VMNet: Voxel-Mesh Network for Geodesic-Aware 3D Semantic Segmentation

Zeyu HU*[1], Xuyang Bai[1], Jiaxiang Shang[1], Runze Zhang[2], Jiayu Dong[2], Xin Wang[2], Guangyuan Sun[2], Hongbo Fu†[3], and Chiew-Lan Tai[1]

[1]Hong Kong University of Science and Technology
[2]Lightspeed & Quantum Studios, Tencent
[3]City University of Hong Kong

Abstract

In recent years, sparse voxel-based methods have become the state-of-the-arts for 3D semantic segmentation of indoor scenes, thanks to the powerful 3D CNNs. Nevertheless, being oblivious to the underlying geometry, voxel-based methods suffer from ambiguous features on spatially close objects and struggle with handling complex and irregular geometries due to the lack of geodesic information. In view of this, we present Voxel-Mesh Network (VMNet), a novel 3D deep architecture that operates on the voxel and mesh representations leveraging both the Euclidean and geodesic information. Intuitively, the Euclidean information extracted from voxels can offer contextual cues representing interactions between nearby objects, while the geodesic information extracted from meshes can help separate objects that are spatially close but have disconnected surfaces. To incorporate such information from the two domains, we design an intra-domain attentive module for effective feature aggregation and an inter-domain attentive module for adaptive feature fusion. Experimental results validate the effectiveness of VMNet: specifically, on the challenging ScanNet dataset for large-scale segmentation of indoor scenes, it outperforms the state-of-the-art SparseConvNet and MinkowskiNet (74.6% vs 72.5% and 73.6% in mIoU) with a simpler network structure (17M vs 30M and 38M parameters). Code release: https://github.com/hzykent/VMNet

1. Introduction

Thanks to the tremendous progress of RGB-D scanning methods in recent years [63, 27, 10], reliable tracking and reconstruction of 3D surfaces using hand-held, consumer-grade devices have become possible. Using these methods, large-scale 3D datasets with reconstructed surfaces and semantic annotations are now available [8, 4]. Nevertheless, compared to 3D surface reconstruction, 3D scene understanding, i.e., understanding the semantics of reconstructed scenes, is still a relatively open research problem.

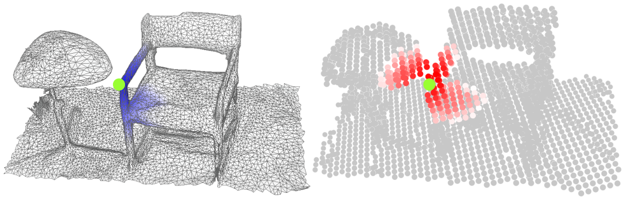

Figure 1. **Illustration of geodesic information loss caused by voxelization.** Considering the green point at the arm of a chair, on an input 3D mesh surface (**Left**), its geodesic neighbors (blue) can be easily collected, and the points of different objects are naturally separated. After voxelization (**Right**), geodesic information is discarded and only Euclidean neighbors (red) that are agnostic to the underlying surface can be extracted. The scan section is taken from the ScanNet dataset [8].

Inspired by the success of 2D CNN in image semantic segmentation [5, 36], researchers have paid much attention to the straightforward extension of this idea to 3D, by performing volumetric convolution on regular grids [39, 66, 44]. Specifically, surface reconstructions are first projected to a discrete 3D grid representation, and then 3D convolutional filters are applied to extract features by sliding kernels over neighboring grid voxels [54, 62, 72]. Such features can be smoothly propagated in the Euclidean domain to accumulate strong contextual information. Unfortunately, dense voxel-based methods require intensive computational power and are thus limited to low-resolution cases [35]. To process large-scale data, sparse voxel convolutions [17, 7] have been proposed to lower the computational requirement by ignoring inactive voxels. Benefiting from the efficient sparse voxel convolutions, complex networks have been built, achieving leading results on several 3D semantic segmentation benchmarks [8, 4] and outperforming other methods by large margins.

