlessly with graph-based convolutions within their CP-Net framework, enhancing computational efficiency. Zhai et al. introduced a novel multiscale dynamic GCN [79] that employs farthest point sampling and multiscale KNN grouping to comprehensively cover and analyze point clouds. This model utilizes EdgeConv operations to aggregate local features effectively, thereby improving classification performance. Lei et al. [80] developed a spherical kernel for efficient graph convolution in 3D point clouds, systematically quantifying the local 3D space and reducing computational complexity by avoiding the generation of edge-dependent filters. Lin et al. introduced 3D-GCN [81], which processes point cloud data through learnable deformable convolution kernels, defining a graph max pooling mechanism to extract structural information at various scales while maintaining translational and scale invariance. 3D-GCN learns the shape and weights of the convolution kernels during training, using directional vectors and similarity measures to enable effective feature extraction from unordered 3D point cloud data.

Li et al. presented the multiscale receptive fields graph attention network (MRF-GAT) [82], employing multiscale attention modules to concentrate on local details and enhance feature extraction. This approach effectively integrates edge and neighbor information for improved feature representation. In [83], Yang and Gao combined manifold learning with graph neural networks to create Manifold-Net, enhancing geometric continuity representation and overall network performance. This model incorporates local linear embedding (LLE) and manifold projection (MP) modules to transform 3D point features into a more manageable 2D space, maintaining structural integrity through a graph based on KNN. Wang et al. [84] introduced a novel end-to-end model leveraging graph convolutional networks (GCNs) to manage pose variations in three-dimensional point cloud data. Initially, the point cloud is represented using spherical coordinates rather than the traditional Cartesian coordinates, simplifying both computation and representation. A pose auxiliary network is then developed to estimate rotational changes within the point cloud. This network, in conjunction with a GCN, effectively classifies point clouds.

Zhang et al. introduced LKPO-GNN [85], which transforms unordered 3D point clouds into ordered sequences, forming local topological structures using multidirectional KNN, and represents them globally through graph neural networks. HDGN [86] employs a directional KNN graph and hard directional graph convolution to reduce computational costs by utilizing single weight matrices in convolutions. Wang et al. proposed the deep normalized Reeb graph convolution (DNRGC) [87] to capture the topologies of point clouds and enhance classification accuracy via normalized graph convolutions. Li et al. developed PointVGG [88], which expands the receptive field and aggregates detailed geometric information in point clouds by applying point convolution and pooling, integrated with graph structures for advanced feature learning. Srivastava and Gaurav [89] refined graph construction by including significant local geometric data and nonlinear projections through MLPs. To enhance the classification accuracy of airborne multispectral LiDAR (MS-LiDAR) point clouds, Zhao et al. proposed the feature reasoning-based graph convolution network (FR-GCNet) [90]. FR-GCNet performs semantic labeling on all points directly, investigating representative features on both global and local scales, and utilizes a GCN with a global reasoning unit to reveal spatial relationships and capture global context features, while integrating a local reasoning unit to dynamically learn edge features of each local graph and assign attention weights. Xu et al. introduced Grid-GCN [91], a method designed for fast and scalable learning of point clouds, based on the coverage-aware grid query (CAGQ) that enhances spatial coverage and reduces theoretical time complexity by utilizing the efficiency of grid space. This method also includes a grid context aggregation (GCA) module, which performs graph convolution by establishing local graphs on each point cluster and aggregating local contextual information. Hu et al. developed the vector attention graph convolution network (VA-GCN) [92] using the vector attention convolution (VAConv) module, which efficiently aggregates local information by leveraging elevation and azimuthal angles between vectors for hierarchical feature extraction and fusion across different scales. Kumar et al., introduced extended graph convolutional networks (EGCNs) [93] for 3D object classification, optimizing graph construction and prediction speed by sampling fewer points without compromising accuracy. Li et al. described a graph attention-based deep neural network [94], employing an encoder-decoder structure with graph attention mechanisms to preserve geometric information during feature extraction and integration. Huang et al. proposed the dual-graph attention convolution network (DGACN) [95], integrating two types of graph attention convolutions to concurrently learn intrinsic and extrinsic features of point clouds. Att-AdaptNet [96] combines attention-based global feature masking with adaptive graph convolution to effectively process irregular and sparse point clouds by employing a dual-branch architecture to dynamically learn and emphasize critical local geometric features and their interrelations.

AGNet [97] employs attention pooling to enhance local feature extraction. Zhang et al. combined PointNet with GCN in a dual-branch graph-based parallel branch network (Graph-PBN) [98], enhancing feature aggregation with the novel EAGConv operator. Lu and colleagues developed the 3D convolution-transformer network (3DCTN) [99], merging local feature aggregation via graph convolutions with global learning through transformers for efficient and comprehensive feature extraction. Liu et al. introduced the sparse graph convolutional neural network (SGCNN) [100] to address efficiency limitations of traditional methods in 3D point cloud classification, featuring sparse graph convolution (SGC) and sparse feature encoding (SFE) modules to reduce computational complexity and enrich representations by focusing on sparse neighbor information. Lin and Feragen developed differential graph convolution (DiffConv) [101], which adapts to local density variations using density dilation neighborhoods and learned masking attention mechanisms, effective in noisy environments for hierarchical learning in advanced 3D shape classification and scene understanding. 3D graph convolutional network (3D-GCN) [102] utilizes 3D convolution kernels and a graph max-pooling mechanism to process data across scales, ensuring translation and scale invariance. Tamajo introduced a CNN-style graph convolution unit, the shrinking unit [103], which employs autocorrelation

and locality-based pooling to compress point cloud inputs and enable effective feature learning through stacking. Ref. [104] integrated complex structural features and multi-hop information into a single convolution step, improving accuracy but raising computational and scalability challenges.

Wei et al. proposed the adaptive graph convolution (AGConv) [105] on 3D point clouds, dynamically generating adaptive convolution kernels to capture diverse relationships between different semantic parts and enhance geometric information accuracy. Khodadad et al., introduced the multilevel graph convolutional neural network (MLGCN) [106], leveraging precomputed KNN graphs for ultra-efficient 3D point cloud analysis within each GNN block. Qin et al. developed the nonlocal graph attention network (NLGAT) [107] for robust 3D shape classification, employing two subnetworks to generate novel global descriptors, the global relationship network (GRN) and the global structure network (GSN), focusing on capturing global and local features, respectively, and improving classification through Gram matrices to maintain rotational invariance. The partial accuracies of the essential methods are presented in Tables 3 and 4.

### 3.1.3. Discussion

In GCNs, both spectral and spatial approaches represent two predominant frameworks, each adapted to distinct applications, demonstrating unique strengths and limitations. The spectral methods are deeply rooted in graph spectral theory, utilizing the eigendecomposition of the graph Laplacian to define convolutions. This approach theoretically ensures efficient capture of the global structural properties, particularly suited to regular graph data. However, these methods are computationally intensive, are sensitive to graph structural changes, and have limited scalability, making them less suitable for large-scale graph data processing. In contrast, spatial methods update node features by directly operating between nodes and their neighbors, mirroring traditional CNNs' approach to image processing. These methods, not reliant on spectral analysis, are better suited for dynamic or irregular graph structures, such as point clouds, and offer easier parallel processing and scalability.

In the task of point cloud shape classification, each method leverages its strengths. Spectral-based GCNs utilize global information to process highly abstract shape features but are inefficient for large-scale point cloud data and highly sensitive to minor perturbations. Conversely, spatial-based GCNs capture local shape features effectively through flexible node aggregation strategies, providing greater adaptability but potentially requiring more complex network designs to optimize performance. Compared to other traditional point cloud processing techniques, such as multiview projection and voxelization, spectral and spatial graph convolutional methods enable direct learning on point clouds, reducing information loss during preprocessing and allowing for a more comprehensive understanding of the data's complex geometry and topology. Unlike point-based methods and direct convolution on point clouds, GCNs utilize graph structures to directly exploit relationships between points, excelling in capturing complex topological structures and relationships crucial for classification or detection tasks. However, GCNs typically require a graph structure as input, necessitating the construction of an effective graph structure before application, which can be nonintuitive or challenging in some point cloud data contexts.

The selection of an appropriate graph convolutional network type depends on specific application requirements, data scale, and processing capabilities. When selecting an appropriate GCN methodology for specific point cloud analysis tasks, one should choose between spectral and spatial approaches based on the type of data being processed and the specific application requirements. For instance, spectral-based methods such as RGCNN [46], AGCN [47], and HGNN [48] focus on global information processing and the capture of complex topological structures, making them suitable for scenarios where the graph structure is relatively stable and a high level of abstract feature representation is necessary. If the application demands are centered around capturing local features and rapidly adapting to dynamic or irregular graph structures, spatial methods or those based

on local spectral techniques would be more appropriate. For example, the local spectral graph convolutions proposed by [87] optimize feature learning on KNN graphs by avoiding the precomputation of Laplacian matrices, thereby enhancing the extraction of local features. Additionally, 3SGCN [52] effectively processes multispectral point cloud data by combining spectral and spatial graphs, reducing noise artifacts and making it suitable for remote sensing applications. The innovativeness and scalability of the technology should also be considered. For instance, SyncSpecCNN [53] demonstrates the potential of spectral methods in handling diverse datasets by using an expanded convolutional kernel and an innovative spectral transformation network to effectively share weights and perform multiscale analysis across varying sizes and shapes.

**Table 3.** Comparative analysis of 3D point cloud classification and segmentation method performance across ModelNet40, ShapeNet, ScanNet, S3DIS, and other datasets.

Method	Classification Results			Segmentation Results		
	Datasets	mA	OA	Datasets	Cat mIoU	Ins mIoU
ECC [61]	ModelNet40	-	83.2	-	-	-
SpiderCNN [62]	ModelNet40 SHREC15	90.7 -	92.4 95.8	ShapeNet	85.3	-
G3DNet [63]	ModelNet40 Sydney Urban Objects	- -	91.7 72.7	-	-	-
KCNet [64]	ModelNet40	-	91.0	ShapeNet	84.7	-
GAPNet [65]	ModelNet40	89.7	92.4	ShapeNet	84.7	-
DGCNN [66]	ModelNet40	90.7	93.5	ShapeNet S3DIS	85.2 56.1	- 84.1
LDGCNN [67]	ModelNet40	90.3	92.9	ShapeNet	85.1	-
DGCNN with AFF [68]	ModelNet40	90.6	93.6	ShapeNetPart	85.6	-
Geometric attentional DGCNN [69]	ModelNet40	91.5	94.0	ShapeNet	84.6	86.3
DPAM [72]	ModelNet40	-	91.9	ShapeNetPart	86.1	-
SRINet [73]	ModelNet40	-	87.01	ShapeNetPart	89.24	-
DCG-Net [74]	ModelNet40	-	93.4	ShapeNetPart	82.3	85.4
RI-GCN [75]	ModelNet40	-	91.0	-	-	-
GGM-Net [77]	ModelNet40	89.0	92.6	-	-	-
CPL [78]	ModelNet40	90.53	92.41	-	-	-
MSDynamic GCN [79]	ModelNet40	-	91.79	ShapeNetPart	85.47	-
Spherical kernel [80]	ModelNet40	-	92.1	ShapeNetPart RueMonge2014 ScanNet S3DIS(Area5)	84.9 66.3 61.0 68.9	86.8 - - 88.6
3D-GCN [81]	ModelNet40	-	92.1	ShapeNetPart	82.1	85.1
MRFGAT [82]	ModelNet40	90.1	92.5	-	-	-
Manifold-Net [83]	ModelNet40	90.1	93.0	S3DIS	72.6	-
LKPO-GNN [85]	ModelNet40	-	91.4	ShapeNetPart ScanNet S3DIS	85.6 58.4 64.6	- - 85.8
HDGN [86]	ModelNet40	-	93.9	ShapeNet	85.4	-

**Table 4.** Comparative analysis of 3D point cloud classification and segmentation method performance across ModelNet40, ShapeNet, S3DIS, and other datasets.

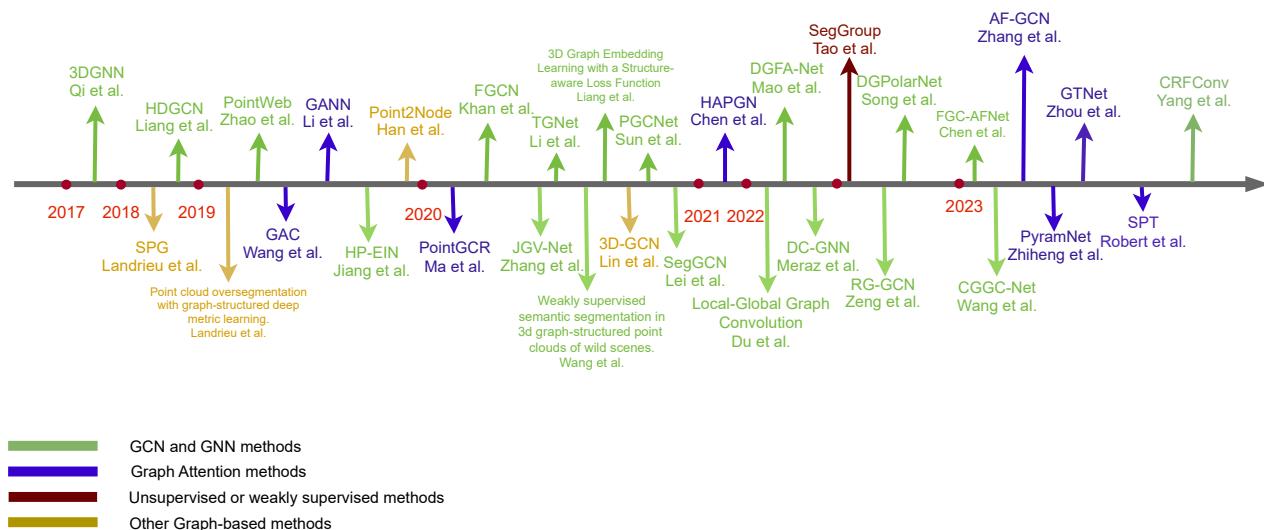
Method	Datasets	Classification Results		Datasets	Segmentation Results		
		mA	OA		Cat mIoU	Ins mIoU	OA
DNRGC [87]	ModelNet40	87.12	89.91	S3DIS	41.9	-	-
PointVGG [88]	ModelNet40	-	93.6	ShapeNet	-	86.1	-
LGFGC [89]	ModelNet40	94.1	96.9	ShapeNet	72.4	-	-
				ScanNetV2	72.4	-	-
				S3DIS(Area5)	69.4	-	89.4
				PL3D	78.5	-	98.0
Grid-GCN [91]	ModelNet40	91.3	93.1	ScanNet	-	-	85.4
				S3DIS	57.75	-	86.94
VA-GCN [92]	ModelNet40	91.4	94.3	S3DIS(Area5)	56.9	-	-
				ShapeNet	82.6	85.5	-
EGCN [93]	ModelNet40	-	90.59	-	-	-	-
	Oakland	-	89.71	-	-	-	-
GADNN [94]	ModelNet40	-	92.9	ShapeNet	85.8	-	-
				S3DIS	87.5	-	-
DGACN [95]	ModelNet40	91.2	94.1	-	-	-	-
	ScanObjectNN	77.9	82.1	-	-	-	-
Att-AdaptNet [96]	ModelNet40	-	93.8	-	-	-	-
AGNet [97]	ModelNet40	90.9	93.6	ShapeNetPart	85.4	-	-
				S3DIS	59.6	-	85.9
Graph-PBN [98]	ModelNet40	90.8	93.4	ShapeNet	85.5	-	-
				S3DIS	59.2	-	84.9
3DCTN [99]	ModelNet40	91.2	93.3	-	-	-	-
SGCNN [100]	ModelNet40	90.4	93.4	-	-	-	-
	ScanObjectNN	-	86.5	-	-	-	-
diffConv [101]	ModelNet40	90.6	93.6	Toronto3D	76.73	-	-
				ShapeNetPart	85.7	-	-
3D-GCN [102]	ModelNet40	-	92.1	ShapeNetPart	82.7	85.6	-
				S3DIS(Area5)	51.9	-	84.6
Shrinking unit [103]	ModelNet10	-	90.6	-	-	-	-
SAMHGC [104]	ModelNet40	91.4	93.6	-	-	-	-
AGConv [105]	ModelNet40	90.7	93.4	ShapeNetPart	83.4	86.4	-
				S3DIS	67.9	-	90.0
				Paris-Lille 3D	76.9	-	-
MLGCN [106]	ModelNet40	-	90.7	ShapeNetPart	83.2	84.6	-
NLGAT [107]	ModelNet40	-	94.0	-	-	-	-

