Serialization based Point Cloud Oversegmentation

Chenghui Lu[1,2], Jianlong Kwan[1,2], Dilong Li[1,2*], Ziyi Chen[1,2], Haiyan Guan[3]
[1]Department of Computer Science & Xiamen CVPR Key Laboratory, Huaqiao University
[2]Fujian Key Lab. of Big Data Intell. and Security, Huaqiao University
[3]Nanjing University of Information Science and Technology
{23013083012, jianlongkwan}@stu.hqu.edu.cn, {scholar.dll, chenziyihq}@hqu.edu.cn

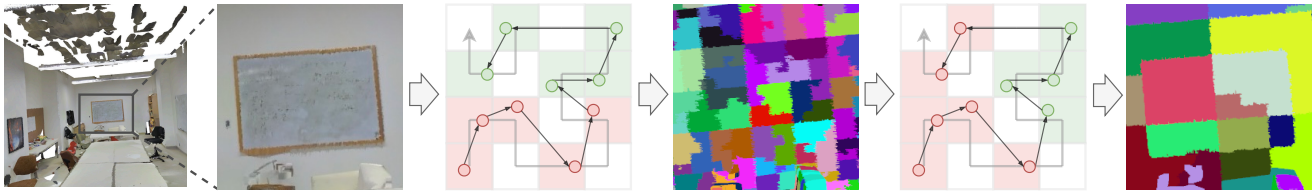


Figure 1. Our method efficiently generates superpoints from serialized point clouds by optimizing boundaries through reallocating points from each segment to the most appropriate surrounding segments (shown in green and red).

Abstract

Point cloud oversegmentation, as a fundamental preprocessing step for 3D understanding, is a challenging task due to its spatial proximity and semantic similarity requirements. Most existing works struggle to efficiently group semantically consistent points into superpoints while maintaining spatial proximity. In this paper, we propose a novel serialization based point cloud oversegmentation method, which leverages serialization to avoid complex spatial queries, directly accessing neighboring points through sequence locality for similarity matching and superpoint clustering. Specifically, we first serialize point clouds into a Hilbert curve and spatially-continuously partition them into initial segments. Then, to guarantee the internal semantic consistency of superpoints, we design an adaptive update algorithm that clusters superpoints by matching feature similarities between neighboring segments and refines segment features via Cross-Attention. Experiments on large-scale indoor and outdoor datasets demonstrate state-of-the-art performance in point cloud oversegmentation. Moreover, it is also adaptable to semantic segmentation and achieves promising performance. The code is available at https://github.com/CHL-glitch/SPCNet.

1. Introduction

Point cloud oversegmentation partitions point clouds into multiple spatially and semantically homogeneous regions, termed superpoints, which adaptively represent a set of points. Due to their representativeness, superpoint-based methods can significantly reduce computational complexity and improve semantic consistency. Therefore, it is becoming increasingly important in the field of point cloud processing and has attracted considerable attention from many researchers.

Early point cloud oversegmentation approaches primarily combined hand-crafted features with clustering or graph partitioning to generate superpoints. Papon et al. [26] introduced VCCS, which employed Fast Point Feature Histograms, selected seed points via octree structures, and performed K-means clustering. Lin et al. [22] reformulated the task as a subset selection problem while still using FPFH descriptors. Guinard et al. [12] simplified the process into a graph cut problem by extracting geometric features before applying a greedy algorithm. Similarly, Landrieu et al. [19] partitioned superpoints by solving an energy minimization problem with graph-based simplicity penalties. However, handcrafted features have limited representation, and the non-differentiable superpoint assignments prevent end-to-end training.

Recent point cloud oversegmentation methods generate superpoints using deep networks. Landrieu et al. [18] first used deep networks to extract point embeddings instead of handcrafted features, but their framework is not end-to-end. SPNet [15] proposed an end-to-end superpoint network that achieved clustering by iteratively constructing a

point-superpoint association map. However, their Farthest Point Sampling (FPS) based k-nearest neighbor (KNN) association map faced efficiency bottlenecks, and the unlimited range of original points being clustered resulted in irregular superpoint boundaries and even introduced noise. SuperLiDAR [16] combined Breadth-First Search (BFS) grouping with local discriminative loss to generate compact superpoints, and while it achieved $O(V + E)$ complexity on sparse LiDAR data, it may approach $O(V^2)$ in dense scenarios, limiting scalability. Those point-based methods design complex algorithms to cluster superpoints and/or inefficiently balance semantic consistency and spatial proximity, resulting in high computational burden.

To address this issue, we propose an efficient serialization based point cloud oversegmentation method. Inspired by PTv3 [38], we adopt a serialization approach instead of traditional point-based methods to avoid high computational complexity spatial queries such as FPS and KNN, as shown in Fig. 1. We find that one of the representative serialization strategies, the Hilbert curve [14], which has excellent locality-preserving properties, is crucial for spatial proximity in oversegmentation tasks. Thus, we apply the Hilbert curve to serialize point clouds and initialize superpoints. To achieve internal semantic consistency for the superpoints, we design an adaptive superpoint update algorithm that clusters semantically similar points within prepartitioned segments to update superpoints. Specifically, we use cosine similarity to evaluate the semantic similarity between points and nearby segments, which serves as the metric for associating points and superpoints to reallocate points, then apply Cross-Attention to update superpoint features. By combining these two steps and iterating them, the algorithm can obtain superpoints with both spatial proximity and semantic consistency.