Despite the remarkable achievements, voxel-based methods are not perfect. One of their major limitations is the geodesic information loss caused by the voxelization process (see Fig. 1). Recent public datasets like ScanNet [8] provide 3D scene reconstructions in the form of high-quality triangular meshes, in which the surface information is naturally encoded. On these meshes, vertices belonging to different objects are well separated, and geodesic

*intern at Tencent Lightspeed & Quantum Studios
†corresponding author

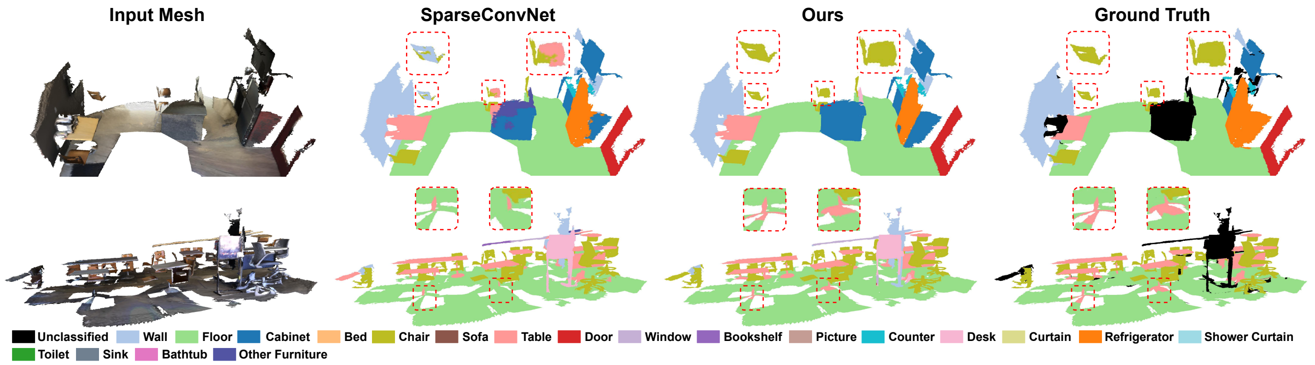

Figure 2. **Limitations of voxel-based methods.** (**Upper**) Some points of "chairs" are mistakenly classified into nearby classes in the Euclidean space by SparseConvNet [17] since the convolutional filters produce ambiguous features for spatially close objects. (**Lower**) On areas with complex and irregular geometries (*e.g.*, the base parts of "tables"), SparseConvNet fails to predict correct results due to the lack of geodesic information about shape surfaces.

features can be easily aggregated through edge connectivities. However, the voxelization process omits all mesh edges and only retains Euclidean positions of mesh vertices. Consequently, convolutional filters operating on voxels are agnostic to the underlying surfaces and, therefore, result in two problems. First, these filters generate similar features for voxels that are close in the Euclidean domain, even though these voxels may belong to different objects and are distant in the geodesic domain. As shown in the top example of Fig. 2, these ambiguous features produce suboptimal predictions for objects that are spatially close. Second, without the geodesic information about shape surfaces, these Euclidean convolutions may struggle with learning specific object shapes. As shown in the lower example of Fig. 2, this property is problematic for segmentation on areas with complex and irregular geometries.

We have discussed the advantages of voxel-based methods on contextual learning and their problems on geodesic information loss. It is appealing to design a method resolving the problems while retaining these advantages by leveraging both the Euclidean and geodesic information. A possible solution is to take voxels and the original meshes as the sources for the Euclidean and geodesic information, respectively. It is therefore natural to ask how these two representations can be combined in a common architecture.

To address this question, we propose the Voxel-Mesh network (VMNet), a novel deep hierarchical architecture for geodesic-aware 3D semantic segmentation. Starting from a mesh representation, to extract informative contextual features in the Euclidean domain, we first voxelize the input mesh and apply sparse voxel convolutions. Next, to incorporate the geodesic information, the extracted contextual features are projected from the Euclidean domain to the geodesic domain, specifically, from voxels to mesh vertices. These projected features are further fused and aggregated to combine both the Euclidean and geodesic information.

In order to build such a deep architecture that is capable of effectively learning useful features incorporating information from the two domains, it is critical to design proper ways to aggregate intra-domain features and to fuse inter-domain features. In view of the great success of self-attention operators for feature processing [59, 41, 34], we therefore present two key components of VMNet: Intra-domain Attentive Aggregation Module and Inter-domain Attentive Fusion Module. The former aims to aggregate the projected features on the original meshes to incorporate the geodesic information and the latter focuses on the effective fusion of features from the two domains.

We conduct extensive experiments to demonstrate the effectiveness of our method on the popular ScanNet v2 benchmark [8] and the recent Matterport3D benchmark [4]. VMNet outperforms existing sparse voxel-based methods SparseConvNet [17] and MinkowskiNet [7] (74.6% vs 72.5% and 73.6% in mIoU) with a simpler network structure (17M vs 30M and 38M parameters) on the ScanNet dataset and sets a new state-of-the-art on the Matterport3D dataset.