### 3.2. Segmentation

#### 3.2.1. Semantic Segmentation and Instance Segmentation

In 3D point cloud segmentation tasks, GNNs demonstrate their strengths by comprehensively understanding the global geometric structure and fine-grained details of each point. GNNs effectively capture and propagate spatial relationships and feature information in point clouds through constructing inter-point relational graphs, enabling accurate differentiation between various semantic entities. In semantic segmentation tasks, GNNs leverage graph structure to aggregate features at both global and local scales, enhancing the recognition of each point's semantic role and combining local features with neighborhood structural information to improve handling of sparse or irregular point cloud data. A chronological overview of methods for segmentation tasks over recent years is shown in Figure 5. Qi et al., introduced a novel RGBD semantic segmentation approach [108] by constructing a 3DGNN, which converts 2D pixel points into 3D points and constructs a KNN graph, dynamically updating each node's hidden state via iterative functions to

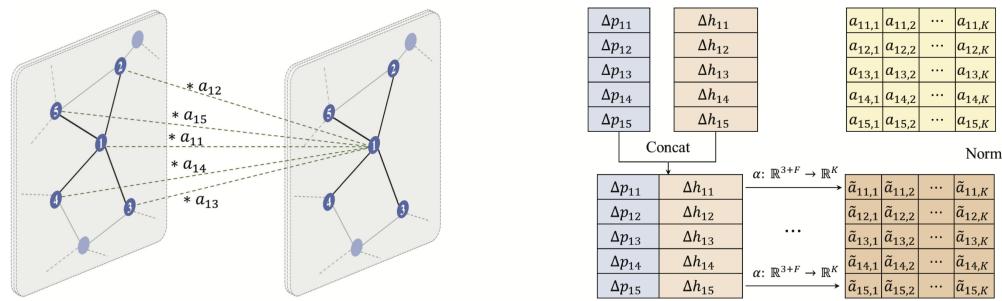
effectively capture long-range dependencies by combining 2D CNN-extracted appearance features with 3D geometric context.



**Figure 5.** Chronological overview of the most relevant segmentation methods.

Landrieu and Simonovsky proposed a deep learning method for large-scale point cloud semantic segmentation with superpoint graphs (SPG) [109], which segments scanned scenes into geometrically uniform elements, capturing the organizational structure of 3D point clouds and providing a compact and information-rich representation for contextual relationships between object parts. They also introduced a supervised framework for oversegmenting point clouds into pure superpoints, formalized as a deep metric learning problem using adjacency graphs, and proposed a graph structural contrastive loss to aid in identifying boundaries between objects. This framework significantly enhances the recognition of object parts by considering entire object parts instead of individual points or voxels, providing detailed descriptions of the relationships between adjacent objects and aiding in contextual classification. Te and colleagues proposed a regularized graph CNN (RGCNN) [46] for 3D point cloud segmentation, which directly processes point clouds by designing a three-layer GCNN that utilizes Chebyshev polynomials to define convolutions on the graph, thus accommodating dynamic graph structures. By incorporating a graph signal smoothing prior into the loss function, RGCNN regularizes the learning process, enhancing the model's robustness to noise and variations in point cloud density.

Liang et al. introduced a hierarchical depthwise graph convolutional neural network (HDGCN) [110] for 3D semantic segmentation, featuring a novel DGConv unit that combines pointwise and depthwise graph convolutions to aggregate and extract both local and global features. Landrieu and Boussaha [111] presented a deep metric learning framework for oversegmenting point clouds into superpoints, employing a graph-structured contrastive loss to produce high-contrast embeddings at object boundaries. Wang et al. proposed the graph attention convolution (GAC) [112], modifying convolution kernels based on object structure and applying attention weights to enhance segmentation accuracy, as shown in Figure 6. Similarly, Li et al. developed a graph attention neural network (GANN) [113] that refines point cloud recognition by weighting the relationships between points using an attention mechanism. PointWeb's [114] adaptive feature adjustment (AFA) module intensively connects points within local neighborhoods to optimize feature aggregation.



**Figure 6.** Illustration of GAC (left), the output is a weighted combination of the neighbors of point 1. Furthermore, The attention mechanism employed in GAC for dynamically attentional weights generating. It receives the neighboring vertices' spatial positions and features as input, and then maps them to normalized attentional weights(right).

Jiang et al. introduced a hierarchical point-edge interaction network (HPEIN) [115] that dynamically constructs edges between points to label 3D scenes, utilizing a dual-branch encoder-decoder architecture to integrate features at multiple levels and employing edge upsampling to connect features across different levels. The AdaAgg scheme, inspired by graph convolutions, encodes edge features to create linkage weights, updating point features as weighted sums of adjacent points and showing modest gains over traditional pooling methods. Point2Node [116] enhances spatial feature descriptions through adaptive aggregation, dynamically exploring and aggregating node correlations across levels using a dynamic node correlation (DNC) module. Lei et al. proposed SegGCN [117], a graph convolutional network based on a fuzzy spherical kernel that applies depth-separable convolutions for efficient semantic segmentation. Song et al. introduced DGPolNet [118] for LiDAR point cloud segmentation, transforming data into polar coordinates and employing a dynamic graph convolution network to address point cloud sparsity. Zhang et al., utilized a U-Net-based sparse convolutional neural network (SU-Net) [119] for ALS point cloud data, enhancing spatial contextual feature extraction and combining it with a conditional random field (CRF) model for improved ground object recognition.

Ma et al., introduced the point global context reasoning (PointGCR) [120] module, which acquires global contextual information in 3D point cloud features through a ChannelGraph. This lightweight plug-and-play unit effectively learns channel dependencies, enhancing segmentation networks with advanced attention mechanisms and dimensional transformations. Shi et al. [121] introduced a deep feature-based graph convolutional network for urban 3D point clouds, which integrates 2D CNN and GCN layers to enhance both stability and segmentation accuracy by combining local and global features. Li et al. developed TGNet [122], employing a Taylor Gaussian mixture model (GMM) to refine local geometric feature representations in point clouds. TGNet employs TGConv units and a parameterized pooling layer to extract and aggregate multiscale semantic features, also incorporating a conditional random field (CRF) to improve segmentation results. Wang et al. [123] proposed a large-scale segmentation framework, trained using two-dimensional supervision, which utilizes a graph-based pyramid feature network (GPFN) and an observability network (OBSNet) to manage occlusion and enhance the correlation between 2D and 3D data, showing competitive performance across synthetic and real-world datasets. Liang et al. [124] introduced a submanifold sparse convolutional network for generating semantic predictions and instance embeddings, incorporating an attention-based KNN method and a subsequent graph convolutional network to refine the embeddings with a structure-aware loss function, enhancing the distinctiveness of each 3D instance embedding.

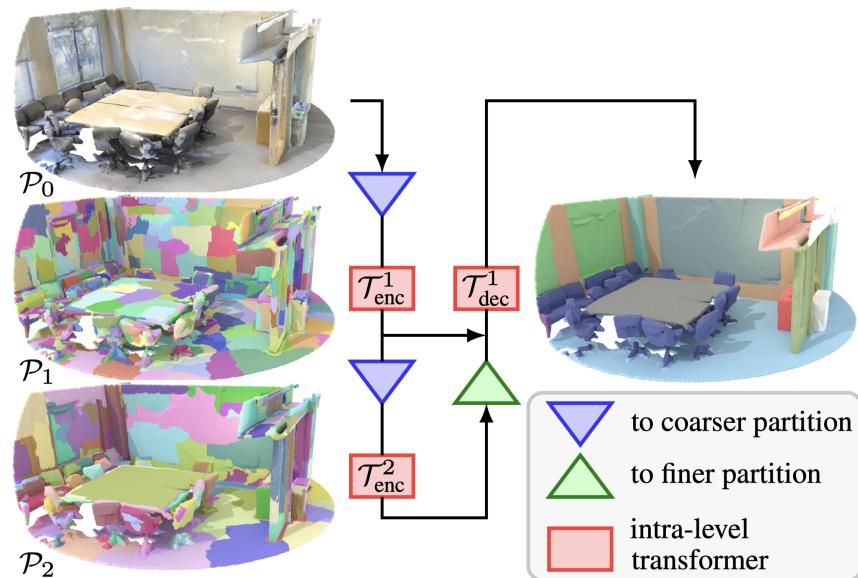
Sun et al., introduced PGCNet [125], a novel indoor scene point cloud semantic segmentation framework that effectively addresses the challenges of large-scale point cloud data by adopting surface patches as the data representation. This method extracts surface patches from the input point cloud, employing a region-growing strategy, computes feature

descriptors for each patch, and constructs a scene patch graph (SPG) where surface patches serve as graph nodes. The PGCNet framework processes the SPG through a dynamic graph U-Net (DGU) module, which includes dynamic edge convolution operations capable of updating the graph structure dynamically within a U-shaped encoder-decoder architecture and encodes hierarchical edge features. Utilizing PGCNet, the input scene can be segmented into room layouts and indoor objects, facilitating rich semantic annotations of various indoor scenes. Chen et al., developed a hierarchical attentive pooling graph network (HAPGN) [126], which employs a gated graph attention network (GGAN) and a hierarchical graph pooling module (HiGPool) to effectively capture the local and hierarchical structure of point clouds, addressing the issue of sample imbalance with a variant of the focal loss function. Du et al., developed the local-global graph convolutional method (LGGCM) [127] for point cloud semantic segmentation, which uses local spatial attention convolution (LSA-Conv) to enhance feature representation and robustness through adaptive neighborhood graph processing and a global attention mechanism. Mao et al. introduced DGFA-Net [128], a graph convolutional approach for point cloud semantic segmentation, which employs dilated graph convolution and a pyramid decoder to enhance multiscale feature representation and diversity in receptive fields. Meraz et al., introduced the drop channel graph neural network (DC-GNN) [129], which uses a KNN-based drop channel technique and hierarchical feature selection to dynamically construct graphs for object classification and part segmentation in point clouds.

Tao et al., introduced SegGroup [130], a weakly supervised method for 3D instance and semantic segmentation that significantly reduces annotation costs by employing single-point instance location as segment-level supervision. This method first oversegments the point cloud and then uses a GCN to propagate label information across segments, thereby generating point-level pseudo labels. This approach achieves performance comparable to fully supervised methods, validating the efficacy of low-cost annotations for 3D segmentation. Zeng et al. proposed a random graph convolutional network (RG-GCN) [131], which incorporates a random graph module to enhance data diversity, addressing the scarcity of training samples in supervised learning for point cloud segmentation. This network also features a module that effectively captures local features. Xu et al. developed the learnable-graph convolutional neural network (LG-CNN) [131], which includes an adaptive neighborhood search to update central point information and an adaptive graph convolution (IT-Conv) for managing diverse point relationships efficiently.

Chen et al. introduced FGC-AFNet [132], which incorporates a feature graph convolution (FGC) module and dual attentive fusion (AF) mechanisms. This network captures local geometrical features efficiently and fuses semantic features across multiple levels, employing an adaptive weighted cross-entropy loss to address data imbalance. CGGC-Net [133] combines edge convolution with attention mechanisms to enhance the representation of geometric structures and semantic relations, optimized through a multitask strategy that includes a category-aware contrastive loss. Zhang et al. proposed AF-GCN [134], which merges graph convolution with self-attention to improve long-range context modeling and spatial feature projection, thus enhancing graph representation. PyramNet [135] integrates a graph embedding module (GEM) with a pyramid attention network (PAN), maintaining fine geometric details and robust expression of local features. Zhou et al. developed GTNet [136], featuring a graph transformer that comprises local and global transformer modules. The local transformer calculates weights dynamically for neighboring points, enabling intra-domain cross-attention with updated graph relations, while the global transformer uses global self-attention to expand the receptive field and capture coarse-grained features. GTNet effectively learns both local fine-grained and global coarse-grained features from point clouds, overcoming limitations of previous methods. Yang et al. [137] introduced reconceptualizing conditional random fields (CRFs) within the feature space as opposed to the conventional label space. This reconceptualization captures feature structural properties via a continuous quadratic energy model solved by message-passing graph convolution within a deep network. the proposed CRFConv, em-

bedded in an encoder–decoder network, enhances feature localization by restoring details lost during encoding. Robert et al. introduced SPT [138], which enhances the efficiency of large-scale 3D point cloud semantic segmentation by leveraging a hierarchical superpoint structure and a transformer-based model, based on the construction of superpoints proposed in [109]. SPT achieves faster preprocessing and reduced computational requirements. Key innovations include a new algorithm for partitioning point clouds into hierarchical superpoints, as shown in Figure 7, using handcrafted node features and adjacency features and constructing a superpoint graph connecting adjacent components, as well as a sparse self-attention mechanism to capture relationships between superpoints at multiple scales, then classifying the finest-scale superpoints. The partial accuracies of the essential methods are presented in Tables 5 and 6.



**Figure 7.** Superpoint transformer. The proposed architecture is represented with two partition levels, P1 and P2. A transformer-based module is used to leverage the context at different scales, leading to large receptive fields.

### 3.2.2. Discussion

GNNs have demonstrated significant advantages in handling complex geometric structures inherent in 3D point cloud segmentation tasks, primarily through their capacity to model the intricate inter-point relationships within graph-based data structures. GNNs are adept at capturing both global and local structural information, crucial for accurately segmenting and classifying various semantic components of point clouds. They dynamically update graph structures and employ advanced techniques such as attention mechanisms and hierarchical pooling to adaptively learn relevant features. For instance, the use of hierarchical attention in models like HAPGN [126] allows for a dynamic focus on relevant local features, enhancing segmentation accuracy in complex scenes. Moreover, GNNs are particularly robust in dealing with sparse and irregular data, employing methods like depthwise convolutions and adaptive feature aggregation to maintain high performance. For example, PointWeb [114] optimizes the feature aggregation process by densely connecting points within local neighborhoods through its adaptive feature adjustment (AFA) module.