Furthermore, to optimize superpoint feature representations, we employ a stacked Graph Transformer [42] and Graph Convolutional Network [42] to capture the global context between superpoints. Similar to prior work, SPCNet is designed as an end-to-end oversegmentation network and can also be integrated with downstream tasks such as semantic segmentation. To fully evaluate SPCNet, we conduct experiments on four large-scale datasets: S3DIS [1], ScanNet [9], nuScenes [3], and SemanticKITTI [2]. Experimental results demonstrate that SPCNet achieves state-of-the-art performance in all core oversegmentation metrics (BR, BP, and F1) across all datasets. The contributions of this paper are as follows:

- To our best knowledge, SPCNet is the first serialization based network for point cloud oversegmentation.
- We introduce a serialization strategy to initialize superpoints, which can efficiently produce superpoints with spontaneous spatial proximity.
- We propose an adaptive superpoint update algorithm to strengthen internal semantic consistency of superpoints.
- Our method achieves remarkable performance in point cloud oversegmentation across multiple datasets. It can be integrated with semantic segmentation networks and shows promising performance.

2. Related Work

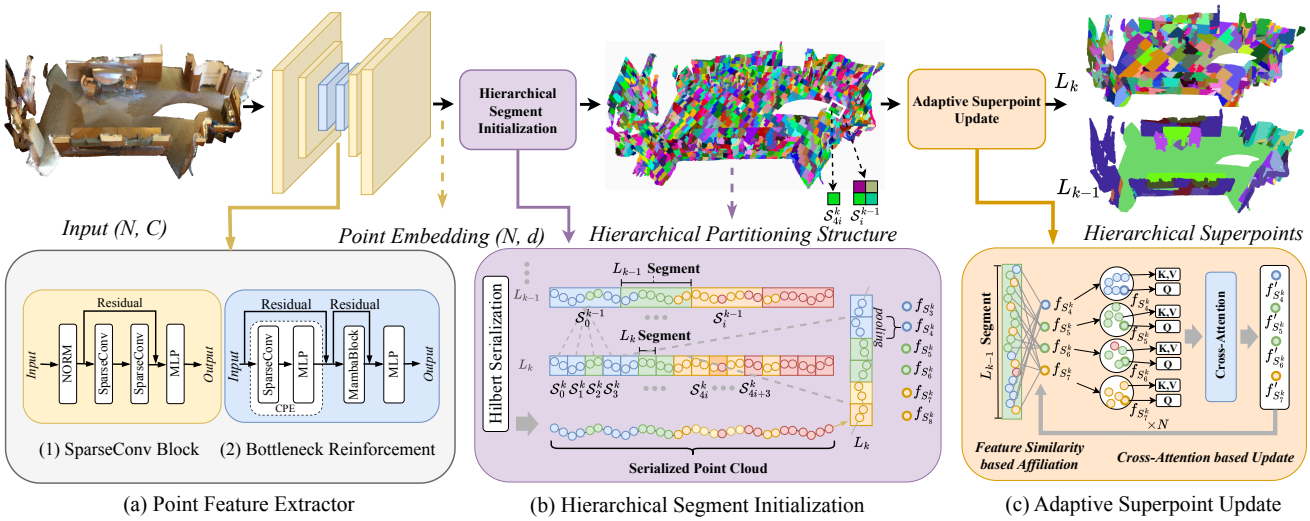
2.1. Deep Learning in Point Cloud

Deep learning approaches for 3D point cloud primarily follow three paradigms: projection-based, voxel-based, and point-based methods. Projection-based methods project unordered points onto 2D planes for CNN processing [6, 20, 21, 32], sacrificing 3D geometry for computational efficiency. Voxel-based methods discretize space into grids for 3D convolution [25, 31], improved by sparse convolution techniques [8, 10, 35] to mitigate memory costs, though kernel size limitations remain a constraint. Point-based methods [24, 27, 28, 33, 43] process raw point clouds directly and have evolved toward transformer architectures [13, 29, 37, 40, 44]. These methods offer powerful representation capabilities while struggling with scalability. The recently proposed SPT [29] leverages hierarchical superpoints and sparse self-attention to efficiently process large-scale scenes, achieving superior semantic segmentation with reduced model size and computational cost compared to traditional approaches.

2.2. Point Cloud Oversegmentation

Existing point cloud oversegmentation methods can be broadly categorized into optimization-based methods and deep learning-based approaches. Early methods such as Papon et al. [26] proposed the VCCS method, which used FPFH as input features, selected seed points using an octree, and performed point cloud superpoint segmentation using the K-means clustering algorithm. Lin et al. [22] viewed the superpoint segmentation problem as a subset selection problem and employed subset selection methods to generate superpoints, still using FPFH as point cloud feature descriptors. Guinard et al. [12] simplified the superpoint segmentation problem into a graph cut problem, extracting features such as local linearity, flatness, divergence, and normal vectors from the point cloud, and applied a greedy graph cut algorithm to generate superpoints. Xiao et al. [39] defined the superpoint segmentation problem as an energy minimization problem and proposed a merging-exchange optimization framework to generate supervoxels. These methods often require complex and computationally expensive preprocessing steps, and their performance is limited by the insufficient expressiveness of handcrafted features. Recent deep learning methods overcome these limitations.

Supervised Superpoint (SSP) method [18] pioneered deep network-based point cloud embedding, although it



(a) Point Feature Extractor (b) Hierarchical Segment Initialization (c) Adaptive Superpoint Update

Figure 2. The architecture includes three stages. First, point embeddings are extracted using a U-Net backbone with sparse convolutions in shallow layers and a Mamba bottleneck enhanced by Conditional Position Encoding (CPE). Second, the point cloud is serialized via a Hilbert curve to initialize segments for superpoint aggregation, with granularity at each level controlled by the hyperparameter m. Finally, points are reassigned across segments to group semantically similar points into coherent superpoints, completing the clustering process.

still depended on optimization techniques from [12]. SPNet [15] further advanced the field by introducing learnable point-to-superpoint associations using FPS for center selection, enabling joint optimization in both coordinate and feature spaces. Furthermore, SuperLiDAR [16] improved efficiency by combining a BFS-based grouping algorithm with a local discriminative loss to generate compact superpoints. However, this method results in high computational complexity in spatial query operations during superpoint generation, especially for dense point cloud oversegmentation.