To summarize, our contributions are threefold:

1. We propose a novel deep architecture, VMNet, which operates on the voxel and mesh representations, leveraging both the Euclidean and geodesic information.
2. We propose an intra-domain attentive aggregation module, which effectively refines geodesic features through edge connectivities.
3. We propose an inter-domain attentive fusion module, which adaptively combines Euclidean and geodesic features.

2. Related Work

In this section, we first review relevant works on 3D semantic segmentation, organized according to their inherent convolutional categories, and then discuss the application of attention mechanism in 3D semantic segmentation.

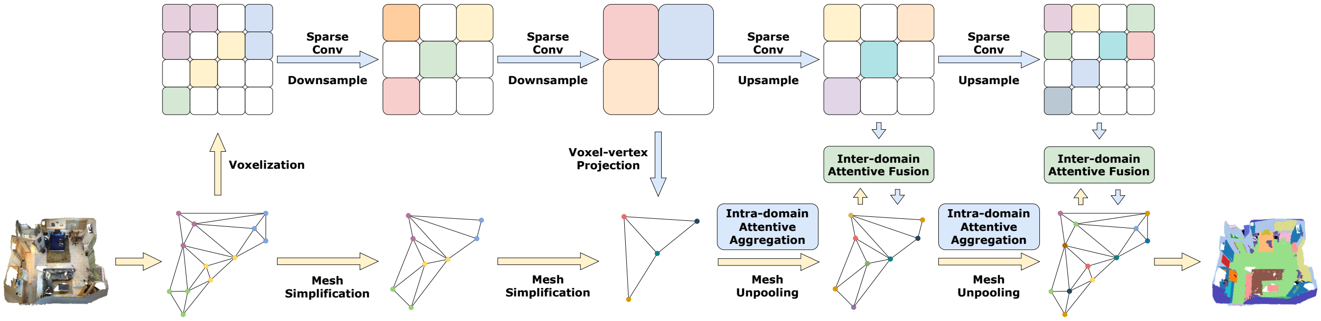

Figure 3. **Overview of Voxel-Mesh Network (VMNet).** Taking a colored mesh as input, we first rasterize it and apply voxel-based sparse convolutions to extract contextual information in the Euclidean domain. These features are then projected from voxels to vertices, and are further aggregated and fused in the geodesic domain producing distinctive per-vertex features. For simplicity, skip connections between the encoder and decoder are neglected here and only three levels of hierarchical voxel downsampling and mesh simplification are shown. The detailed network structure can be found in **Supplementary Section A**.

2D-3D. A conventional way of performing 3D semantic segmentation is to first represent 3D shapes through their 2D projections from various viewpoints, and then leverage existing image segmentation techniques and architectures from the 2D domain [30, 29]. Instead of choosing a global projection viewpoint, some researchers have proposed to project local neighborhoods to local tangent planes and process them with 2D convolutions [57, 68, 23]. Taking the RGB frames as additional inputs, other researchers have proposed methods that combine 2D and 3D features through 2D-3D projection [9, 20]. Although these methods can easily benefit from the success of image segmentation techniques (mainly based on 2D CNNs), they often require a large amount of additional 2D data, involve a complex multi-view projection process, and rely heavily on viewpoint selection. Some of these methods have attempted to utilize geodesic information implicitly through mesh textures [23] or point normal [57]. They achieve fairly decent results but fail to fully exploit the geodesic information.

PointConv & SparseConv. Partly due to the difficulties of handling mesh edges in deep neural networks, most existing 3D semantic segmentation methods take raw point clouds or transformed voxels as input [3, 30, 50, 1, 47, 43, 45]. Point-based methods apply convolutional kernels to the local neighborhoods of points obtained using k-NN or spherical search [70, 61, 60, 55, 22, 65, 21]. Numerous designs of point-based convolutional kernels have been proposed [31, 28, 58, 37, 69]. In the case of voxel-based methods, the raw 3D data is first transformed into a voxel representation and then processed by standard CNNs [39, 44, 62, 72, 24]. To address the cubic memory and computation consumption problem of voxel-based operations, recent works have made efforts to propose efficient sparse voxel convolutions [17, 7, 56]. In both point-based and voxel-based methods, features are aggregated over the Euclidean space only. In contrast, we additionally consider geodesic information of the underlying object surfaces.