However, GNNs face challenges that could hinder their broader adoption. The computational complexity associated with graph construction and updates can be substantial, potentially limiting use in real-time applications or scenarios involving large-scale data without significant computational resources. The effectiveness of GNNs is also highly dependent on the quality of the underlying graph structure, with poor construction potentially leading to suboptimal outcomes. For example, if the construction of the graph

is not precise enough, models like TGNet [122], which utilizes a Taylor Gaussian mixture model (GMM), may not effectively refine the representation of local geometric features. Scalability remains a concern, as the quadratic increase in connections with additional points challenges the application of GNNs to very large datasets.

When compared to traditional methods such as voxel-based or multiview CNNs, GNNs offer reduced information loss during preprocessing and can exploit the complex topologies of point clouds more directly. However, methods like octrees or spatial hashing may provide faster processing speeds for certain applications, though possibly at the expense of higher information loss or reduced detail accuracy.

For selecting an appropriate GNN model for specific applications, it is crucial to consider the task's specific requirements. High precision in geometric detail might necessitate models with high-resolution graph structures or those capable of multiscale feature extraction. For instance, DGFA-Net [128] enhances multiscale feature representation and diversity of receptive fields through the use of dilated graph convolution and a pyramid decoder. Applications demanding rapid processing may benefit from simpler GNN architectures or hybrid models combining graph-based processing with traditional methods. For dynamically changing environments, models with adaptive graph convolution capabilities are preferable due to their ability to update graph structures in response to data changes. Thus, while GNNs present a powerful tool for 3D point cloud segmentation, the choice of a specific model should be carefully guided by both the inherent capabilities of the GNN architecture and the specific needs of the application to ensure the effectiveness of GNN deployment in real-world scenarios.

**Table 5.** Comparative analysis of 3D point cloud classification and segmentation method performance across ModelNet40, ShapeNet, S3DIS, and other datasets (part 2).

Method	Datasets	Classification Results		Datasets	Segmentation Results		
		mA	OA		Cat mIoU	Ins mIoU	OA
SPG [109]	-	-	-	Semantic3D	76.2	-	92.9
	-	-	-	S3DIS	62.1	-	85.5
HDGCN [110]	-	-	-	S3DIS(Area5)	59.33	-	-
	-	-	-	Paris-Lille 3D	68.30	-	-
GSDML [111]	-	-	-	S3DIS(6-fold)	68.4	-	87.9
	-	-	-	vKITTI	52.0	-	84.3
GAC [112]	-	-	-	S3DIS(Area5)	62.85	-	87.79
	-	-	-	Semantic3D	70.8	-	91.9
GANN [113]	ModelNet40	-	91.4	S3DIS(Area5)	57.42	-	85.31
				ShapeNetPart	86.3	-	-
PointWeb [114]	ModelNet40	89.4	92.3	S3DIS(Area5)	60.28	-	86.97
Point2Node [116]	ModelNet40	-	93.0	S3DIS(Area5)	62.96	-	88.81
				ScanNet	-	-	86.3
SegGCN [117]	-	-	-	S3DIS(Area5)	63.6	-	88.2
	-	-	-	ScanNet	58.9	-	-
JGV-Net [119]	-	-	-	ISPRS	85.0	-	-
	-	-	-	DFC2019	0.990	-	-
PointConv-GCR [120]	-	-	-	ScanNet	60.8	-	-
	-	-	-	S3DIS	52.42	-	-
	-	-	-	Semantic3D	69.5	-	92.1
FGCN [121]	-	-	-	S3DIS	52.17	-	-
	-	-	-	Semantic3D	62.40	-	-
	-	-	-	ShapeNetPart	-	-	83.1

**Table 6.** Comparative analysis of 3D point cloud classification and segmentation method performance across ModelNet40, ShapeNet, S3DIS, and other datasets (part 1).

Method	Datasets	Classification Results		Datasets	Segmentation Results		
		mA	OA		Cat mIoU	Ins mIoU	OA
TGNet [122]	-	-	-	ScanNet	62.2	-	-
	-	-	-	S3DIS(Area5)	57.8	-	88.5
	-	-	-	Paris-Lille-3D	68.17	-	96.97
3DGELS [124]	-	-	-	ScanNet v2	45.9	-	-
	-	-	-	NYUv2	43.0	-	-
PGCNet [125]	-	-	-	S3DIS(Area5)	53.60	-	86.24
	-	-	-	ScanNet	83.9	-	-
HAPGN [126]	ModelNet40	89.4	91.7	ShapeNet	89.3	-	-
				S3DIS	-	-	85.8
LGGCM [127]	-	-	-	S3DIS	63.28	-	88.77
	-	-	-	ScanNetV1	42.2	-	87.3
	-	-	-	ScanNetV2	64.4	-	88.6
	-	-	-	ShapeNetPart	86.6	-	-
DGFA-Net [128]	-	-	-	S3DIS(Area5)	65.8	-	88.2
	-	-	-	ShapeNetPart	83.8	85.5	-
	-	-	-	Toronto3D	64.25	-	94.78
DC-GNN [129]	ModelNet40	-	93.64	ShapeNetPart	84.55	-	-
SegGroup [130]	-	-	-	ScanNet	62.7	-	-
RG-GCN [131]	-	-	-	S3DIS(6-fold)	63.7	-	88.1
	-	-	-	Toronto3D	74.5	-	96.5
FGC-AFNet [132]	-	-	-	S3DIS	71.2	-	88.6
	-	-	-	Toronto3D	81.92	-	96.58
CGGC-Net [133]	-	-	-	SemanticKITTI	58.4	-	-
	-	-	-	S3DIS	70.2	-	88.0
AF-GCN [134]	-	-	-	ShapeNetPart	85.3	87.0	-
	-	-	-	S3DIS	73.3	-	91.5
PyramNet [135]	ModelNet40	88.3	91.5	ShapeNet	83.9	-	-
				S3DIS	55.6	-	85.6
GTNet [136]	ModelNet40	92.6	93.2	ShapeNetPart	85.1	-	-
				S3DIS	64.3	-	86.6
CRFConv [137]	-	-	-	ShapeNet Part	83.5	85.5	-
	-	-	-	S3DIS(Area5)	66.2	-	89.2
	-	-	-	Semantic3D	74.9	-	94.2
SPT [138]	-	-	-	S3DIS(6-Fold)	76.0	-	-
	-	-	-	S3DIS(Area5)	68.9	-	-
	-	-	-	KITTI-360 Val	63.5	-	-
	-	-	-	DALES	79.6	-	-
PAConv [139]	ModelNet40	-	93.9	ShapeNetPart	84.6	86.1	-
				S3DIS	66.58	-	-

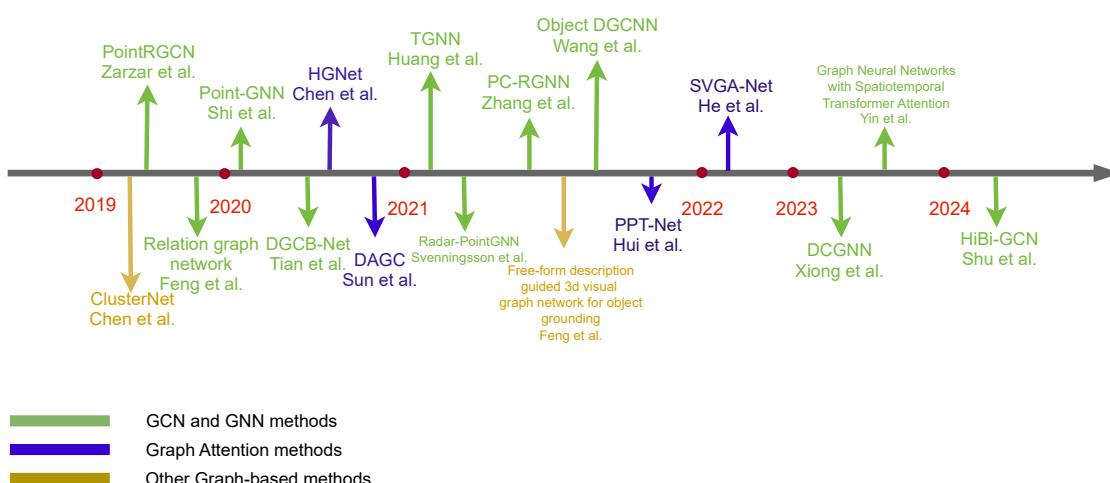
### 3.3. Object Detection

#### 3.3.1. Three-Dimensional Object Detection and Place Recognition

The task of object detection in point clouds involves recognizing and locating objects within 3D spaces. This process primarily requires extracting relevant information from unordered and unevenly dense 3D point cloud data to identify various objects and their boundaries. Traditional methods, which rely on geometric feature extraction and manually designed descriptors, often fall short in complex environments. Deep learning, notably CNNs, has made substantial advances in image recognition recently. However, it is a challenge to adapt these methods to point cloud data due to their irregular structures and sparse information.

Place recognition in point clouds involves recognizing known scenes or locations within 3D spaces, essential for robotic navigation, autonomous vehicles, and augmented

reality applications. This task also demands the extraction of effective information from extensive 3D point cloud data to accurately recognize and differentiate scenes or environments. Traditional methods, which often utilize specific environmental landmarks or manually designed feature descriptors, may lack accuracy and robustness in complex or dynamically changing environments. A chronological overview of methods for object detection and place recognition tasks over recent years is shown in Figure 8. Chen et al. proposed a rotation-invariant point cloud representation that ensures consistent representation regardless of orientation. They also introduced ClusterNet [140], which utilizes hierarchical clustering to explore point cloud geometry and facilitate feature learning. Unlike DGCNN [66], ClusterNet employs EdgeConv layers on aggregated features from clustered point sets, thus enhancing feature learning through dense sampling and dynamic graphs.



**Figure 8.** Chronological overview of 3D object detection and place recognition method.

Zarzar et al. introduced PointRGCN [141], a graph convolutional network designed for 3D vehicle detection, featuring R-GCN and C-GCN modules that aggregate features and refine candidate boxes with contextual data. Feng et al. [142] developed a 3D object detection framework using a relation graph network that improves bounding box accuracy through point attention pooling. This method integrates semantic and directional features to construct object-to-object relation graphs, thereby enhancing the detection of spatial and semantic relationships. Shi et al. introduced Point-GNN [143], a graph neural network for 3D object detection in point clouds. This network encodes clouds into a neighbor graph for classifying object categories and shapes and innovatively includes an automatic registration mechanism to minimize translational variations, complemented by a merging and scoring system to enhance detection accuracy across multiple vertices. Liang et al. [144] developed a deep continuous fusion method for multisensor 3D object detection, employing continuous convolutional layers to integrate image and LiDAR feature maps at various resolutions, thereby enhancing detection accuracy. This method encodes detailed geometric relations across modalities, establishing a reliable multisensor detector. Huang introduced the text-guided graph neural network (TGNN) [145], which integrates textual queries with point cloud features to guide the learning of point offsets toward object centers, enhancing feature fusion and object candidate accuracy through attention-driven graph layers.

Tian et al. developed the dynamic graph convolutional broad network (DGCB-Net) [146], utilizing edge convolutional layers for local feature extraction from point cloud graphs, augmented by a broad network structure for enhanced feature aggregation. Chen et al. introduced the hierarchical graph network (HGNet) [147], employing a shape attention graph convolution (SA-GConv) to delineate object shapes and a U-shaped network for integrating multilevel features in 3D object detection, complemented by a graph convolution-based proposal reasoning module for bounding box prediction. Svennings-

son et al. proposed Radar-PointGNN [148], a novel graph-based approach for radar point clouds, marking a significant evaluation of graph convolution in radar-based multiobject recognition. Zhang et al. addressed the issue of point cloud sparsity with PC-RGNN [149], combining point cloud completion and graph neural networks to enhance feature encoding through a local-global attention mechanism. Feng et al. devised a free-form description guided 3D visual graph network [150], leveraging a language scene graph and a multilevel 3D bounding box relationship graph to improve object grounding in point cloud scenes.

Wang et al. developed object DGCNN [151], an adaptation of the dynamic graph convolutional network for direct 3D object modeling and detection. This model optimizes sparse structures, eliminating the need for nonmaximum suppression and surpassing traditional self-attention mechanisms in performance metrics. He et al. introduced SVGA-Net [152], which constructs local complete graphs within segmented 3D spherical voxels and a global KNN graph across voxels, enhancing feature extraction and bounding box accuracy through a novel sparse-to-dense regression module. DCGNN [153] employs density clustering and graph neural networks for single-stage 3D detection, optimizing point cloud partitioning and enriching feature details through local and global graph-based interactions. Yin et al. [154] proposed a hybrid approach that combines graph neural networks with spatiotemporal transformer attention for 3D video object detection from point clouds. This method distinguishes short-term and long-term temporal patterns using a grid messaging passing network (GMPNet) and an attention spatiotemporal transformer GRU (AST-GRU), respectively, enhancing detection accuracy for small and moving objects.

Shu et al. developed the hierarchical bidirectional graph convolutional network (HiBi-GCN) [155], which constructs hierarchical structures through unsupervised clustering. This network utilizes pooling and merging strategies for graph fusion, enabling the extraction of discriminative global descriptors for place recognition. It incorporates a position-only point cloud framework to optimize 3D bounding box estimation and employs a novel point attention pooling method that enhances proposal features and improves bounding box regression performance. Sun et al. introduced the dual attention graph convolutional approach (DAGC) [156], which isolates task-relevant features using a dual attention module and a residual graph convolution network (ResGCN) to aggregate local multilayer neighborhood features, significantly enhancing feature discrimination for place recognition. Hui et al. proposed the pyramid point cloud transformer network (PPT-Net) [157] for large-scale place recognition, employing hierarchical attention mechanisms and graph convolutions within pyramid point transformer modules to capture spatial relations at varying scales. This approach integrates features through a pyramid VLAD module, culminating in robust global descriptors. The partial accuracies of the essential methods are presented in Tables 7 and 8.

### 3.3.2. Discussion

Point cloud target detection and location recognition tasks play a critical role in the processing of extensive data in three-dimensional spaces, aiming to identify and locate objects or scenes. Traditional methods rely on manually designed feature descriptors and environmental landmark recognition, such as buildings or landmarks used for robot navigation and autonomous driving. However, these approaches often have limited effectiveness in complex and dynamically changing environments, as well as in disordered and unevenly distributed point cloud data.

The introduction of GNNs has brought innovative solutions to this field, effectively adapting to the disorder and irregular structure of the data through the construction of complex graph structures. For instance, ClusterNet [140] and HGNet [147] utilize hierarchical clustering and multilevel feature integration, not only enhancing the capability to extract features from point clouds but also improving the quality of feature representation and the capture of spatial relationships through the graph convolution process, significantly increasing accuracy and robustness in recognition.

Compared to other methods such as VoxelNet [158], PV-RCNN [159], VoteNet [160], and RandLA-Net [161], GNNs demonstrate their unique advantages and limitations. VoxelNet and PV-RCNN enhance the efficiency of data processing and maintain the original geometric information of point clouds through voxelization, suitable for three-dimensional target detection in autonomous driving. VoteNet employs a deep learning-based voting mechanism to predict the center of objects through key points in the cloud, effectively locating objects. RandLA-Net reduces the computational complexity of handling large-scale point clouds through a random point sampling method.