2.3. Serialization-based Method

Recent works [5, 23, 34, 38] have introduced serialization-based approaches that differ from traditional point cloud processing by transforming unstructured point clouds into ordered sequences to preserve spatial proximity. For example, OctFormer [34] uses octree-based ordering similar to z-ordering, offering scalability but limited by octree constraints. FlatFormer [23] employs window-based sorting to group point pillars, though its scalability in receptive field is limited, making it suitable for pillar-based 3D object detection. Building on these methods, PTv3 [38] extends attention scales by replacing KNN-based query with serialized neighborhoods, enhancing efficiency and scalability.

3. Method

Our oversegmentation network comprises three key steps: point embedding extraction (Sec. 3.1), hierarchical segment initialization (Sec. 3.2), and adaptive superpoint update (Sec. 3.3). The overall process is shown in Fig. 2.

3.1. Backbone

Optimization-based oversegmentation methods are constrained by the quality of handcrafted features, while learning based approaches similarly encounter limitations in deep feature representation.

Sparse 3D U-Net. Let the input point cloud be $\mathcal{P} \in \mathbb{R}^{N\times(C+3)}$, where each of the N points is represented by its features (*e.g.* normal vectors, RGB) and spatial coordinates (x, y, z). Following [10], we first voxelize $\mathcal{P}$ using the operator $\mathcal{V}$, and then process the discretized data with a U-Net architecture based on sparse convolutions (SpConv). In the shallow part of the U-Net [30], SpConv layers capture fine-grained local geometric information. These operations produce feature representations that maintain high spatial resolution and extract detailed geometric structures crucial for accurate object boundary delineation.

Bottleneck Reinforcement. Pure SpConv restricts oversegmentation effectiveness due to their inherently limited receptive field. We enhance our U-Net backbone by integrating Mamba [11, 45] at the bottleneck of the U-Net, as shown in Fig. 2 (a). After the shallow SpConv layers have abstracted local features, the Mamba module is applied to perform global context interaction. Mamba, with its linear-complexity global modeling capability, compensates for the local inductive bias of convolution and promotes semantic consistency across superpoints.

3.2. Hierarchical Segment Initialization

Methodology Justification. To efficiently partition point clouds into multiple segments, we propose a serialization-

based partitioning method that utilizes the spatial locality preservation property of the Hilbert curve [38, 41] to divide the same point cloud into hierarchical structures of different granularities. Specifically, the Hilbert curve maps adjacent point clusters in 3D Euclidean space to contiguous positions in a 1D sequence, thereby simplifying the spatial proximity-based 3D partitioning task into an iterative partitioning of a 1D sequence. This transformation enables us to efficiently divide point clouds into compact segments in 3D space, significantly enhancing partitioning efficiency. This enables (1) efficient 1D clustering that avoids expensive 3D queries and clustering; and (2) a simplified oversegmentation process that decouples spatial locality from semantic similarity by performing only semantic comparisons within already spatially adjacent segments to complete clustering.

Partitioning Strategy. Due to the variable receptive field of differently sized superpoints, smaller superpoints preserve more detailed local geometric information, whereas larger superpoints capture more comprehensive global semantic context. Therefore, we propose to construct a hierarchical partitioning structure to fully utilize multi-scale information. Given an input point cloud $\mathcal{P}$ and its corresponding features $F = \{f_i\}_{i=1}^{N}$, where $f_i \in \mathbb{R}^d$ is generated by our backbone, we construct a hierarchical partitioning structure $\mathcal{H} = \{L_k\}_{k=1}^{K}$. Each layer L_k contains the complete serialized point cloud and is recursively quaternary partitioned into a set of segments $L_k = \{\mathcal{S}_i^k\}_{i=0}^{M_k-1}$, where $\mathcal{S}_i^k$ represents the i-th segment, as shown in Fig. 2 (b). Specifically, the 3D point cloud is first serialized into a 1D sequence via Hilbert space-filling curve mapping [38], denoted as $\mathcal{P}_{seq} \in \mathbb{R}^{N\times d}$. The sequence is then padded to $\mathcal{P}_{pad} \in \mathbb{R}^{N_{pad}\times d}$ where $N_{pad} = \lceil N/M_K \rceil \cdot M_K$, ensuring divisibility by the finest segment count M_K. The serialized point cloud $\mathcal{P}_{pad}$ is recursively partitioned using 1D quaternary splitting, such that at each layer L_k, it is uniformly divided into $M_k = 2^{m-2k+2}$ segments. For each layer L_k, $\mathcal{S}_i^k$ can be represented as:

$$\mathcal{S}_i^k = \{\mathcal{P}_{pad}[j] \mid j \in [i \cdot l_k, (i+1)\cdot l_k - 1]\}, \quad (1)$$

where $l_k = N_{pad}/M_k$ represents the segment length for each segment in the k-th layer. Recursive partitioning establishes fixed parent-child indexing between layers. For any inter-layer gap $l \geq 1$, the hierarchical inclusion relationship is defined as:

$$\mathcal{S}_i^k = \bigcup\nolimits_{j\in[i\cdot 4^l,\ (i+1)\cdot 4^l-1]} \mathcal{S}_j^{k+l}. \quad (2)$$

At the finest layer L_K, segment features are generated through permutation-invariant aggregation of constituent points:

$$f_{S_i^K} = \mathcal{A}\big(f_j \mid p_j \in \mathcal{S}_i^K\big) \in \mathbb{R}^C, \quad (3)$$

where $\mathcal{A} : \mathbb{R}^{n_i\times C} \to \mathbb{R}^C$ denotes a permutation-invariant operator (*e.g.* max-pooling) with $n_i = |\mathcal{S}_i^K| = N_{pad}/M_K$.