GraphConv. Graph convolution networks can be grouped into spectral networks [12, 53] and local filtering networks [38, 2, 40]. Spectral networks work well on clean synthetic data, but are sensitive to reconstruction noise and are thus not applicable to 3D semantic segmentation. Local filtering networks define handcrafted coordinate systems and apply convolutional operations over patches. For 3D semantic segmentation, these methods often perform over local neighborhoods of point clouds [26, 32] and are thus oblivious to the underlying geometry.

Our method falls into both the SparseConv and GraphConv categories. It is similar in spirit to the recent work of Schult et al. [51], which combines a Euclidean-based and a geodesic-based graph convolutions. However, instead of concatenating features obtained from different convolutional filters as done in [51], we first accumulate strong contextual information in the Euclidean domain and then adaptively fuse and aggregate geometric information in the geodesic domain, leading to a significant better segmentation performance (see Section 4.3).

Attention. For 3D semantic segmentation, most existing methods implement attention layers operating on the local neighborhoods of point clouds for feature aggregation [15, 60] or on downsampled point sets for context augmentation [67, 64]. In our work, instead of operating on point clouds, we build attentive operators applying on triangular meshes. Moreover, in contrast to previous works that process features in a single domain, we propose both an intra-domain module and an inter-domain module.

3. Method

In this section, we first introduce the network architecture in Section 3.1. Then the voxel-based contextual feature aggregation branch is described in Section 3.2. Sections 3.3 and 3.4 depict the proposed attentive modules for intra-domain feature aggregation and inter-domain feature

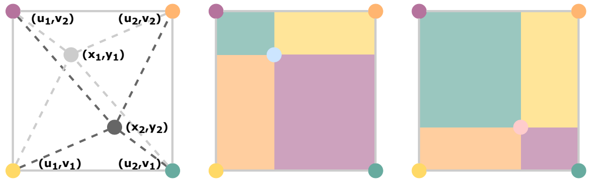

Figure 4. **2D illustration of voxel-vertex projection.** Vertices (x_1, y_1) and (x_2, y_2) share the same set of neighboring voxels but their projected features are different through trilinear interpolation (bilinear interpolation for the 2D case).

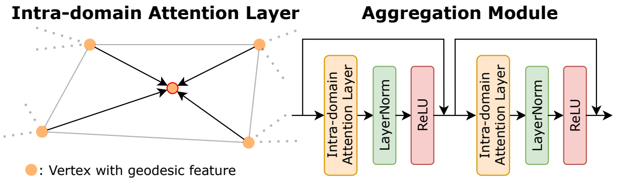
Intra-domain Attention Layer
Aggregation Module
Intra-domain Attention Layer
LayerNorm
ReLU
Intra-domain Attention Layer
LayerNorm
ReLU
: Vertex with geodesic feature

Figure 5. **Illustration of intra-domain attentive aggregation module. (Left)** Intra-domain attention layer operates on mesh vertices aggregating geodesic information through edge connectivities. **(Right)** The aggregation module consists of two attention layers with skip connections.

fusion. Finally, we discuss two well-known mesh simplification methods which build a mesh hierarchy for multi-level feature learning in Section 3.5.

3.1. Network Architecture

VMNet deals with two types of 3D representations: voxels and meshes. As depicted in Fig. 3, the network consists of two branches: according to their operating domains, we denote the upper one as the *Euclidean branch* and the lower one as the *geodesic branch*.

To accumulate contextual information in the Euclidean domain, taking a mesh as input, the colored vertices are first voxelized and then fed to the Euclidean branch. Building on sparse voxel-based convolutions, we construct a U-Net [48] like encoder-decoder structure, where the encoder is symmetric to the decoder, including skip connections between both. Multi-level sparse voxel-based feature maps $(\mathcal{S}^0, ..., \mathcal{S}^l, ..., \mathcal{S}^L)$ can be extracted from the decoder.