In comparison, GNNs exhibit greater adaptability, especially in tasks involving complex topological structures or dynamically changing point clouds. However, the computational complexity and resource demands of GNNs may not be suitable for applications requiring real-time processing. In contrast, methods like VoxelNet and PV-RCNN maintain good accuracy while ensuring efficiency, suitable for environments with limited computational resources. VoteNet and RandLA-Net also demonstrate unique advantages in specific scenarios, such as VoteNet's efficiency in object localization through its voting mechanism and RandLA-Net's high-speed processing capabilities in handling large-scale data. Selecting the appropriate point cloud processing technology requires consideration of specific application needs, data characteristics, and available computational resources. GNNs offer a powerful framework particularly suited for scenarios requiring advanced feature integration and complex spatial relationship understanding, while other methods may have advantages in terms of real-time performance or resource efficiency.

**Table 7.** Comparative evaluation of 3D object detection methods on the KITTI Dataset by object type and difficulty level. The modalities are LiDAR (L) and image (I). 'E', 'M', and 'H' represent easy, moderate, and hard classes of objects, respectively.

Method	Modality	Cars			Pedestrians			Cyclists		
		E	M	H	E	M	H	E	M	H
PointRGCN [141]	L	85.97	75.73	70.60	-	-	-	-	-	-
Point-GNN [143]	L	88.33	79.47	72.29	51.92	43.77	40.14	78.60	63.48	57.08
ContFuse [144]	L&I	82.54	66.22	64.04	-	-	-	-	-	-
Radar-PointGNN [148]	L	-	-	-	-	-	-	-	-	-
PC-RGNN [149]	L	89.13	79.90	75.54	-	-	-	-	-	-
SVGA-Net [152]	L	87.33	80.47	75.91	48.48	40.39	37.92	78.58	62.28	54.88
DCGNN [153]	L	89.65	79.80	74.52	-	-	-	-	-	-

**Table 8.** Performance metrics of 3D object detection methods across multiple datasets.

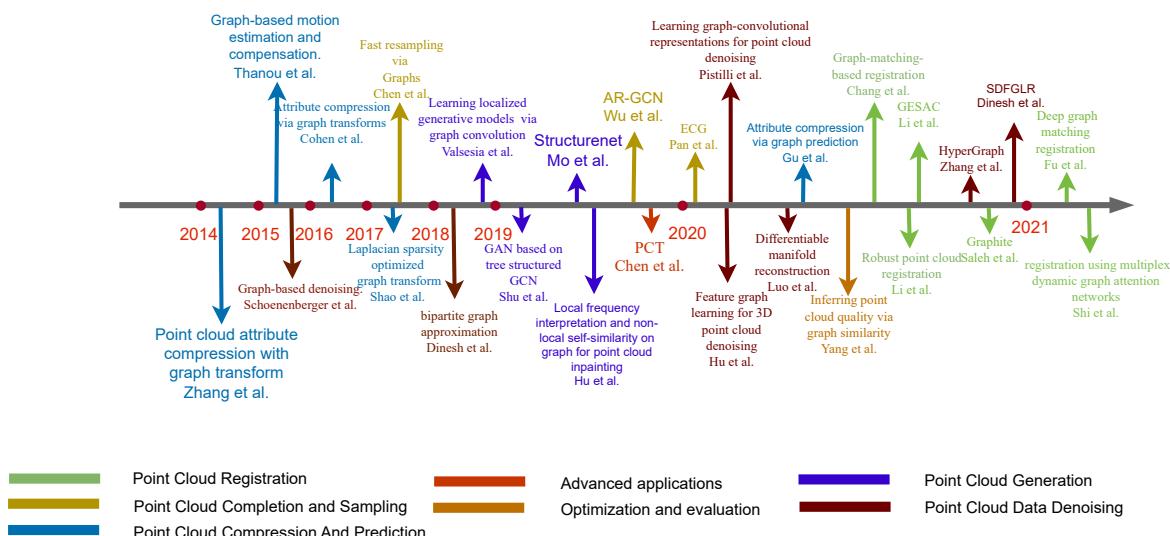
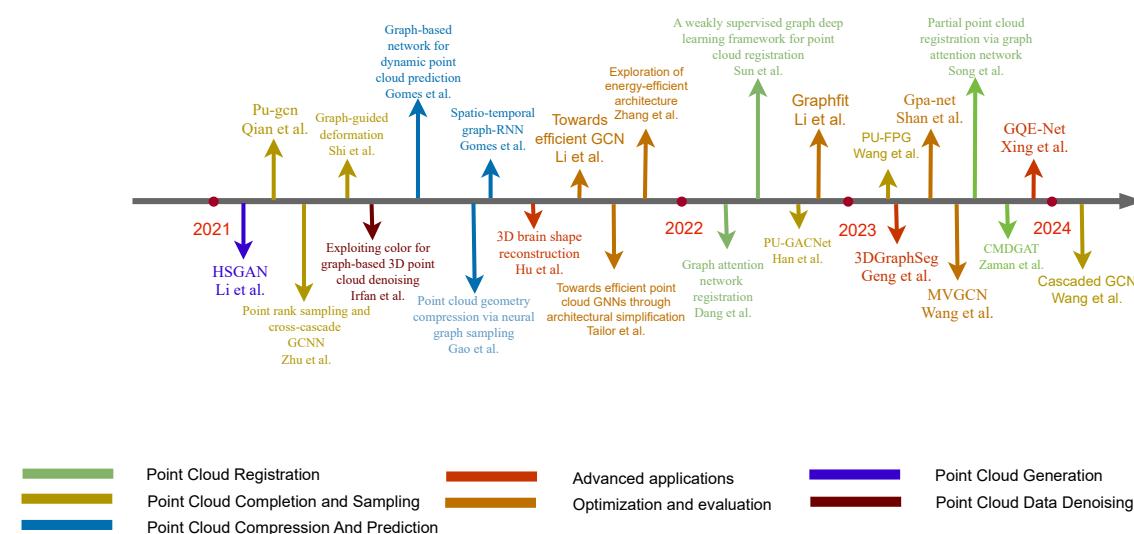
Methods	Datasets						
RGN-3DOD [142]	SunRGB-D	mAP@0.25	59.2				
	ScanNet	mAP@0.25	48.5	mAP@0.5	26.0		
TGNN [145]	ScanRefer	Unique Acc@0.25	68.61	Multiple Acc@0.5	29.84	Overall Acc@0.5	Acc@0.25
	Validation		68.30		23.18	37.37	29.70
	Test			58.90	33.10	25.30	41.00
HGNet [147]	SunRGB-D	mAP@0.25	61.6	cvAP 0.31			
Radar-PointGNN [148]	nuScenes Car	AP 10.1		ATE 0.69	ASE 0.20	AOE 0.38	AVE 0.95

**Table 8.** Cont.

Methods	Datasets	Acc@0.5(Unique)	Acc@0.5(Multiple)	Acc@0.5(Overall)
FDG3D-VGN [150]	ScanRefer	75.40	30.20	43.16
GSTA-3DVOD-PC [154]	nuScenes	NDS 71.8	mAP 67.4	
DAGC [156]	ScanNetV2	mAP@0.25 61.3	mAP@0.50 34.4	cvAP@0.25 0.38
PPT-Net [157]	Oxford	U.S. 98.4	R.A. 99.7	B.D. 95.3

### 3.4. Others

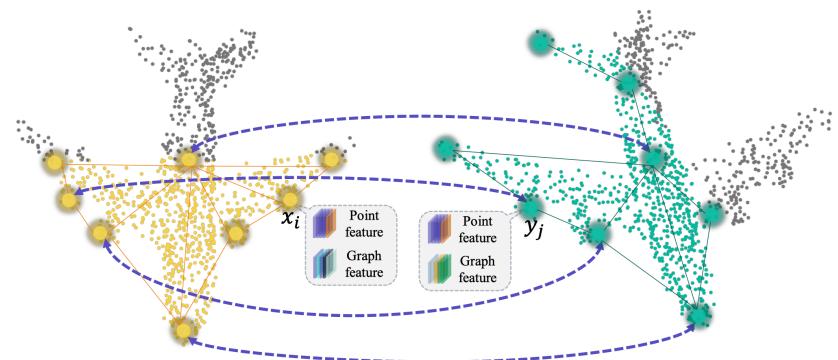
In this section, we summarize the applications of graph-based and GNN methods in the areas of registration, generation, completion and sampling, denoising, compression and prediction, advanced applications, optimization, and evaluation. A chronological overview of methods over recent years is shown in Figures 9 and 10.

**Figure 9.** Chronological overview of other relevant methods from 2014 to 2021.**Figure 10.** Chronological overview of other relevant methods from 2021 to present.

### 3.4.1. Point Cloud Registration

Point cloud registration is crucial for applications such as 3D reconstruction, robotic navigation, and augmented reality, aiming to align multiple point cloud datasets into a unified coordinate system. Traditional methods like the iterative closest point (ICP) and its variants often struggle with large-scale, noisy, or feature-deficient point clouds due to their reliance on minimizing geometric distances.

GNNs address these limitations by processing unstructured point cloud data directly, preserving geometric details more effectively. Fu et al. [162] introduced a robust point cloud registration framework utilizing deep graph matching (RGM), as shown in Figure 11. This framework transforms point clouds into graphs, applies graph convolutions to extract deep features, and employs an AIS module with affinity layers, instance normalization, and the Sinkhorn algorithm to predict and refine correspondences between graph nodes. The method significantly reduces incorrect correspondences and enhances registration accuracy using the Hungarian algorithm and singular value decomposition (SVD) during testing. Chang et al. [163] proposed a nonrigid point cloud registration method that enhances accuracy in nonrigid environments through graph matching. This method segments the point cloud into smaller, more manageable rigid parts, simplifying the correspondence search and effectively handling large motions and multi-to-one matching problems to robustly identify semantically equivalent parts. Li et al. [164] introduced a point cloud registration method utilizing topological graphs combined with the Cauchy weighted  $l_q$ -norm, which is particularly effective in handling outliers and partial overlaps. The method starts with feature point sets represented as topological graphs, simplifying the matching process to an edge matching problem, employing the weighted  $l_q$ -norm for coarse registration, and decomposing the problem into simpler subproblems of rotation and translation estimation. In GESAC [165], graph enhanced sample consensus, Li et al. modified the traditional RANSAC framework to improve outlier management. GESAC employs a two-stage filtering strategy to identify viable subsets for registration, using shape annealing for robust transformation estimation instead of the classic least squares method.



**Figure 11.** The idea of point cloud registration based on graph matching. Dashed lines represent correspondences. Point features and graph features are extracted directly from points and graphs, respectively. The two points  $x_i$  and  $y_j$  have similar point features due to their similar local geometries but have different graph features because the graph topologies around them differ. Thus, they are not mismatched when using graph-based matching.

Graphite framework [166] enhances feature extraction and keypoint detection in point cloud registration through graph representation, significantly improving accuracy and efficiency. The method segments point clouds into patches and employs graph convolutional networks to learn features, integrating a novel loss function based on the Cauchy weighted  $l_q$ -norm to enhance model robustness. Continuing innovations, Shi et al. [167] utilize graph attention networks to enrich feature representation and sustain knowledge continuity through dynamic and continual learning mechanisms. Lai-Dang et al. introduced a framework named DFGAT [168], which combines graph attention networks with

optimal transport algorithms to efficiently extract dense features for point cloud registration through deep learning. Concurrently, Song et al. and Zaman et al. [169,170] developed methods for partial point cloud registration and continual learning in point cloud registration, respectively, both emphasizing the importance of attention mechanisms and knowledge preservation in enhancing registration accuracy. Sun et al. [171] proposed an end-to-end 3D graph deep learning framework that addresses challenges in dense point aggregation and complex spatial structures using a weakly supervised approach. This framework enhances keypoint extraction and feature description by integrating MLP and GCN techniques.

### 3.4.2. Discussion

GNNs have revolutionized the field of point cloud registration by exploiting graph-based structures to tackle the unique challenges associated with processing 3D data. These methods present notable advantages over traditional techniques such as the iterative closest point (ICP) [172] and its variants, especially when dealing with complex, noisy, and feature-sparse environments. The strength of GNNs lies in their ability to process unstructured data while preserving geometric details more effectively. For instance, the robust point cloud registration framework introduced by [162] utilizes deep graph matching to transform point clouds into graph structures, enabling the application of graph convolutions to extract detailed feature representations. This approach not only improves accuracy but also reduces incorrect correspondences through the use of affinity layers, instance normalization, and optimization algorithms such as the Sinkhorn algorithm. Moreover, the adaptability of GNNs is evident in methods proposed by [163], which effectively manage nonrigid environments and challenges such as outliers and partial overlaps. These methods segment the point cloud into more manageable parts or employ novel algorithms like the Cauchy weighted  $l_q$ -norm, demonstrating GNNs' capability to handle significant transformations and complex matching problems.

Comparatively, recent methods, such as PointNetLK [173], deep closest point [174], and 3DRegNet [175], utilize deep learning techniques to directly learn feature representations and geometric transformations from data. These methods are designed to operate end-to-end and often integrate innovations such as self-supervised learning mechanisms, which are highly effective in partial-to-partial registrations and can handle substantial variations in data. While GNNs show considerable promise, they also exhibit limitations that may impact their suitability for specific applications. For instance, the computational complexity and resource demands of GNN-based methods can be substantial, making them less ideal for real-time applications or environments with limited processing capabilities. Additionally, setting up and tuning these networks requires considerable expertise in both graph theory and neural network architectures, which can be a barrier to their widespread adoption.

In terms of applicability to point cloud registration, both GNNs and more recent deep learning-based methods continue to be relevant. Graph-based methods and GNNs are particularly suited for complex scenarios where the relationships and structural integrity of the data are critical. However, for applications where speed and lower computational overhead are paramount, newer deep learning approaches might be more advantageous due to their efficiency and scalability. While graph methods and GNNs remain applicable and highly effective in modern point cloud registration tasks, the choice between these and newer deep learning techniques should be guided by the specific requirements of the application, including the need for accuracy, robustness, computational efficiency, and ease of implementation.

### 3.4.3. Point Cloud Generation

Point cloud generation is crucial in 3D vision, aiming to reconstruct or synthesize detailed 3D shapes from various data sources. Traditional methods such as voxelization and multiview reconstruction have evolved with the integration of advanced techniques

like generative adversarial networks (GANs) [176], which significantly enhance the generation of high-quality point clouds by learning to discriminate between real and synthesized data. However, these methods often face challenges with complex or sparse data, especially in preserving detailed geometric and topological features. GNNs improve point cloud generation by directly processing unstructured data, thus maintaining more original details and enhancing efficiency. GNNs excel in learning complex patterns and generating fine local structures, which are crucial for modeling intricate 3D objects and scenes. Valsesia et al. [177] proposed a GNN model that uses graph convolutions to learn domain features and graph embeddings without predefined graph structures. Their approach also incorporates upsampling within the generator to exploit data self-similarity, thereby improving feature localization and model efficiency. Shu et al. developed tree-GAN [178], utilizing a tree-structured graph convolutional network (TreeGCN) as the generator to leverage ancestral information in graph convolutions, thereby enhancing feature representation. They also introduced the Frechet point cloud distance (FPD) to evaluate the quality of generated 3D point clouds, supporting unsupervised multiclass 3D point cloud generation that allows for the creation of semantically varied point clouds without prior knowledge.