3.3. Adaptive Superpoint Update

Given the structure $\mathcal{H}$, the goal is to refine the partition $L_k = \{S_1^k, \ldots, S_{M_k}^k\}$ with features $f_{S_i^k} \in \mathbb{R}^d$ into semantically consistent superpoints $L'_k = \{S_1^{'k}, \ldots, S_{M_k}^{'k}\}$ with precise geometric boundaries and updated features $f'_{S_i^k} \in \mathbb{R}^d$. This is achieved by dynamically allocating raw points within coarse-grained segments to their appropriate sub-segments based on semantic similarity, a process that redistributes raw points within segments to fit object boundaries. This process can be transformed into a boundary fitting problem between adjacent segments in 1D sequence point clouds, as shown in Fig. 2 (c).

Feature Similarity based Affiliation. To establish the affiliation relationship between raw points within each segment and nearby segments, we consider two adjacent levels in $\mathcal{H}$: the finest $L_k = \{S_1^k, S_2^k, \ldots, S_{M_k}^k\}$ and its coarser predecessor $L_{k-1} = \{S_1^{k-1}, S_2^{k-1}, \ldots, S_{M_{k-1}}^{k-1}\}$ (we only use $l = 1$ defined in Eq. (2)). The hierarchical relationship between these levels is given by (1) $M_{k-1} = \frac{M_k}{4}$, meaning each coarser level has one-fourth the segments of the finer level and (2) Each segment S_i^{k-1} at level L_{k-1} contains four consecutive segments from level L_k: $S_i^{k-1} = S_{4i}^k \cup S_{4i+1}^k \cup S_{4i+2}^k \cup S_{4i+3}^k$.

For each raw point $p_i \in S_i^{k-1}$, we identify the most similar initialized segment feature from its four containing segments S_{4i}^k, S_{4i+1}^k, S_{4i+2}^k, and S_{4i+3}^k. Meanwhile, to expand the receptive field of raw points relative to their potential superpoints, we incorporate segment features from eight sub-segments at L_k belonging to the left and right adjacent segments S_{i-1}^{k-1} and S_{i+1}^{k-1}, forming a local feature set $\{f_{S_{4i+j}^k} \mid j \in \{-4, -3, \ldots, 7\}\}$, resulting in a set of 12 initial segment features. For each segment S_i^{k-1} with $N_{k-1} = 4N_{pad}/M_k$ points, we compute a similarity matrix $\mathbf{A}_i \in \mathbb{R}^{N_{k-1}\times 12}$. Each entry $a_{j,t}$ measures the similarity between the j-th point and the t-th segment feature in the above set of segment features:

$$a_{j,t} = \frac{f_{p_j} \cdot f_{S_t^k}}{\|f_{p_j}\| \|f_{S_t^k}\|}, \quad j \in [1, N_{k-1}], \quad t \in [1, \tau], \quad (4)$$

This process generates similarity matrices $\{\mathbf{A}_i\}_{i=1}^{M_{k-1}}$ for the entire level L_{k-1}. Point-to-superpoint assignments are determined by selecting the superpoint with the highest similarity for each point, which is then mapped to global superpoint indices $\phi(p_j) \in [1, M_k]$. The resulting assignment index set $Assi = \{\phi(p_j)\}_{j=1}^{n}$ provides a key reference for updating superpoint features in subsequent steps.

Cross-Attention based Update. Cross-Attention [4] enables dynamic and adaptive relationships between superpoints and their constituent points. Through feature similarity based affiliation, precise similarity relationships are established, guiding accurate superpoint feature updates. For

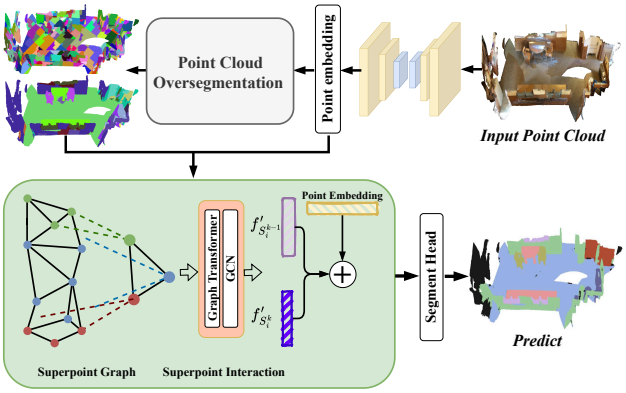


Figure 3. The end-to-end semantic segmentation pipeline constructs a graph over superpoints, captures global context using graph transformers, and integrates point-level embeddings for final semantic prediction, where $\oplus$ denotes simple feature addition.

the assignment indices $Assi = \{\phi(p_j)\}_{j=1}^{n}$, the inverse mapping $\phi^{-1}(i) = \{p_j \mid \phi(p_j) = i\}$ represents the original points contained within superpoint $S_i^{'k}$. Based on $Assi$, each superpoint's feature is updated solely from its member points. Using superpoint feature $f'_{S_i^k}$ as the Query and its member point features $\{f_{p_j} \mid p_j \in \phi^{-1}(i)\}$ as Keys and Values, the update formula is:

$$f'_{S_i^k} = \text{CrossAttn}\left(\underbrace{f_{S_i^k}}_{\text{Query}}, \underbrace{\{f_{p_j} \mid p_j \in \phi^{-1}(i)\}}_{\text{Key/Value}}\right), \quad (5)$$

Attention weights are computed through scaled dot product, allowing each superpoint to focus on the most informative points within its local neighborhood:

$$\text{CrossAttn}(f_{S_i^k}, \{f_{p_j}\}) = \text{Softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^\top}{\sqrt{d}}\right)\mathbf{V}_i, \quad (6)$$

where $\mathbf{Q}_i = f_{S_i^k} W_Q \in \mathbb{R}^{1\times d}$, $\mathbf{K}_i = \text{Concat}(\{f_{p_j} W_K \mid p_j \in \phi^{-1}(i)\}) \in \mathbb{R}^{n_i^k \times d}$, $\mathbf{V}_i = \text{Concat}(\{f_{p_j} W_V \mid p_j \in \phi^{-1}(i)\}) \in \mathbb{R}^{n_i^k \times d}$, $n_i^k = |\phi^{-1}(i)|$ is the number of points contained in each superpoint, and $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are learnable projection matrices. For coarse-grained superpoints, we apply the same update logic, with interactions between fine-grained features and their coarse-grained initializations.