Although these contextual features offer valuable semantic cues for scene understanding, their unawareness of the underlying geometric surfaces will lead to sub-optimal results. Therefore, to incorporate geodesic information, the accumulated contextual features are projected from the Euclidean domain to the geodesic domain for further processing (Section 3.2). In the geodesic branch, we prepare a hierarchy of simplified meshes $(\mathcal{M}^0, ..., \mathcal{M}^l, ..., \mathcal{M}^L)$, in which each level of simplified mesh $\mathcal{M}^l$ corresponds to a downsampling level of sparse voxels $\mathcal{S}^l$. Trace maps of the mesh simplification processes are saved for unpooling operations between mesh levels. At the first level of the decoding process (level L), the features are projected from voxels $\mathcal{S}^L$ to mesh vertices $\mathcal{M}^L$ and then refined through intra-domain attentive aggregation (Section 3.3). The resulting geodesic features of $\mathcal{M}^L$ are unpooled to the next level $\mathcal{M}^{L-1}$. At each following level l, the Euclidean features projected from $\mathcal{S}^l$ and the unpooled geodesic features of $\mathcal{M}^l$ are first adaptively combined through inter-domain attentive fusion (Section 3.4) and then the fused features are further refined through intra-domain attentive aggregation before being unpooled to the next level. Please find the detailed network structure in **Supplementary Section A**.

3.2. Voxel-based Contextual Feature Aggregation

Voxelization. At mesh level $\mathcal{M}^0$, with all edge connectivities omitted, the input features (colors) of mesh vertices $\{(V_i, f_i)\}$ are transformed into the voxel cells $\{V_{u,v,w}\}$ by averaging all features f_i whose corresponding coordinate $V_i : (x_i, y_i, z_i)$ falls into the voxel cell (u, v, w):

$$f_{u,v,w} = \frac{1}{N_{u,v,w}} \sum_{i=1}^{n} B[floor(x_i \cdot r) = u, \\ floor(y_i \cdot r) = v, floor(z_i \cdot r) = w] \cdot f_i, \tag{1}$$

where r denotes the voxel resolution, $B[\cdot]$ is the binary indicator of whether vertex V_i belongs to the voxel cell (u, v, w), and $N_{u,v,w}$ is the number of vertices falling into that cell [35].

Contextual Feature Aggregation. To accumulate contextual information, we construct a simple U-Net [48] structure based on voxel convolutions. We adopt the sparse implementation provided by [56].

Voxel-vertex Projection. With the contextual features aggregated in the Euclidean domain, at each level l, we transform the features of voxels $\mathcal{S}^l$ back to vertices $\mathcal{M}^l$ for further processing in the geodesic domain. Inspired by previous works [35, 56], we compute each vertex's feature utilizing trilinear interpolation over its neighboring eight voxels. Through this means, the projected features are distinct even for the vertices sharing the same set of neighboring voxels. A 2D illustration of the projection is shown in Fig. 4.

3.3. Intra-domain Attentive Aggregation Module

After contextual feature aggregation and voxel-vertex projection, to effectively refine the projected features, we design an intra-domain attentive aggregation module operating on the geodesic domain. As shown in Fig. 5 (Left), at each mesh level, we perform attentive aggregation on the graph $\mathcal{G} = (\mathcal{V}, E)$ induced by the underlying mesh $\mathcal{M}$. Note that we neglect the level superscript l to ease readability. Our intra-domain attention layer is based on the standard scalar attention [59], which is often used for point clouds in 3D semantic segmentation, but not for triangular

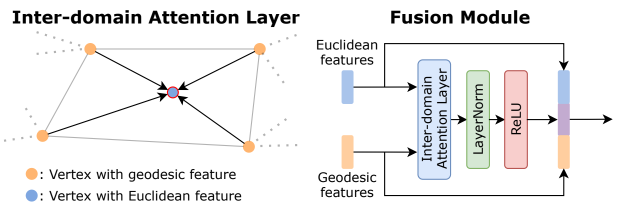
Inter-domain Attention Layer
Fusion Module
Euclidean features
Geodesic features
Inter-domain Attention Layer
LayerNorm
ReLU
●: Vertex with geodesic feature
●: Vertex with Euclidean feature

Figure 6. **Illustration of inter-domain attentive fusion module. (Left)** Inter-domain attention layer adaptively combines geodesic features and Euclidean features on mesh vertices. **(Right)** The fused feature map generated by the inter-domain attention layer is further combined with the original geodesic feature map and the projected Euclidean feature map through concatenation.

meshes. Specifically, at layer k, the output feature $f_i^{'geo}$ of vertex V_i with an input feature f_i^{geo} is computed as:

$$\begin{aligned} f_i^{'geo} &= \rho_k^{intra}(f_i^{geo}) + \sum_{j\in N_i} \omega_{ij}\alpha_k^{intra}(f_j^{geo}), \\ \omega_{ij} &= softmax(\frac{\varphi_k^{intra}(f_i^{geo})^T\psi_k^{intra}(f_j^{geo})}{\sqrt{d}}), \end{aligned} \tag{2}$$

where N_i is the one-ring neighborhood of vertex V_i. The functions ρ_k^{intra}, α_k^{intra}, φ_k^{intra}, and ψ_k^{intra} are vertex-wise feature transformations implemented by MLP, ω_{ij} is the attention coefficient, and d is the size of output feature channels. Since the positional information is naturally embedded in the voxel-based contextual feature aggregation step, we do not implement a position encoding function explicitly. Our attention layer is inspired by the implementation in [52], which operates on abstract graphs for semi-supervised node classification, while our method operates on 3D mesh graphs for geodesic feature aggregation.