Another approach, StructureNet [179] by Mo et al., encodes 3D shapes into n-ary graphs that represent continuous geometric transformations and structural changes, employing GNNs for order-invariant encoding. This method excels in diverse shape generation and structure-aware operations such as shape interpolation and editing. Similarly, Li et al. [180] utilized a hierarchical self-attention graph learning framework that combines GCNs with self-attention mechanisms to effectively capture both global and local topological features, ensuring detailed and precise 3D shape generation. The integration of a novel loss function with gradient penalty enhances training stability and mitigates mode collapse in GANs [176].

#### 3.4.4. Discussion

Point cloud generation is a critical technique for reconstructing or synthesizing detailed three-dimensional shapes from various data sources. Traditional methods, such as voxelization and multiview reconstruction, often struggle to maintain high-quality geometric and topological details when dealing with complex or sparse datasets. GANs [176] have been extensively employed to enhance the quality of point cloud generation, with PSG-GANs [181] demonstrating strong capabilities in handling complex geometric shapes and TreeGAN [182] and PointFlow [183] showing significant advancements in generating hierarchically rich and continuously reliable point clouds, respectively. Despite the notable progress made by GANs in generation quality, they still face challenges in preserving detailed features in scenarios of data scarcity. GNNs offer a powerful alternative by directly processing unstructured data, which helps better preserve original details and enhance processing efficiency. Ref. [177] leverages graph convolutions to learn neighborhood features without the need for a predefined graph structure, effectively improving feature localization and model efficiency. Further, StructureNet [179] employs GNNs to encode three-dimensional shapes, representing continuous geometric transformations and structural changes, optimizing shape generation and structural perception operations. HSGAN [180] utilizes a hierarchical self-attention and graph convolutional network framework to precisely capture both global and local topological features.

Compared to traditional methods and GANs, GNNs provide improved efficiency and a stronger framework for handling data and capturing complex details, making them particularly suitable for point cloud generation applications that require high fidelity and intricate details. However, the complexity of GNN models and their demand for substantial computational resources may limit their applicability in certain scenarios.

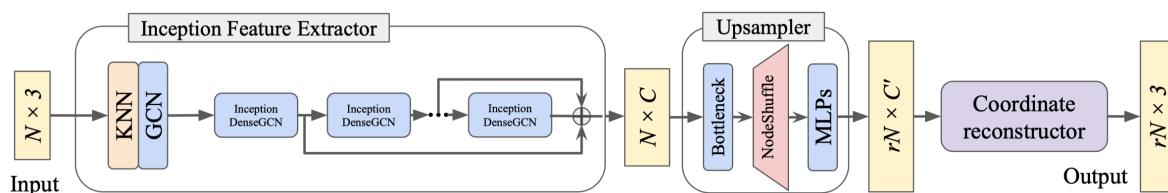
#### 3.4.5. Point Cloud Completion and Sampling

Point cloud completion and upsampling are pivotal tasks in 3D computer vision, aiming to reconstruct high-quality, high-resolution 3D models from incomplete or low-

resolution data. Traditional methods often falter when handling complex geometries, hence the shift towards GNNs. GNNs are adept at reconstructing missing structures by exploiting the global distribution and local nuances of point clouds. In upsampling, GNNs evaluate the local point neighborhoods to insert new points judiciously, thus maintaining the original geometry and aesthetic integrity. Chen et al. [184] introduced a graph signal processing-based resampling strategy for point clouds that preserves structural integrity through graph filtering. Fu et al. [185] tackled the challenges of point cloud inpainting, particularly for clouds marred by acquisition flaws or intricate designs. Their method identifies nonlocal self-similarities and conceptualizes the inpainting task as an optimization problem, incorporating techniques such as voxelization and automatic hole detection. Wu et al. introduced the AR-GCN model [186], which enhances low-resolution point clouds to high-resolution through adversarial residual graph networks, optimizing a GCN with residual links and an unpooling layer to upscale point clouds effectively.

Pan et al. introduced the ECG network [187], enhancing edge features and local details through an edge-aware feature expansion (EFE) module. This network processes data in two phases: skeleton generation and detail refinement. Zhu et al. developed PRSCN [188], utilizing point rank sampling (PRS) and a cross-cascade module (CCM) to integrate features effectively across scales, with a focus on local geometric details. Shi et al. [189] proposed treating point cloud completion as a control-point-guided mesh deformation, refining this process with a graph convolutional network. Wang et al. [190] introduced the cascaded graph convolutional completion network (CGCN), an innovative point cloud completion method that leverages a cascaded encoder-decoder architecture for accurately predicting and reconstructing missing points in partial 3D shapes. Utilizing multilevel feature extraction and a folding-refinement decoder, this method integrates local and global features through graph convolutional operations.

For point cloud upsampling, Qian et al. developed PU-GCN [191], which integrates a NodeShuffle module into the GCN architecture, incorporating a new multiscale point feature extractor, Inception DenseGCN, to enhance performance. The PU-GCN architecture is shown in Figure 12. Han et al. introduced PU-GACNet [192], employing a graph attention convolution (GAC) module that dynamically integrates spatial positions and feature attributes, further supported by an edge-sensitive node shuffle module to preserve local geometric details. PU-FPG [193] introduced a point cloud upsampling model that integrates cascaded networks with graph information to enhance the structural integrity and clarity of upsampled point clouds. Utilizing metrics such as local coordinate difference, local normal difference, and describing index, the model effectively identifies critical form descriptors, addressing common issues like topological inaccuracies and vague contours in upsampled data.



**Figure 12.** PU-GCN architecture.

#### 3.4.6. Discussion

In point cloud completion and upsampling, GNNs exhibit distinct advantages and limitations compared to the aforementioned deep learning approaches. Deep learning methods such as PCN [194], PU-Net [195], TopNet [196], PUGeo-Net [197], and PF-Net [198] effectively handle point cloud data completion and upsampling through specifically designed network architectures, offering innovative solutions to particular problems. For instance, PCN effectively addresses large-scale point cloud omissions through an encoder-decoder architecture; PU-Net employs multilevel feature extraction and expansion strategies to

increase point count while preserving geometric details; and PUGeo-Net maintains post-upsampling geometric accuracy through a geometry-centric network.

In contrast, GNNs emphasize the preservation of structural integrity and precise handling of local details in point cloud completion and upsampling tasks. Capitalizing on the inherent properties of graph structures, GNNs can capture both global and local relationships within point clouds, which is particularly crucial for handling complex geometric structures. This capability makes GNNs particularly effective in reconstructing missing structures and detailed upsampling, as demonstrated in the work of [187,188], in which networks are specifically designed to enhance edge features and local geometric details, making them highly suitable for high-fidelity 3D modeling applications such as virtual reality and precision industrial design. GNNs still face longstanding challenges, especially in terms of computational resources and data dependency. Due to the complexity of graph processing, GNNs typically require more computational resources and extensive training data, which can be a limiting factor in resource-constrained environments. In comparison, deep learning methods like PU-Net often focus on specific types of point cloud processing, such as completing extensive missing areas or maintaining details in upsampling, and may be more efficient for particular tasks. These methods, through end-to-end training, can learn effective feature representations directly from data without the need for complex graph structure design and processing as required by GNNs. Each method has its unique strengths and limitations, and the optimal choice should be based on the specific nature and goals of the task. Zhang et al. [199] introduced a novel hypergraph-based model for enhancing the processing of 3D point clouds. By leveraging tensor-based techniques to estimate hypergraph spectral components and frequency coefficients, the model capably addresses complex relationships within point cloud data, outperforming traditional graph methods particularly in noise-prone environments.

### 3.4.7. Point Cloud Denoising

Recent advancements in 3D point cloud denoising utilize graph-based methods to leverage geometric structures for effective noise reduction. Schoenenberger and colleagues [200] proposed using convex optimization on graph signals to correct positional noise and eliminate outliers. Dinesh et al. [201] introduced a bipartite graph approximation method that applies KL divergence for segmentation and graph total variation regularization via the alternating direction method of multipliers (ADMM) to denoise point clouds. Hu et al. [202] developed a feature graph learning approach, optimizing the Mahalanobis distance to effectively estimate the graph Laplacian matrix, thus enhancing computational efficiency without full eigendecomposition. Pistilli's [203] deep neural network, equipped with graph convolutional layers, dynamically constructs similarity graphs to extract features and improve surface estimation, addressing permutation invariance in point cloud processing. Luo et al. [204] combined multilayer dynamic graph convolutions with differentiable pooling to filter noise while preserving the underlying surface details. Irfan and colleagues [205] focused on exploiting color and geometry correlations using a KNN graph, achieving superior denoising performance. Collectively, these methods mark significant advancements in graph-based point cloud denoising, each uniquely contributing to the field's development. Dinesh et al. [206] introduced a novel 3D point cloud denoising method using a signal-dependent feature graph Laplacian regularizer (SDFGLR) that enhances surface normal smoothness through bipartite graph segmentation, effectively optimizing for Gaussian and Laplacian noise types with demonstrated superiority in extensive testing, though it is computationally intensive. Ref. [203] introduced removing outliers using a unified model that integrates a graph convolutional neural network (GCNN), overcoming the challenges of irregular domain and permutation invariance in point clouds by dynamically constructing neighborhood graphs based on high-dimensional feature similarity.

### 3.4.8. Discussion

In point cloud denoising, traditional methods and graph neural network approaches each have their unique advantages and limitations. For instance, PointProNets [207], by directly processing raw point cloud data without the need for preprocessing or complex data transformations, demonstrate robust capabilities in extracting both global and local features. However, these methods are relatively sensitive to noise, which may necessitate additional denoising steps to optimize performance. In contrast, graph neural network methods, such as convex optimization on graph signals by [200], the bipartite graph approximation and KL divergence method by [201], and the dynamic graph convolutions by [204], effectively leverage the geometric structures of point clouds for precise noise mitigation and terrain preservation. These techniques are particularly suited for handling complex data structures and adapting to varying point cloud configurations. On the other hand, unsupervised learning approaches such as [208] reduce dependence on labeled data, making them applicable in scenarios where data labeling is challenging. When selecting appropriate point cloud denoising technology, one should consider specific application requirements, processing efficiency, denoising quality, and noise type adaptability. Graph neural networks, with their advantages in structured data processing, prove more effective in scenarios requiring high precision and complex data handling, while non-graph-based deep learning methods are better suited for applications demanding real-time processing.

### 3.4.9. Compression and Prediction

Recent advancements in graph-based methods have significantly improved point cloud compression and prediction, enhancing the handling of large-scale data. Zhang et al. [209] developed a method using graph transforms for attribute compression in 3D point clouds, improving storage and transmission efficiency by decorrelating data signals within small neighborhoods. Thanou et al. [210] and Anis et al. [211] explored dynamic 3D point cloud compression through graph-based motion estimation and graph wavelet transforms combined with subdivision meshes, respectively, both leveraging temporal correlations to optimize compression. Addressing sparse point cloud challenges, Cohen et al. proposed using block-based prediction with optimized graph transforms to compress attributes while maintaining spatial accuracy. Shao et al. [212] utilized k-d trees and optimized Laplacian sparsity for better energy compaction in attribute compression. For predictive applications, Gu et al. [213] introduced a scheme for color attribute compression via graph prediction, reducing redundancy by predicting point colors before applying graph transform for residual encoding. Gomes et al. [214] employed a graph-based network to predict future states of dynamic point cloud sequences by integrating topological information for enhanced compression and developed a spatio-temporal Graph-RNN model [215] that effectively simulates long-term trajectories by learning both topological and geometric features, demonstrating predictive accuracy superior to conventional models. Gao et al. [216] used neural graph sampling within a variational autoencoder framework, leveraging multiscale sampling to efficiently compress point cloud geometry, as demonstrated on the ShapeNetCorev2 dataset.

### 3.4.10. Discussion

When selecting technology for point cloud compression and prediction, it is essential to choose the method that best suits the characteristics of the data and the specific application requirements. For processing large-scale static point cloud data, deep learning methods such as [217,218] may be more appropriate as they utilize deep autoencoder networks, variational autoencoders (VAEs), and voxel to provide automated and efficient compression. On the other hand, for point clouds that exhibit dynamic changes or have complex topological structures, graph neural network approaches like those proposed by [214], which include graph-based networks and spatio-temporal Graph-RNN models, offer higher compression efficiency and predictive accuracy, particularly in scenarios that require leveraging temporal correlations and topological information.

### 3.4.11. Advanced Applications

Chen et al. [219] introduced a GNN-based system for representing large-scale 3D point clouds, particularly for autonomous driving applications. The system combines voxelization with graph inception networks to effectively handle large-scale scenes while minimizing discretization errors. Geng et al. [220] proposed a point cloud semantic segmentation framework for high-speed railway environments, utilizing the local embedding superpoint graph (LE-SPG) method to compress data into a concise graph representation. This maintains the data's topological structure and uses a gated integrated graph convolutional network (GIGCN) with a gated hidden state integration layer to prevent gradient problems and maximize feature utilization across layers. Hu et al. [221] developed a point cloud generative model using tree-structured graph convolutions for 3D brain shape reconstruction. This model combines medical imaging and 3D shape representation to improve accuracy in brain surgery, generating 3D point clouds from 2D images using a network of graph convolutional and branching blocks, and includes an edge-aware feature expansion module for preserving geometric details during upsampling. GQE-Net [222] improves compressed point cloud color quality using a parallel–serial graph attention module and a feature refinement module leveraging geometric data. Demonstrating significant enhancements in PSNR (peak signal to noise ratio) and BD-rate (Bjøntegaard delta), GQE-Net effectively restores color attributes. Feng et al. [223] introduced semisupervised techniques, including a graph-widening module and enhanced GCNs for efficient processing of complex point clouds from mobile laser scanning (MLS) systems, reducing the reliance on extensive annotated data.

### 3.4.12. Optimization and Evaluation

GCN studies for point cloud processing have focused on enhancing computational efficiency and accuracy. Li et al. [224] streamlined the basic graph convolution workflow involving KNN searches and MLP, reducing computational complexity and memory usage. Tailor and colleagues [225] improved GNN efficiency by maintaining the initial feature extraction layer and simplifying subsequent layers, thus preserving performance with lower computational demands. Zhang et al. [226] enhanced energy efficiency in graph-based deep learning architectures by optimizing EdgeConv operations and introducing a query-and-exchange-based distributed model to improve SIMD performance. In terms of specialized applications, Li's team created GraphFit [227] for point cloud normal estimation, integrating an adaptive module with an attention mechanism for varying point densities. Yang et al. developed GraphSIM [228] for quality assessment using graph similarity techniques sensitive to the human visual system. GPA-Net [229] excels in no-reference quality assessment, surpassing existing methods in stability and accuracy. Wang et al. introduced MVGCN [230], a multiview framework for identifying surface defects in complex structures, such as aircraft fuselages and concrete samples, by distinguishing between defect and nondefect areas.