3.4. Superpoint Interaction

The superpoint features $f'_{S_i^k}$ encode geometric-semantic information for locally homogeneous points. To leverage these features for semantic segmentation, inspired by SPT [29], we integrate a superpoint interaction module with our oversegmentation network, forming an end-to-end framework, as shown in Fig. 3. In our hierarchical design, fine-grained superpoints capture local details while coarse grained ones represent object-level semantics. We construct multi-level superpoint graphs and employ a cascaded Transformer-GCN [42] to facilitate cross-superpoint interactions and global context modeling. In the Graph Transformer, a superpoint graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is built by linking each superpoint with its k nearest neighbors in feature space, where $\mathcal{V} = \{f'_{S_i^k}\}_{i=1}^{M_k}$. Edge features are defined via relative position encoding as $\mathbf{e_{ij}} = g(p_j - p_i)$, with p_i and p_j representing node coordinates and g being a linear mapping. A subsequent GCN processes this graph to enhance local feature interactions through message passing among adjacent superpoints. Finally, we fuse backbone point features with their corresponding multi-level superpoint before feeding them into the segmentation head.

3.5. Superpoint Aggregation Loss

We propose a superpoint aggregation loss to optimize superpoint generation, inspired by LMNN [36]. This comprehensive loss consists of three components. First, a compactness loss ℓ_{compact} enforces intra-superpoint feature cohesion by minimizing point-to-superpoint distances. Second, a distinction loss ℓ_{dist} encourages inter-class separability by maximizing the distances between superpoints of different classes, where each superpoint's label is determined by majority voting among its points. Third, a purity loss ℓ_{purity} maintains label consistency within each superpoint by penalizing high entropy in its class distribution. The overall loss function is defined as:

$$\begin{aligned}\ell_{sp} = &\frac{1}{N_s}\sum_{i=1}^{N_s}\frac{\sum_{p\in S_i}\|f_i - f'_{S_i^k}\|^2}{|S_i|} + \\ &\frac{1}{N_c}\sum_{i,j\in c, i\neq j}\max(0, \delta_{\text{dist}} - \|f'_{S_i^k} - f'_{S_j^k}\|)^2 + \\ &\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{k=1}^{K} p_{i,k}\log(p_{i,k}),\end{aligned} \quad (7)$$

where f_i denotes the feature of point p, $f'_{S_i^k}$ represents the feature of superpoint $S_i^{'k}$, δ_{dist} is a predefined separation margin, $p_{i,k}$ is the probability of superpoint i belonging to class k, N_s is the number of superpoints, and N_c is the number of superpoint pairs involved in ℓ_{dist}.

4. Experiments

4.1. Datasets and Metrics

Datasets. We evaluate SPCNet on four datasets with diverse characteristics. **S3DIS** [1] contains $274M$ points from six office areas, with Area 5 as the test set. **ScanNet** [9] includes 1,513 indoor scans from 707 scenes with

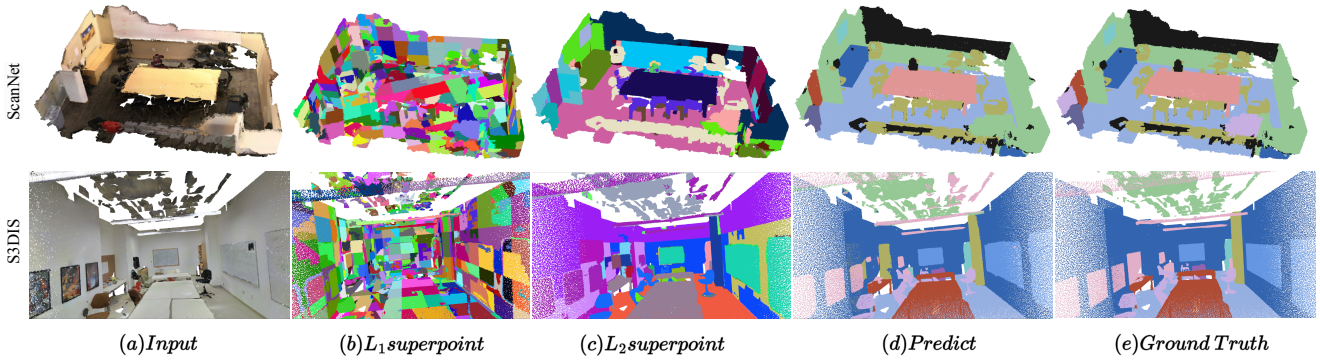


Figure 4. The visualization of superpoints generated on the ScanNet (first row) and S3DIS (second row) datasets shows a two-level hierarchy: L_1 with 1024 superpoints and L_2 with 256 superpoints. Finer L_1 superpoints merge based on semantic similarity to form larger L_2 superpoints. For example, in S3DIS, L_1 superpoints of table parts merge into a single table superpoint, and L_1 superpoints of whiteboard sections form a complete whiteboard superpoint. Our superpoints are regular and orderly, with nearly rectangular shapes in semantically consistent areas (*e.g.* floors, tables), maintaining continuity, precise object boundaries, and demonstrating superior regularity, compactness, and purity.

20 categories, averaging 148k points/scan. **nuScenes** [3] comprises 1,000 scenes from 32-beam LiDAR, split 750/150/150 for train/val/test, with 17 categories. **SemanticKITTI** [2] is an outdoor self-driving benchmark with 22 sequences and 20 categories, using sequences 00-10 (densely annotated) for training and 11-22 for testing.