## 4. Trends and Challenges

### 4.1. Observed Trends

In the rapidly evolving field of point cloud analysis using graph methods, several key trends have been observed:

- Hybrid graph models: The integration of both spectral and spatial graph approaches, exemplified by models such as 3SGCN [52], indicates a trend towards leveraging the strengths of each method to enhance classification accuracy and mitigate noise. This hybrid strategy provides robust handling of the complex data structures inherent in point clouds.
- Dynamic and adaptive graphs: Techniques like RGCNN [46] and AGCN [47], which dynamically update the graph's structure or adapt the graph topology based on the data, represent a shift towards more adaptable models that can respond to the unique characteristics of each dataset.

- Deep learning integration: The use of deep learning techniques, including autoencoders and advanced convolutional structures in models like DGCNN [151], underscores a growing convergence between deep learning and graph-based methods. This integration enables more effective extraction of complex, high-level features.
- Attention mechanisms and contextual understanding: The adoption of attention mechanisms, as seen in GAPNet [65] and HAPGN [126], is becoming widespread, underscoring an increasing focus on improving the precision of feature weighting and enhancing the contextual understanding of point cloud data. These mechanisms allow models to focus on the most pertinent parts of the data, enhancing learning efficiency and accuracy.
- Unsupervised and semisupervised learning: There is an increasing focus on unsupervised and semisupervised methods, as demonstrated by the framework developed by Shu et al. [54,70,178], reflecting the necessity of managing unlabeled data in practical applications.

#### 4.2. Challenges and Future Trends

Despite significant advancements, several challenges persist that will likely shape future trends in graph-based point cloud analysis:

- Scalability: With the increase in size and complexity of point cloud datasets, scalability is a paramount challenge. Existing methods, particularly those involving dynamic graph updates or spectral transformations, are computationally intensive.
- Robustness to variations: Variability in point cloud data quality, density, and coverage necessitates the development of methods that are robust to these fluctuations, as exemplified by models like 3DTI-Net [51].
- Real-time processing: For applications such as autonomous driving or augmented reality, real-time processing of point clouds is crucial. Current graph-based methods generally lack optimization for real-time performance due to their computational demands.
- Handling of high-dimensional data: Efficiently managing and processing high-dimensional data without losing essential information remains a technical challenge. Techniques that reduce dimensionality while retaining critical details, such as manifold learning, are of interest.
- Integration of global and local features: There is an ongoing need to effectively integrate global and local features to enhance model descriptive power. Future developments may focus on more sophisticated architectures that seamlessly combine these feature levels.
- Advancements in graph convolutional techniques: Future research is likely to concentrate on developing more sophisticated graph convolutional techniques that can better capture the complex structures and relationships within point clouds.
- Enhanced learning paradigms: Moving beyond supervised learning paradigms, there is a potential trend towards more adaptive, continual learning frameworks that can evolve and improve as they are exposed to new data without the need for retraining from scratch.
- Quantitative evaluation of graph-based processing: Many applications require precise and accurate processing of point clouds. Determining how to construct graph structures on point clouds to ensure their accurate and precise representation, and how to quantitatively evaluate graph-based methods, is crucial for ensuring the interpretability and effectiveness of these methods.

#### 5. Conclusions

In this paper, we comprehensively collected graph neural networks and graph-based methods across a spectrum of tasks, marking the first extensive review from a task-oriented perspective on the deployment of GNNs and graph transformers within point cloud contexts. Specifically, we categorized various graph-based methods into nine main categories according to point cloud downstream tasks: classification, segmentation, object recognition

and tracking, registration, generation, completion and sampling, denoising, compression and prediction, and advanced applications. We systematically organized the methodologies for each task and conducted a qualitative comparison between graph methods and conventional deep learning approaches, discussing their respective advantages and disadvantages. Focusing on current trends and the various challenges facing GNN methods in point clouds, we demonstrated innovative advancements in handling complex point cloud data structures, hoping to provide insights for the future development of point cloud processing methods based on graph neural networks.

**Author Contributions:** Conceptualization, D.L.; investigation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, D.L., C.L., J.G. and J.Z.; supervision, J.D. and Z.C.; project administration, D.L.; funding acquisition, D.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grants 42201475, in part by the Natural Science Foundation of Fujian Province under Grant 2021J05059, in part by the Fundamental Research Funds for the Central Universities of Huaqiao University (ZQN-1114), in part by the National Natural Science Foundation of China under Grants 62001175, in part by the Natural Science Foundation of Fujian Province under Grants 2023J01135. The authors would like to acknowledge the anonymous reviewers for their valuable comments in improving the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. 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 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
2. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*; MIT: Cambridge, MA, USA, 2017; Volume 30, pp. 5099–5108.
3. Chen, C.; Wu, Y.; Dai, Q.; Zhou, H.Y.; Xu, M.; Yang, S.; Han, X.; Yu, Y. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. *arXiv* **2022**, arXiv:2209.13232.
4. 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.* **2020**, 43, 4338–4364. [[CrossRef](#)]
5. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*; MIT: Cambridge, MA, USA, 2016; Volume 29, pp. 3844–3852.
6. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
7. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*; MIT: Cambridge, MA, USA, 2017; Volume 30, p. 4755450.
8. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
9. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 16259–16268.
10. 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](#)]
11. Park, C.; Jeong, Y.; Cho, M.; Park, J. Fast point transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16949–16958.
12. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8500–8509.
13. Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; Zhao, H. Point transformer v2: Grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems*; MIT: Cambridge, MA, USA, 2022; Volume 35, pp. 33330–33342.
14. Wu, X.; Jiang, L.; Wang, P.S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; Zhao, H. Point Transformer V3: Simpler Faster Stronger. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 4840–4851.
15. 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 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
16. Uy, M.A.; Pham, Q.H.; Hua, B.S.; Nguyen, T.; Yeung, S.K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1588–1597.

17. 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:1512.03012.
18. 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 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543.
19. 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 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5828–5839.
20. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv* **2017**, arXiv:1704.03847.
21. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
22. 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 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9297–9307.
23. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 202–203.
24. Roynard, 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](#)]
25. Siddiqi, K.; Zhang, J.; Macrini, D.; Shokoufandeh, A.; Bouix, S.; Dickinson, S. Retrieving articulated 3-D models using medial surfaces. *Mach. Vis. Appl.* **2008**, *19*, 261–275. [[CrossRef](#)]
26. 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, Sydney, NSW, Australia, 2–4 December 2013; Volume 2.
27. Rottensteiner, F.; Sohn, G.; Jung, J.; Gerke, M.; Baillard, C.; Benitez, S.; Breitkopf, U. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. I-3* **2012**, *1*, 293–298. [[CrossRef](#)]
28. 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 4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014, Loire Valley, France, 6–8 March 2014.
29. Vallet, B.; Brédif, M.; Serna, A.; Marcotegui, B.; Paparoditis, N. TerraMobilita/iQmulus urban point cloud analysis benchmark. *Comput. Graph.* **2015**, *49*, 126–133. [[CrossRef](#)]
30. Varney, N.; Asari, V.K.; Graehling, Q. DALES: A large-scale aerial LiDAR data set for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 186–187.
31. Pan, Y.; Gao, B.; Mei, J.; Geng, S.; Li, C.; Zhao, H. Semanticpos: A point cloud dataset with large quantity of dynamic instances. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 687–693.
32. Chen, M.; Hu, Q.; Yu, Z.; Thomas, H.; Feng, A.; Hou, Y.; McCullough, K.; Ren, F.; Soibelman, L. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. *arXiv* **2022**, arXiv:2203.09065.
33. Li, M.; Wu, Y.; Yeh, A.G.; Xue, F. HRHD-HK: A benchmark dataset of high-rise and high-density urban scenes for 3D semantic segmentation of photogrammetric point clouds. *arXiv* **2023**, arXiv:2307.07976.
34. Cheung, K.L.; Lee, C.C. ARCH2S: Dataset, Benchmark and Challenges for Learning Exterior Architectural Structures from Point Clouds. *arXiv* **2024**, arXiv:2406.01337.
35. Gaydon, C.; Daab, M.; Roche, F. FRACTAL: An Ultra-Large-Scale Aerial Lidar Dataset for 3D Semantic Segmentation of Diverse Landscapes. *arXiv* **2024**, arXiv:2405.04634.
36. Song, S.; Lichtenberg, S.P.; Xiao, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 567–576.
37. 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 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9552–9557.
38. Chang, M.F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3d tracking and forecasting with rich maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8748–8757.
39. 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. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 2267–2273.

40. Sun, P.; Kretzschmar, 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
41. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
42. Schumann, O.; Hahn, M.; Scheiner, N.; Weishaupt, F.; Tilly, J.F.; Dickmann, J.; Wöhler, C. Radarscenes: A real-world radar point cloud data set for automotive applications. In Proceedings of the 2021 IEEE 24th International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 1–4 November 2021; pp. 1–8.
43. Matuszka, T.; Barton, I.; Butykai, Á.; Hajas, P.; Kiss, D.; Kovács, D.; Kunsagi-Máté, S.; Lengyel, P.; Németh, G.; Pető, L.; et al. aimotive dataset: A multimodal dataset for robust autonomous driving with long-range perception. *arXiv* **2022**, arXiv:2211.09445.
44. Zhang, A.; Eranki, C.; Zhang, C.; Park, J.H.; Hong, R.; Kalyani, P.; Kalyanaraman, L.; Gamare, A.; Bagad, A.; Esteve, M.; et al. Towards robust robot 3d perception in urban environments: The ut campus object dataset. In *IEEE Transactions on Robotics*; IEEE: New York City, NY, USA, 2024.
45. Zhang, X.; Wang, L.; Chen, J.; Fang, C.; Yang, L.; Song, Z.; Yang, G.; Wang, Y.; Zhang, X.; Li, J. Dual radar: A multi-modal dataset with dual 4d radar for autonomous driving. *arXiv* **2023**, arXiv:2310.07602.
46. Te, G.; Hu, W.; Zheng, A.; Guo, Z. Rgcn: Regularized graph cnn for point cloud segmentation. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Republic of Korea, 22–26 October 2018; pp. 746–754.
47. Xie, Z.; Chen, J.; Peng, B. Point clouds learning with attention-based graph convolution networks. *Neurocomputing* **2020**, 402, 245–255. [[CrossRef](#)]
48. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3558–3565.
49. Wang, C.; Samari, B.; Siddiqi, K. Local spectral graph convolution for point set feature learning. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 52–66.
50. Zhang, Y.; Rabbat, M. A graph-cnn for 3d point cloud classification. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 6279–6283.
51. Pan, G.; Wang, J.; Ying, R.; Liu, P. 3dti-net: Learn inner transform invariant 3d geometry features using dynamic gcn. *arXiv* **2018**, arXiv:1812.06254.
52. Wang, Q.; Zhang, X.; Gu, Y. Spatial-Spectral Smooth Graph Convolutional Network for Multispectral Point Cloud Classification. In Proceedings of the IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 1062–1065.
53. Yi, L.; Su, H.; Guo, X.; Guibas, L.J. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2282–2290.
54. Lu, Q.; Chen, C.; Xie, W.; Luo, Y. PointNGCNN: Deep convolutional networks on 3D point clouds with neighborhood graph filters. *Comput. Graph.* **2020**, 86, 42–51. [[CrossRef](#)]
55. Chen, S.; Duan, C.; Yang, Y.; Li, D.; Feng, C.; Tian, D. Deep unsupervised learning of 3D point clouds via graph topology inference and filtering. *IEEE Trans. Image Process.* **2019**, 29, 3183–3198. [[CrossRef](#)] [[PubMed](#)]
56. Huang, H.; Fang, Y. Adaptive wavelet transformer network for 3d shape representation learning. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021.
57. Wen, C.; Long, J.; Yu, B.; Tao, D. PointWavelet: Learning in Spectral Domain for 3-D Point Cloud Analysis. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: New York City, NY, USA, 2024.
58. Yi, Y.; Lu, X.; Gao, S.; Robles-Kelly, A.; Zhang, Y. Graph classification via discriminative edge feature learning. *Pattern Recognit.* **2023**, 143, 109799. [[CrossRef](#)]
59. Wu, B.; Lang, B. MSGCN: A multiscale spatio graph convolution network for 3D point clouds. *Multimed. Tools Appl.* **2023**, 82, 35949–35968. [[CrossRef](#)]
60. Chen, L.; Wei, G.; Wang, Z. PointAGCN: Adaptive Spectral Graph CNN for Point Cloud Feature Learning. In Proceedings of the 2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Jinan, China, 14–17 December 2018; pp. 401–406.
61. Simonovsky, M.; Komodakis, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3693–3702.
62. Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; Qiao, Y. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 87–102.
63. Dominguez, M.; Dhamdhhere, R.; Petkar, A.; Jain, S.; Sah, S.; Ptucha, R. General-purpose deep point cloud feature extractor. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Munich, Germany, 8–14 September 2018; pp. 1972–1981.
64. Shen, Y.; Feng, C.; Yang, Y.; Tian, D. Mining point cloud local structures by kernel correlation and graph pooling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4548–4557.
65. Chen, C.; Fragonara, L.Z.; Tsourdos, A. GAPointNet: Graph attention based point neural network for exploiting local feature of point cloud. *Neurocomputing* **2021**, 438, 122–132. [[CrossRef](#)]