Evaluation Metrics. Following SPNet [15] and SSP [18], we use Oracle Overall Accuracy (OOA), Boundary Recall (BR), Boundary Precision (BP), and F1 score to evaluate superpoints. BR and BP assess boundary quality, while OOA represents the upper bound of semantic segmentation accuracy using superpoints. The F1 score balances recall and precision, defined as $F1 = \frac{2 \cdot BP \cdot BR}{BP+BR}$. For semantic segmentation in indoor and outdoor scenarios, we adopt mean intersection over union (mIoU).

4.2. Model Configuration

We maintain a consistent architecture across all experiments with minimal dataset-specific adjustments. Our backbone features a four-stage symmetric encoder-decoder structure, each with a single block depth. The encoder has embedding dimensions [32, 64, 128, 256] and the decoder [128, 64, 64], with Mamba blocks using conditional positional encoding at the 256 stage. We implement a three-level hierarchy (m=8, K=2, l=1), organizing points into n, 1024, 256 segments. Training is performed on dual NVIDIA RTX 4090 GPUs with the Adam optimizer.

4.3. Point Cloud Oversegmentation

Quantitative Results. As shown in Tab. 1, we evaluate SPCNet on three datasets, reserving ScanNet for ablation studies, ensuring fair comparison with consistent superpoint counts. For example, in S3DIS Area 5, methods like VCCS, Lin et al., SPG, SSP, and SPNet generate around 1050 superpoints, matching our 1024 Hilbert partitions. Our deep learning-based approach outperforms traditional methods, benefiting from the specialized backbone network's ability to extract rich geometric features that capture complex local structures. Unlike methods such as SPNet [15], which rely on FPS+KNN for presampling and soft assignment map-

Method	BR	BP	F1	OOA
S3DIS Area 5 [1]				
VCCS [26]	62.06	10.86	18.48	95.22
SPG [19]	52.06	13.11	20.94	95.84
SSP [18]	80.73	13.02	22.42	**97.04**
SPNet [15]	84.75	13.14	22.65	96.50
SPCNet(Ours)	**89.50**	**26.78**	**40.50**	96.81
nuScenes [3]				
SPG [19]	28.20	17.26	21.41	89.22
SSP [18]	22.04	15.63	18.28	92.01
SPNet [15]	68.92	18.34	28.97	87.67
SuperLiDAR [16]	74.72	25.12	37.59	96.31
SPCNet(Ours)	**88.77**	**32.82**	**47.52**	**96.33**
SemanticKITTI [2]				
SPG [19]	25.64	15.82	19.56	86.86
SSP [18]	18.74	10.67	13.59	92.27
SPNet [15]	56.52	14.78	23.43	92.24
SuperLiDAR [16]	65.52	20.52	31.25	96.21
SPCNet(Ours)	**73.18**	**30.15**	**42.57**	**96.84**

Table 1. Comparison results of generated superpoints on the S3DIS, nuScenes, and SemanticKITTI datasets.

ping without limiting the scope of raw points being clustered, resulting in blurry boundaries, our approach leverages the Hilbert curve's locality, limiting the range of raw point clustering (preventing clustering of semantically similar but spatially distant points), and finally refining superpoint through the Cross-Attention, minimizing noise and maintaining precise, simple boundaries, as shown in Fig. 5 (d) and (e). Visualization results in Fig. 4 show that our method tends to aggregate superpoints within object interiors while maintaining distinct object boundaries, especially in challenging cases like wall-whiteboard interfaces, resulting in higher BR and BP. Additionally, our method excels at merging semantically continuous regions (*e.g.* floors, tabletops), leading to fewer incorrect superpoint boundaries and substantially higher BP metrics. Fig. 6 shows the performance of different methods on SemanticKITTI.

Method	Device	Inference Time (ms) data proc. / backbone + SP ini. + SP gen.
SPG [19]	CPU	13152
SSP [18]	RTX 3090	16115 (11812 + 0 + 13426)
SPNet [15]	RTX 3090	12182 (11812 + 0 + 1154)
SuperLiDAR [16]	RTX 3090	72(0 + 0 + 72)
SPCNet (ours)	**RTX 4090**, RTX 3060	**12 (6 + 0.4 + 6)**, 61

Table 2. The inference time of different oversegmentation methods on SemanticKITTI. The inference time consists of data preprocessing/backbone (red/blue), superpoint initialization (green), and superpoint generation (purple).

Time Costs. We evaluated oversegmentation network efficiency by measuring inference time per scan on SemanticKITTI validation (batch size=1) in Tab. 2. For optimization based SPG [19], we followed SuperLiDAR's [16] configuration using Core i5 CPU. For learning-based methods (SSP [18], SPNet [15], SuperLiDAR), we used SuperLiDAR's results (RTX 3090). Our SPCNet was benchmarked on RTX 4090 and 3060 GPUs. Our method achieved total inference times of 12 ms (RTX 4090) and 61 ms (RTX 3060), achieving $100\times$ speedup versus SPG/SSP/SPNet, and $5\times$ versus SuperLiDAR. Even on RTX3060, our inference is 10ms faster than SuperLiDAR, indicating at least $3\times$ efficiency accounting for hardware differences. This stems from: (1) efficient superpoint initialization (0.4 ms) that compresses FPS+KNN sampling time in SPNet; (2) the application of a localized attention mechanism within the superpoint refinement (6 ms). Our backbone architecture allows further acceleration using lightweight alternatives like PointNet[27]. Traditional methods face key bottlenecks: SPG/SSP rely on optimization-based generation (*e.g.* graph cuts), SSP/SPNet suffer from redundant processing due to mixed handcrafted/learned features, and SuperLiDAR's BFS-based grouping has complexity dependent on scene density, potentially reaching $O(n^2)$ in dense scenes, versus our serialization based approach ($O(n)$).