66. 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.* **2019**, *38*, 1–12. [[CrossRef](#)]
67. Zhang, K.; Hao, M.; Wang, J.; de Silva, C.W.; Fu, C. Linked dynamic graph cnn: Learning on point cloud via linking hierarchical features. *arXiv* **2019**, arXiv:1904.10014.
68. Guo, R.; Zhou, Y.; Zhao, J.; Man, Y.; Liu, M.; Yao, R.; Liu, B. Point cloud classification by dynamic graph CNN with adaptive feature fusion. *IET Comput. Vis.* **2021**, *15*, 235–244. [[CrossRef](#)]
69. Cui, Y.; Liu, X.; Liu, H.; Zhang, J.; Zare, A.; Fan, B. Geometric attentional dynamic graph convolutional neural networks for point cloud analysis. *Neurocomputing* **2021**, *432*, 300–310. [[CrossRef](#)]
70. Hassani, K.; Haley, M. Unsupervised multi-task feature learning on point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8160–8171.
71. Zhang, L.; Zhu, Z. Unsupervised feature learning for point cloud by contrasting and clustering with graph convolutional neural network. *arXiv* **2019**, arXiv:1904.12359.
72. Liu, J.; Ni, B.; Li, C.; Yang, J.; Tian, Q. Dynamic points agglomeration for hierarchical point sets learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7546–7555.
73. Sun, X.; Lian, Z.; Xiao, J. Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 980–988.
74. Bazazian, D.; Nahata, D. DCG-net: Dynamic capsule graph convolutional network for point clouds. *IEEE Access* **2020**, *8*, 188056–188067. [[CrossRef](#)]
75. Kim, S.; Park, J.; Han, B. Rotation-invariant local-to-global representation learning for 3d point cloud. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 8174–8185.
76. Wen, C.; Li, X.; Yao, X.; Peng, L.; Chi, T. Airborne LiDAR point cloud classification with global-local graph attention convolution neural network. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 181–194. [[CrossRef](#)]
77. Li, D.; Shen, X.; Yu, Y.; Guan, H.; Wang, H.; Li, D. GGM-net: Graph geometric moments convolution neural network for point cloud shape classification. *IEEE Access* **2020**, *8*, 124989–124998. [[CrossRef](#)]
78. Nezhadarya, E.; Taghavi, E.; Razani, R.; Liu, B.; Luo, J. Adaptive hierarchical down-sampling for point cloud classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12956–12964.
79. Zhai, Z.; Zhang, X.; Yao, L. Multi-scale dynamic graph convolution network for point clouds classification. *IEEE Access* **2020**, *8*, 65591–65598. [[CrossRef](#)]
80. Lei, H.; Akhtar, N.; Mian, A. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3664–3680. [[CrossRef](#)] [[PubMed](#)]
81. Lin, Z.H.; Huang, S.Y.; Wang, Y.C.F. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1800–1809.
82. Li, X.A.; Wang, L.Y.; Lu, J. Multiscale receptive fields graph attention network for point cloud classification. *Complexity* **2021**, *2021*, 1–9. [[CrossRef](#)]
83. Yang, D.; Gao, W. Pointmanifold: Using manifold learning for point cloud classification. *arXiv* **2020**, arXiv:2010.07215.
84. Wang, H.; Zhang, Y.; Liu, W.; Gu, X.; Jing, X.; Liu, Z. A novel GCN-based point cloud classification model robust to pose variances. *Pattern Recognit.* **2022**, *121*, 108251. [[CrossRef](#)]
85. Zhang, W.; Su, S.; Wang, B.; Hong, Q.; Sun, L. Local k-nns pattern in omni-direction graph convolution neural network for 3d point clouds. *Neurocomputing* **2020**, *413*, 487–498. [[CrossRef](#)]
86. Dominguez, M.; Ptucha, R. Directional graph networks with hard weight assignments. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 7439–7446.
87. Wang, W.; You, Y.; Liu, W.; Lu, C. Point cloud classification with deep normalized Reeb graph convolution. *Image Vis. Comput.* **2021**, *106*, 104092. [[CrossRef](#)]
88. Li, R.; Zhang, Y.; Niu, D.; Yang, G.; Zafar, N.; Zhang, C.; Zhao, X. PointVGG: Graph convolutional network with progressive aggregating features on point clouds. *Neurocomputing* **2021**, *429*, 187–198. [[CrossRef](#)]
89. Srivastava, S.; Sharma, G. Exploiting local geometry for feature and graph construction for better 3d point cloud processing with graph neural networks. In Proceedings of the 2021 IEEE INternational Conference on Robotics and Automation (ICRA), Xian, China, 30 May–5 June 2021; pp. 12903–12909.
90. Zhao, P.; Guan, H.; Li, D.; Yu, Y.; Wang, H.; Gao, K.; Junior, J.M.; Li, J. Airborne multispectral LiDAR point cloud classification with a feature Reasoning-based graph convolution network. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *105*, 102634. [[CrossRef](#)]
91. Xu, Q.; Sun, X.; Wu, C.Y.; Wang, P.; Neumann, U. Grid-gcn for fast and scalable point cloud learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5661–5670.
92. Hu, H.; Wang, F.; Le, H. Va-gcn: A vector attention graph convolution network for learning on point clouds. *arXiv* **2021**, arXiv:2106.00227.
93. Kumar, S.; Katragadda, S.R.; Abdul, A.; Reddy, V.D. Extended Graph Convolutional Networks for 3D Object Classification in Point Clouds. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*. [[CrossRef](#)]

94. Xun, L.; Feng, X.; Chen, C.; Yuan, X.; Lu, Q. Graph attention-based deep neural network for 3d point cloud processing. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; pp. 1–6.
95. Huang, C.Q.; Jiang, F.; Huang, Q.H.; Wang, X.Z.; Han, Z.M.; Huang, W.Y. Dual-graph attention convolution network for 3-D point cloud classification. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: New York City, NY, USA, 2022.
96. Yue, Y.; Li, X.; Peng, Y. A 3D Point Cloud Classification Method Based on Adaptive Graph Convolution and Global Attention. *Sensors* **2024**, *24*, 617. [[CrossRef](#)]
97. Jing, W.; Zhang, W.; Li, L.; Di, D.; Chen, G.; Wang, J. AGNet: An attention-based graph network for point cloud classification and segmentation. *Remote Sens.* **2022**, *14*, 1036. [[CrossRef](#)]
98. Zhang, C.; Chen, H.; Wan, H.; Yang, P.; Wu, Z. Graph-pbn: Graph-based parallel branch network for efficient point cloud learning. *Graph. Model.* **2022**, *119*, 101120. [[CrossRef](#)]
99. Lu, D.; Xie, Q.; Gao, K.; Xu, L.; Li, J. 3DCTN: 3D convolution-transformer network for point cloud classification. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24854–24865. [[CrossRef](#)]
100. Liu, S.; Liu, D.; Chen, C.; Xu, C. SGCCNN for 3D point cloud classification. In Proceedings of the 2022 14th International Conference on Machine Learning and Computing, Guangzhou, China, 18–21 February 2022; pp. 419–423.
101. Lin, M.; Feragen, A. DiffConv: Analyzing irregular point clouds with an irregular view. In Proceedings of the European Conference on Computer Vision. Springer, Tel Aviv, Israel, 23–27 October 2022; pp. 380–397.
102. Lin, Z.H.; Huang, S.Y.; Wang, Y.C.F. Learning of 3d graph convolution networks for point cloud analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 4212–4224. [[CrossRef](#)]
103. Tamajo, A.; Plaß, B.; Klauer, T. Shrinking unit: A Graph Convolution-Based Unit for CNN-like 3D Point Cloud Feature Extractors. *arXiv* **2022**, arXiv:2209.12770.
104. Li, Y.; Tanaka, Y. Structure-Aware Multi-Hop Graph Convolution for Graph Neural Networks. *IEEE Access* **2022**, *10*, 16624–16633. [[CrossRef](#)]
105. Wei, M.; Wei, Z.; Zhou, H.; Hu, F.; Si, H.; Chen, Z.; Zhu, Z.; Qiu, J.; Yan, X.; Guo, Y.; et al. AGConv: Adaptive graph convolution on 3D point clouds. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: New York City, NY, USA, 2023.
106. Khodadad, M.; Rezanejad, M.; Kasmaee, A.S.; Siddiqi, K.; Walther, D.; Mahyar, H. MLGCN: An Ultra Efficient Graph Convolution Neural Model For 3D Point Cloud Analysis. *arXiv* **2023**, arXiv:2303.17748.
107. Qin, S.; Li, Z.; Liu, L. Robust 3D shape classification via non-local graph attention network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5374–5383.
108. Qi, X.; Liao, R.; Jia, J.; Fidler, S.; Urtasun, R. 3d graph neural networks for rgbd semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5199–5208.
109. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4558–4567.
110. Liang, Z.; Yang, M.; Deng, L.; Wang, C.; Wang, B. Hierarchical depthwise graph convolutional neural network for 3D semantic segmentation of point clouds. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, BC, Canada, 20–24 May 2019; pp. 8152–8158.
111. Landrieu, L.; Boussaha, M. Point cloud oversegmentation with graph-structured deep metric learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 7440–7449.
112. Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; Shan, J. Graph attention convolution for point cloud semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10296–10305.
113. Li, Z.; Zhang, J.; Li, G.; Liu, Y.; Li, S. Graph attention neural networks for point cloud recognition. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 387–392.
114. Zhao, H.; Jiang, L.; Fu, C.W.; Jia, J. Pointweb: Enhancing local neighborhood features for point cloud processing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5565–5573.
115. 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 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 10433–10441.
116. Han, W.; Wen, C.; Wang, C.; Li, X.; Li, Q. Point2node: Correlation learning of dynamic-node for point cloud feature modeling. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10925–10932.
117. Lei, H.; Akhtar, N.; Mian, A. Seggcnn: Efficient 3d point cloud segmentation with fuzzy spherical kernel. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11611–11620.
118. Song, W.; Liu, Z.; Guo, Y.; Sun, S.; Zu, G.; Li, M. DGPolarnet: Dynamic graph convolution network for LiDAR point cloud semantic segmentation on polar BEV. *Remote Sens.* **2022**, *14*, 3825. [[CrossRef](#)]
119. Zhang, J.; Hu, X.; Dai, H. A graph-voxel joint convolution neural network for ALS point cloud segmentation. *IEEE Access* **2020**, *8*, 139781–139791. [[CrossRef](#)]
120. Ma, Y.; Guo, Y.; Liu, H.; Lei, Y.; Wen, G. Global context reasoning for semantic segmentation of 3D point clouds. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Seattle, WA, USA, 13–19 June 2020; pp. 2931–2940.

121. Khan, S.A.; Shi, Y.; Shahzad, M.; Zhu, X.X. FGCN: Deep feature-based graph convolutional network for semantic segmentation of urban 3D point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 198–199.
122. Li, Y.; Ma, L.; Zhong, Z.; Cao, D.; Li, J. TGNet: Geometric graph CNN on 3-D point cloud segmentation. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3588–3600. [[CrossRef](#)]
123. Wang, H.; Rong, X.; Yang, L.; Feng, J.; Xiao, J.; Tian, Y. Weakly supervised semantic segmentation in 3d graph-structured point clouds of wild scenes. *arXiv* **2020**, arXiv:2004.12498.
124. Liang, Z.; Yang, M.; Wang, C. 3D Graph Embedding Learning with a Structure-aware Loss Function for Point Cloud Semantic Instance Segmentation. *arXiv* **2019**, arXiv:1902.05247.
125. Sun, Y.; Miao, Y.; Chen, J.; Pajarola, R. PGCNet: Patch graph convolutional network for point cloud segmentation of indoor scenes. *Vis. Comput.* **2020**, *36*, 2407–2418. [[CrossRef](#)]
126. Chen, C.; Qian, S.; Fang, Q.; Xu, C. HAPGN: Hierarchical attentive pooling graph network for point cloud segmentation. *IEEE Trans. Multimed.* **2020**, *23*, 2335–2346. [[CrossRef](#)]
127. Du, Z.; Ye, H.; Cao, F. A novel local-global graph convolutional method for point cloud semantic segmentation. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: New York City, NY, USA, 2022.
128. Mao, Y.; Sun, X.; Chen, K.; Diao, W.; Guo, Z.; Lu, X.; Fu, K. Semantic segmentation for point cloud scenes via dilated graph feature aggregation and pyramid decoders. *arXiv* **2022**, arXiv:2204.04944.
129. Meraz, M.; Ansari, M.A.; Javed, M.; Chakraborty, P. DC-GNN: Drop channel graph neural network for object classification and part segmentation in the point cloud. *Int. J. Multimed. Inf. Retr.* **2022**, *11*, 123–133. [[CrossRef](#)]
130. Tao, A.; Duan, Y.; Wei, Y.; Lu, J.; Zhou, J. Seggroup: Seg-level supervision for 3d instance and semantic segmentation. *IEEE Trans. Image Process.* **2022**, *31*, 4952–4965. [[CrossRef](#)]
131. Zeng, Z.; Xu, Y.; Xie, Z.; Wan, J.; Wu, W.; Dai, W. Rg-gcn: A random graph based on graph convolution network for point cloud semantic segmentation. *Remote Sens.* **2022**, *14*, 4055. [[CrossRef](#)]
132. Chen, J.; Chen, Y.; Wang, C. Feature graph convolution network with attentive fusion for large-scale point clouds semantic segmentation. In *IEEE Geoscience and Remote Sensing Letters*; IEEE: New York City, NY, USA, 2023.
133. Wang, X.; Yang, J.; Kang, Z.; Du, J.; Tao, Z.; Qiao, D. A category-contrastive guided-graph convolutional network approach for the semantic segmentation of point clouds. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*; IEEE: New York City, NY, USA, 2023.
134. Zhang, N.; Pan, Z.; Li, T.H.; Gao, W.; Li, G. Improving graph representation for point cloud segmentation via attentive filtering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1244–1254.
135. Zhiheng, K.; Ning, L. PyramNet: Point cloud pyramid attention network and graph embedding module for classification and segmentation. *arXiv* **2019**, arXiv:1906.03299.
136. Zhou, W.; Wang, Q.; Jin, W.; Shi, X.; Wang, D.; Hao, X.; Yu, Y. GTNet: Graph transformer network for 3D point cloud classification and semantic segmentation. *arXiv* **2023**, arXiv:2305.15213.
137. Yang, F.; Davoine, F.; Wang, H.; Jin, Z. Continuous conditional random field convolution for point cloud segmentation. *Pattern Recognit.* **2022**, *122*, 108357. [[CrossRef](#)]
138. Robert, D.; Raguet, H.; Landrieu, L. Efficient 3d semantic segmentation with superpoint transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 17195–17204.
139. Xu, M.; Ding, R.; Zhao, H.; Qi, X. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3173–3182.
140. Chen, C.; Li, G.; Xu, R.; Chen, T.; Wang, M.; Lin, L. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4994–5002.
141. Zarzar, J.; Giancola, S.; Ghanem, B. PointRGCN: Graph convolution networks for 3D vehicles detection refinement. *arXiv* **2019**, arXiv:1911.12236.
142. Feng, M.; Gilani, S.Z.; Wang, Y.; Zhang, L.; Mian, A. Relation graph network for 3D object detection in point clouds. *IEEE Trans. Image Process.* **2020**, *30*, 92–107. [[CrossRef](#)] [[PubMed](#)]
143. Shi, W.; Rajkumar, R. Point-gnn: Graph neural network for 3d object detection in a point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–17 June 2020; pp. 1711–1719.
144. Liang, M.; Yang, B.; Wang, S.; Urtasun, R. Deep continuous fusion for multi-sensor 3d object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 641–656.
145. Huang, P.H.; Lee, H.H.; Chen, H.T.; Liu, T.L. Text-guided graph neural networks for referring 3d instance segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 1610–1618.
146. Tian, Y.; Chen, L.; Song, W.; Sung, Y.; Woo, S. Dgcb-net: Dynamic graph convolutional broad network for 3d object recognition in point cloud. *Remote Sens.* **2020**, *13*, 66. [[CrossRef](#)]