Ablation Studies. Mamba mitigates the local inductive bias

(a) 3D-UNet only with k=2 (b) Bottleneck reinforcement with k=2 (d) PointNet with k=2 (e) SPNet

Figure 5. Oversegmentation comparisons. (a) Sparse 3D U-Net with neighbor expansion of 2. (b) Adding the Mamba module to the bottleneck. (c) Replacing the backbone with a simplified PointNet, as in SPNet. (d) SPNet results. Notably, while (b) configuration does not fully aggregate semantically consistent objects into complete superpoints, it notably enhances consistency. SPCNet also avoids clustering semantically similar but spatially distant points, ensuring clearer superpoint boundaries.

Backbone Setting	BR(%)	BP(%)	OOA(%)	mIou(%)
3D U-Net	76.33 ↓7.90	28.45 ↓1.70	90.93 ↓1.19	74.10 ↓2.86
+ Mamba fully	78.37 ↓5.86	29.13 ↓1.02	91.24 ↓0.88	76.18 ↓0.78
+ Mamba bottneck (defalut)	**84.23**	**30.15**	**92.12**	**76.96**

Table 3. Performance comparison of different backbone architectures on the ScanNet validation set.

Neighbor Scope	BR(%)	BP(%)	OOA(%)
No expansion	79.86 ↓4.37	28.57 ↓1.58	91.22 ↓0.90
2 expansion (default)	**84.23**	**30.15**	**92.12**
4 expansion	83.67 ↓0.56	29.85 ↓0.30	91.24 ↓0.88
6 expansion	79.93 ↓4.30	28.65 ↓1.50	89.93 ↓2.19

Table 4. The effect of expanding neighbor segment range on superpoint performance on the ScanNet validation set.

of convolutions in oversegmentation, improving superpoint continuity and semantic consistency. As shown in Fig. 5 (a) and (b), this bias causes superpoints in the same semantic region to align with boundaries but fragment due to internal inconsistency. Thus, we retain convolutions in U-Net's shallow layers for local geometric cues to ensure boundary precision, and introduce Mamba at the bottleneck to capture long range semantics. Tables 3 and 4 analyze backbone design and neighbor expansion effects on superpoint quality. We evaluated three backbone configurations: (1) baseline 3D sparse U-Net only, (2) fully Mamba-based U-Net, and (3) our default hybrid design with Mamba modules in the bottleneck layers. Results demonstrate that the baseline 3D sparse U-Net reduces BR, BP, and OOA by 7.90%, 1.70%, and 1.19% respectively compared to our default. Similarly, the fully Mamba-based U-Net underperforms, as Mamba's global context modeling in shallow networks with low-dimensional features smooths out high-frequency boundary information. Our hybrid "shallow convolution + deep Mamba" architecture achieves optimal per-

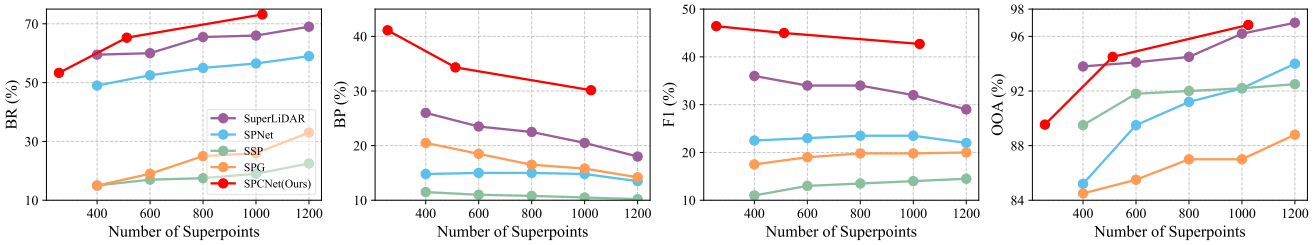


Figure 6. Performance of different methods on the SemanticKITTI validation set under varying numbers of superpoints. It is worth noting that our method can achieve better results than other methods with fewer superpoints (256, 512, and 1024).

Experiment	ScanNet [9]		S3DIS [1]	nuScenes [3]		Sem.KITTI [2]	
Best Model	**Val**	**Test**	**Area5**	**Val**	**Test**	**Val**	**Test**
†SPG [19]	-	-	58.0	-	-	-	-
MinkUNet [8]	72.2	73.6	65.4	73.3	-	63.8	-
†SPG + SSP [19]	-	-	61.7	-	-	-	-
ST [17]	74.3	73.7	72.0	-	-	-	-
†SPT [29]	-	-	68.9	-	-	-	-
PointNeXt [28]	71.5	71.2	70.5	-	-	-	-
OctFormer [34]	75.7	76.6	-	76.1	77.2	64.3	67.8
Swin3D [40]	76.4	-	72.5	-	-	-	-
AF2S3Net [7]	-	-	-	62.2	78.0	74.2	70.8
†SuperLiDAR [16]	-	-	-	-	78.5	-	69.6
PTv2 [37]	75.4	74.2	71.6	80.2	82.6	70.3	72.6
PTv3 [38]	77.5	77.9	**73.4**	**80.4**	**82.7**	**70.8**	**74.2**
Backbone	76.9	-	69.3	75.4	-	67.6	-
†SPCNet(Ours)	**77.8**	**78.0**	72.1	77.6	80.2	69.5	71.9

Table 5. Semantic segmentation results on the ScanNet, S3DIS, nuScenes, and SemanticKITTI datasets. †Superpoint-based.

formance by leveraging convolutions to preserve local details in shallow layers while Mamba modules handle global context modeling in deeper layers.

In the neighborhood expansion experiments (analogous to k-parameter tuning in KNN), we observed that moderately expanding the candidate superpoint pool of original points during the feature similarity based affiliation process significantly improves sampling accuracy. The model achieves optimal performance when K=2; excessive expansion (K=4, 6) leads to performance degradation. This may be because larger receptive fields introduce noise, and the hard assignment approach introduces more uncertainty as the number of candidate targets increases, causing early assignment errors to propagate to later processing stages.