147. Chen, J.; Lei, B.; Song, Q.; Ying, H.; Chen, D.Z.; Wu, J. A hierarchical graph network for 3d object detection on point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 392–401.
148. Svenningsson, P.; Fioranelli, F.; Yarovoy, A. Radar-pointgnn: Graph based object recognition for unstructured radar point-cloud data. In Proceedings of the 2021 IEEE Radar Conference (RadarConf21), Atlanta, GA, USA, 7–14 May 2021; pp. 1–6.
149. Zhang, Y.; Huang, D.; Wang, Y. PC-RGNN: Point cloud completion and graph neural network for 3D object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 3430–3437.
150. Feng, M.; Li, Z.; Li, Q.; Zhang, L.; Zhang, X.; Zhu, G.; Zhang, H.; Wang, Y.; Mian, A. Free-form description guided 3d visual graph network for object grounding in point cloud. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3722–3731.
151. Wang, Y.; Solomon, J.M. Object dgcn: 3d object detection using dynamic graphs. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20745–20758.
152. He, Q.; Wang, Z.; Zeng, H.; Zeng, Y.; Liu, Y. Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 870–878.
153. Xiong, S.; Li, B.; Zhu, S. DCGNN: A single-stage 3D object detection network based on density clustering and graph neural network. *Complex Intell. Syst.* **2022**, *9*, 251776670. [[CrossRef](#)]
154. Yin, J.; Shen, J.; Gao, X.; Crandall, D.J.; Yang, R. Graph neural network and spatiotemporal transformer attention for 3D video object detection from point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *45*, 9822–9835. [[CrossRef](#)] [[PubMed](#)]
155. Shu, D.W.; Kwon, J. Hierarchical bidirected graph convolutions for large-scale 3-D point cloud place recognition. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: New York City, NY, USA, 2023.
156. Sun, Q.; Liu, H.; He, J.; Fan, Z.; Du, X. Dagc: Employing dual attention and graph convolution for point cloud based place recognition. In Proceedings of the 2020 International Conference on Multimedia Retrieval, Dublin, Ireland, 8–11 June 2020; pp. 224–232.
157. Hui, L.; Yang, H.; Cheng, M.; Xie, J.; Yang, J. Pyramid point cloud transformer for large-scale place recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6098–6107.
158. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
159. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10529–10538.
160. Xie, Q.; Lai, Y.K.; Wu, J.; Wang, Z.; Zhang, Y.; Xu, K.; Wang, J. Mlcvnet: Multi-level context votenet for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10447–10456.
161. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11108–11117.
162. Fu, K.; Liu, S.; Luo, X.; Wang, M. Robust point cloud registration framework based on deep graph matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8893–8902.
163. Chang, S.; Ahn, C.; Lee, M.; Oh, S. Graph-matching-based correspondence search for nonrigid point cloud registration. *Comput. Vis. Image Underst.* **2020**, *192*, 102899. [[CrossRef](#)]
164. Li, J.; Zhao, P.; Hu, Q.; Ai, M. Robust point cloud registration based on topological graph and cauchy weighted lq-norm. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 244–259. [[CrossRef](#)]
165. Li, J.; Hu, Q.; Ai, M. GESAC: Robust graph enhanced sample consensus for point cloud registration. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 363–374. [[CrossRef](#)]
166. Saleh, M.; Dehghani, S.; Busam, B.; Navab, N.; Tombari, F. Graphite: Graph-induced feature extraction for point cloud registration. In Proceedings of the 2020 International Conference on 3D Vision (3DV), Fukuoka, Japan, 20–28 November 2020; pp. 241–251.
167. Shi, C.; Chen, X.; Huang, K.; Xiao, J.; Lu, H.; Stachniss, C. Keypoint matching for point cloud registration using multiplex dynamic graph attention networks. *IEEE Robot. Autom. Lett.* **2021**, *6*, 8221–8228. [[CrossRef](#)]
168. Lai-Dang, Q.V.; Nengroo, S.H.; Jin, H. Learning dense features for point cloud registration using a graph attention network. *Appl. Sci.* **2022**, *12*, 7023. [[CrossRef](#)]
169. Song, Y.; Shen, W.; Peng, K. A novel partial point cloud registration method based on graph attention network. *Vis. Comput.* **2023**, *39*, 1109–1120. [[CrossRef](#)]
170. Zaman, A.; Yangyu, F.; Ayub, M.S.; Irfan, M.; Guoyun, L.; Shiya, L. CMDGAT: Knowledge extraction and retention based continual graph attention network for point cloud registration. *Expert Syst. Appl.* **2023**, *214*, 119098. [[CrossRef](#)]
171. Sun, L.; Zhang, Z.; Zhong, R.; Chen, D.; Zhang, L.; Zhu, L.; Wang, Q.; Wang, G.; Zou, J.; Wang, Y. A weakly supervised graph deep learning framework for point cloud registration. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. [[CrossRef](#)]
172. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In *Proceedings of the Sensor fusion IV: Control Paradigms and Data Structures*; Spie: Bellingham, WA, USA, 1992; Volume 1611, pp. 586–606.

173. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. Pointnetlk: Robust & efficient point cloud registration using pointnet. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7163–7172.
174. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 3523–3532.
175. Pais, G.D.; Ramalingam, S.; Govindu, V.M.; Nascimento, J.C.; Chellappa, R.; Miraldo, P. 3dregnet: A deep neural network for 3d point registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7193–7203.
176. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [[CrossRef](#)]
177. Valsesia, D.; Fracastoro, G.; Magli, E. Learning localized generative models for 3d point clouds via graph convolution. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
178. Shu, D.W.; Park, S.W.; Kwon, J. 3d point cloud generative adversarial network based on tree structured graph convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3859–3868.
179. Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N.; Guibas, L.J. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv* **2019**, arXiv:1908.00575.
180. Li, Y.; Baciu, G. HSGAN: Hierarchical graph learning for point cloud generation. *IEEE Trans. Image Process.* **2021**, *30*, 4540–4554. [[CrossRef](#)] [[PubMed](#)]
181. Xiaomao, Z.; Wei, W.; Bing, D. PSG-GAN: Progressive Person Image Generation with Self-Guided Local Focuses. In Proceedings of the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), Washington, DC, USA, 1–3 November 2021; pp. 763–769.
182. Liu, X.; Kong, X.; Liu, L.; Chiang, K. TreeGAN: Syntax-aware sequence generation with generative adversarial networks. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 1140–1145.
183. Yang, G.; Huang, X.; Hao, Z.; Liu, M.Y.; Belongie, S.; Hariharan, B. Pointflow: 3d point cloud generation with continuous normalizing flows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4541–4550.
184. Chen, S.; Tian, D.; Feng, C.; Vetro, A.; Kovačević, J. Fast resampling of three-dimensional point clouds via graphs. *IEEE Trans. Signal Process.* **2017**, *66*, 666–681. [[CrossRef](#)]
185. Hu, W.; Fu, Z.; Guo, Z. Local frequency interpretation and non-local self-similarity on graph for point cloud inpainting. *IEEE Trans. Image Process.* **2019**, *28*, 4087–4100. [[CrossRef](#)]
186. Wu, H.; Zhang, J.; Huang, K. Point cloud super resolution with adversarial residual graph networks. *arXiv* **2019**, arXiv:1908.02111.
187. Pan, L. ECG: Edge-aware point cloud completion with graph convolution. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4392–4398. [[CrossRef](#)]
188. Zhu, L.; Wang, B.; Tian, G.; Wang, W.; Li, C. Towards point cloud completion: Point rank sampling and cross-cascade graph cnn. *Neurocomputing* **2021**, *461*, 1–16. [[CrossRef](#)]
189. Shi, J.; Xu, L.; Heng, L.; Shen, S. Graph-guided deformation for point cloud completion. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7081–7088. [[CrossRef](#)]
190. Wang, L.; Li, J.; Guo, S.; Han, S. A cascaded graph convolutional network for point cloud completion. In *The Visual Computer*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–16.
191. Qian, G.; Abualshour, A.; Li, G.; Thabet, A.; Ghanem, B. Pu-gcn: Point cloud upsampling using graph convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11683–11692.
192. Han, B.; Zhang, X.; Ren, S. PU-GACNet: Graph attention convolution network for point cloud upsampling. *Image Vis. Comput.* **2022**, *118*, 104371. [[CrossRef](#)]
193. Wang, H.; Zhang, C.; Chen, S.; Wang, H.; He, Q.; Mu, H. PU-FPG: Point cloud upsampling via form preserving graph convolutional networks. *J. Intell. Fuzzy Syst.* **2023**, *45*, 8595–8612. [[CrossRef](#)]
194. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. Pcn: Point completion network. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 728–737.
195. Yu, L.; Li, X.; Fu, C.W.; Cohen-Or, D.; Heng, P.A. Pu-net: Point cloud upsampling network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2790–2799.
196. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; Savarese, S. Topnet: Structural point cloud decoder. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 383–392.
197. Qian, Y.; Hou, J.; Kwong, S.; He, Y. PUGeo-Net: A geometry-centric network for 3D point cloud upsampling. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 752–769.
198. Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; Le, X. Pf-net: Point fractal network for 3d point cloud completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7662–7670.
199. Zhang, S.; Cui, S.; Ding, Z. Hypergraph spectral analysis and processing in 3D point cloud. *IEEE Trans. Image Process.* **2020**, *30*, 1193–1206. [[CrossRef](#)] [[PubMed](#)]

200. Schoenenberger, Y.; Paratte, J.; Vandergheynst, P. Graph-based denoising for time-varying point clouds. In Proceedings of the 2015 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), Lisbon, Portugal, 8–10 July 2015; pp. 1–4.
201. Dinesh, C.; Cheung, G.; Bajic, I.V.; Yang, C. Fast 3D point cloud denoising via bipartite graph approximation & total variation. *arXiv* **2018**, arXiv:1804.10831.
202. Hu, W.; Gao, X.; Cheung, G.; Guo, Z. Feature graph learning for 3D point cloud denoising. *IEEE Trans. Signal Process.* **2020**, *68*, 2841–2856. [[CrossRef](#)]
203. Pistilli, F.; Fracastoro, G.; Valsesia, D.; Magli, E. Learning graph-convolutional representations for point cloud denoising. In Proceedings of the European Conference on Computer Vision. Springer, Glasgow, UK, 23–28 August 2020; pp. 103–118.
204. Luo, S.; Hu, W. Differentiable manifold reconstruction for point cloud denoising. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle WA USA, 12–16 October 2020; pp. 1330–1338.
205. Irfan, M.A.; Magli, E. Exploiting color for graph-based 3D point cloud denoising. *J. Vis. Commun. Image Represent.* **2021**, *75*, 103027. [[CrossRef](#)]
206. Dinesh, C.; Cheung, G.; Bajic, I.V. Point cloud denoising via feature graph laplacian regularization. *IEEE Trans. Image Process.* **2020**, *29*, 4143–4158. [[CrossRef](#)] [[PubMed](#)]
207. Roveri, R.; Öztireli, A.C.; Pandele, I.; Gross, M. Pointpronet: Consolidation of point clouds with convolutional neural networks. In *Proceedings of the Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2018; Volume 37, pp. 87–99.
208. Hermosilla, P.; Ritschel, T.; Ropinski, T. Total denoising: Unsupervised learning of 3D point cloud cleaning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 52–60.
209. Zhang, C.; Florencio, D.; Loop, C. Point cloud attribute compression with graph transform. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 2066–2070.
210. Thanou, D.; Chou, P.A.; Frossard, P. Graph-based motion estimation and compensation for dynamic 3D point cloud compression. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3235–3239.
211. Cohen, R.A.; Tian, D.; Vetro, A. Attribute compression for sparse point clouds using graph transforms. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 1374–1378.
212. Shao, Y.; Zhang, Z.; Li, Z.; Fan, K.; Li, G. Attribute compression of 3D point clouds using Laplacian sparsity optimized graph transform. In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; pp. 1–4.
213. Gu, S.; Hou, J.; Zeng, H.; Yuan, H. 3D point cloud attribute compression via graph prediction. *IEEE Signal Process. Lett.* **2020**, *27*, 176–180. [[CrossRef](#)]
214. Gomes, P. Graph-based network for dynamic point cloud prediction. In Proceedings of the 12th ACM Multimedia Systems Conference, Istanbul, Turkey, 28 September–1 October 2021; pp. 393–397.
215. Gomes, P.; Rossi, S.; Toni, L. Spatio-temporal graph-RNN for point cloud prediction. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AL, USA, 19–22 September 2021; pp. 3428–3432.
216. Gao, L.; Fan, T.; Wan, J.; Xu, Y.; Sun, J.; Ma, Z. Point cloud geometry compression via neural graph sampling. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AL, USA, 19–22 September 2021; pp. 3373–3377.
217. Nguyen, D.T.; Quach, M.; Valenzise, G.; Duhamel, P. Lossless coding of point cloud geometry using a deep generative model. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4617–4629. [[CrossRef](#)]
218. Nguyen, D.T.; Quach, M.; Valenzise, G.; Duhamel, P. Multiscale deep context modeling for lossless point cloud geometry compression. In Proceedings of the 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shenzhen, China, 5–9 July 2021; pp. 1–6.
219. Chen, S.; Niu, S.; Lan, T.; Liu, B. PCT: Large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 4395–4399.
220. Geng, Y.; Wang, Z.; Jia, L.; Qin, Y.; Chai, Y.; Liu, K.; Tong, L. 3DGraphSeg: A unified graph representation-based point cloud segmentation framework for full-range highspeed railway environments. *IEEE Trans. Ind. Informatics* **2023**, *19*, 11430–11443. [[CrossRef](#)]
221. Hu, B.; Lei, B.; Shen, Y.; Liu, Y.; Wang, S. A point cloud generative model via tree-structured graph convolutions for 3D brain shape reconstruction. In Proceedings of the Pattern Recognition and Computer Vision: 4th Chinese Conference, PRCV 2021, Beijing, China, 29 October–1 November 2021; Proceedings, Part II 4; Springer: Berlin/Heidelberg, Germany, 2021; pp. 263–274.
222. Xing, J.; Yuan, H.; Hamzaoui, R.; Liu, H.; Hou, J. GQE-Net: A graph-based quality enhancement network for point cloud color attribute. *IEEE Trans. Image Process.* **2023**, *32*, 6303–6317. [[CrossRef](#)]
223. Feng, H.; Li, W.; Luo, Z.; Chen, Y.; Fatholahi, S.N.; Cheng, M.; Wang, C.; Junior, J.M.; Li, J. GCN-based pavement crack detection using mobile LiDAR point clouds. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 11052–11061. [[CrossRef](#)]
224. Li, Y.; Chen, H.; Cui, Z.; Timofte, R.; Pollefeys, M.; Chirikjian, G.S.; Van Gool, L. Towards efficient graph convolutional networks for point cloud handling. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3752–3762.

225. Tailor, S.A.; De Jong, R.; Azevedo, T.; Mattina, M.; Maji, P. Towards efficient point cloud graph neural networks through architectural simplification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2095–2104.
226. Zhang, J.F.; Zhang, Z. Exploration of energy-efficient architecture for graph-based point-cloud deep learning. In Proceedings of the 2021 IEEE Workshop on Signal Processing Systems (SiPS), Coimbra, Portugal, 19–21 October 2021; pp. 260–264.
227. Li, K.; Zhao, M.; Wu, H.; Yan, D.M.; Shen, Z.; Wang, F.Y.; Xiong, G. Graphfit: Learning multi-scale graph-convolutional representation for point cloud normal estimation. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 651–667.
228. Yang, Q.; Ma, Z.; Xu, Y.; Li, Z.; Sun, J. Inferring point cloud quality via graph similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 3015–3029. [[CrossRef](#)] [[PubMed](#)]
229. Shan, Z.; Yang, Q.; Ye, R.; Zhang, Y.; Xu, Y.; Xu, X.; Liu, S. Gpa-net: No-reference point cloud quality assessment with multi-task graph convolutional network. In *IEEE Transactions on Visualization and Computer Graphics*; IEEE: New York City, NY, USA, 2023.
230. Wang, Y.; Sun, W.; Jin, J.; Kong, Z.; Yue, X. MVGCN: Multi-view graph convolutional neural network for surface defect identification using three-dimensional point cloud. *J. Manuf. Sci. Eng.* **2023**, *145*, 031004. [[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.