4.4. Semantic Segmentation

Tab. 5 evaluates our approach on four indoor and outdoor datasets, comparing it with point-based and superpoint-based methods. Our optimized oversegmentation strategy shows significant gains: on S3DIS Area 5, we achieve a 3.2% mIoU improvement over SPT. Compared to state-of-the-art point-based methods, our approach remains competitive, outperforming PTv3 by 0.3% on ScanNet (77.8%) and PTv2 by 0.5% on S3DIS Area 5. Additionally, we achieve 71.9% and 80.2% on Sem.KITTI and nuScenes, improving SuperLiDAR by 1.3% and 1.7%, respectively. These results confirm that our superpoints capture local and global semantic information effectively, while the simple backbone, enhanced by superpoint features, matches the performance of more complex point-based methods, validating the superior quality of the generated superpoints.

Setting	Components	Model Size (m) Backbone + SP gen. + SP int.	mIoU(%)	Δ(%)
backbone (defalut)	-	18.8	76.96	-
Hierarchical Enhancement				
1 partition level	+ Level1 Superpoint	0.24 (0.218 + 0.022)	77.50	+0.54
2 partition level	+ Level2 Superpoint	0.02 (0.014 + 0.006)	77.80	+0.30

Table 6. Semantic segmentation improvement with hierarchical superpoint features in ScanNet validation set. SP gen. means superpoint generation; SP int. means superpoint interaction.

As shown in Tab. 6, integrating hierarchical superpoint features with per-point features extracted by the backbone improved segmentation. First-level superpoints aggregated local semantics, boosting intra-class consistency with a 0.54% mIoU gain. Second-level features captured long-range dependencies, adding another 0.30% improvement. These hierarchical superpoints acted as adaptive receptive fields, enhancing performance with only $0.24M$ parameters.

5. Conclusion

In this paper, we propose a novel serialization based oversegmentation method to address the challenge of efficiently grouping semantically consistent points into superpoints while maintaining spatial proximity. SPCNet achieves state-of-the-art performance in point cloud oversegmentation while being 3× faster than existing methods in inference speed. Extensive experimental results demonstrate the effectiveness and efficiency of SPCNet. Additionally, SPCNet can be flexibly applied to semantic segmentation tasks, effectively improving the accuracy of semantic segmentation.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (Grant No. 42201475).

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 2, 5, 6, 8

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 2, 6, 8

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 6, 8

[4] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 4

[5] Wanli Chen, Xinge Zhu, Guojin Chen, and Bei Yu. Efficient point cloud analysis using hilbert curve. In *European Conference on Computer Vision*, pages 730–747. Springer, 2022. 3

[6] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[7] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12547–12556, 2021. 8

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019. 2, 8

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 5, 8

[10] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 2, 3

[11] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 3

[12] Stéphane Guinard and Loic Landrieu. Weakly supervised segmentation-aided classification of urban scenes from 3d lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:151–157, 2017. 1, 2, 3

[13] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021. 2

[14] David Hilbert and David Hilbert. Über die stetige abbildung einer linie auf ein flächenstück. *Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes: Nebst Einer Lebensgeschichte*, pages 1–2, 1935. 2

[15] Le Hui, Jia Yuan, Mingmei Cheng, Jin Xie, Xiaoya Zhang, and Jian Yang. Superpoint network for point cloud oversegmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5510–5519, 2021. 1, 3, 6, 7

[16] Le Hui, Linghua Tang, Yuchao Dai, Jin Xie, and Jian Yang. Efficient lidar point cloud oversegmentation network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18003–18012, 2023. 2, 3, 6, 7, 8

[17] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8500–8509, 2022. 8

[18] Loic Landrieu and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7440–7449, 2019. 1, 2, 6, 7

[19] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. 1, 6, 7, 8

[20] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 2

[21] Bo Li, T Zhang, and T Xia. Vehicle detection from 3d lidar using fully convolutional network. arxiv 2016. *arXiv preprint arXiv:1608.07916*. 2

[22] Yangbin Lin, Cheng Wang, Dawei Zhai, Wei Li, and Jonathan Li. Toward better boundary preserved supervoxel segmentation for 3d point clouds. *ISPRS journal of photogrammetry and remote sensing*, 143:39–47, 2018. 1, 2

[23] Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. Flatformer: Flattened window attention for efficient point cloud transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1200–1211, 2023. 3

[24] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022. 2

[25] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition.

In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015. 2

[26] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2027–2034, 2013. 1, 2, 6

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 7

[28] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204, 2022. 2, 8

[29] Damien Robert, Hugo Raguet, and Loic Landrieu. Efficient 3d semantic segmentation with superpoint transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17195–17204, 2023. 2, 5, 8

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3

[31] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017. 2

[32] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 2

[33] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 2

[34] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023. 3, 8

[35] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11, 2017. 2

[36] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, 18, 2005. 5

[37] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 2, 8

[38] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 2, 3, 4, 8

[39] Yanyang Xiao, Zhonggui Chen, Zhengtao Lin, Juan Cao, Yongjie Jessica Zhang, Yangbin Lin, and Cheng Wang. Merge-swap optimization framework for supervoxel generation from three-dimensional point clouds. *Remote Sensing*, 12(3):473, 2020. 2

[40] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*, 2023. 2, 8

[41] Guowen Zhang, Lue Fan, Chenhang He, Zhen Lei, Zhaoxiang Zhang, and Lei Zhang. Voxel mamba: Group-free state space models for point cloud based 3d object detection. *arXiv preprint arXiv:2406.10700*, 2024. 4

[42] Peiyan Zhang, Yuchen Yan, Xi Zhang, Chaozhuo Li, Senzhang Wang, Feiran Huang, and Sunghun Kim. Transgnn: Harnessing the collaborative power of transformers and graph neural networks for recommender systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1285–1295, 2024. 2, 5

[43] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5565–5573, 2019. 2

[44] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2

[45] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 3