Building on the intra-domain attention layer, we design an aggregation module performing two steps of attentive feature aggregation on each simplified mesh level (see Fig. 5 (Right)).

3.4. Inter-domain Attentive Fusion Module

Operating on both the voxel and mesh representations poses a demand for Euclidean and geodesic feature fusion. To adaptively combine features from the two domains, we propose an inter-domain attentive fusion module. As depicted in Fig. 6 (Left), between each pair of sparse voxel level $\mathcal{S}$ and mesh level $\mathcal{M}$ (except for level L), we perform attentive fusion on the same graph $\mathcal{G} = (\mathcal{V}, E)$ as the one used for intra-domain aggregation (level superscript l is neglected). However, unlike intra-domain attention, which processes features in the same domain, inter-domain attention takes as input both the geodesic features f^{geo} and the Euclidean features f^{euc} projected from voxels. At layer k, the fused feature f_i^{fuse} of vertex V_i is computed as:

$$\begin{aligned} f_i^{fuse} &= \rho_k^{inter}(f_i^{euc}) + \sum_{j\in N_i} \omega_{ij}\alpha_k^{inter}(f_j^{geo}), \\ \omega_{ij} &= softmax(\frac{\varphi_k^{inter}(f_i^{euc})^T\psi_k^{inter}(f_j^{geo})}{\sqrt{d}}), \end{aligned} \tag{3}$$

where N_i is the same one-ring neighborhood of vertex V_i as the one used for intra-domain aggregation. Unlike the one in intra-domain attention, the inter-domain attention coefficient ω_{ij} is conditioned on both the Euclidean and geodesic features enabling the network to adaptively fuse features from the two domains.

As shown in Fig. 6 (Right), the proposed inter-domain attentive fusion module takes both the Euclidean features and the geodesic features as inputs. These features are fed to one inter-domain attention layer followed by layer normalization and ReLU activation. Before being passed on for further processing, the fused feature map is concatenated with the projected Euclidean feature map and the original geodesic feature map.

3.5. Mesh Simplification

To construct a deep architecture for multi-level feature learning, we generate a hierarchy of mesh levels $(\mathcal{M}^0, ..., \mathcal{M}^l, ..., \mathcal{M}^L)$ of increasing simplicity, interlinked by pooling trace maps. Each level of simplified mesh corresponds to a level of downsampled 3D sparse voxels. For mesh simplification, there are two well-known methods from the geometry processing domain: Vertex Clustering (VC) [49] and Quadric Error Metrics (QEM) [16]. During the vertex clustering process, a 3D uniform grid with cubical cells of a fixed side length is placed over the input graph and all vertices falling into the same cell are grouped. This generates uniform-sampled simplified meshes, possibly with topology changes and non-manifold faces. On the contrary, the QEM method incrementally collapses mesh edges according to an approximate error of the geometric distortion introduced by this collapse, and thus has explicit control over mesh topology. Since our goal is to extract meaningful geodesic information, we prefer the QEM method for its better topology-preserving property. However, directly applying the QEM method on the original meshes results in high-frequency signals in noisy areas [51]. Therefore, we apply the VC method on the original mesh for the first two mesh levels and then apply the QEM method for the remaining mesh levels. We present an ablation study on mesh simplification methods in Section 4.4. Image illustrations can be found in **Supplementary Section B**.

4. Experiments

To demonstrate the effectiveness of our proposed method, we now present various experiments conducted on two large-scale 3D scene segmentation datasets, which contain meshed point clouds of various indoor scenes. We first introduce the datasets and evaluation metrics that we used in Section 4.1, and then present the implementation details for reproduction in Section 4.2. We report the results on the ScanNet and Matterport3D datasets in Section 4.3, and the ablation studies in Section 4.4.

4.1. Datasets and Metrics

ScanNet v2 [8]. ScanNet dataset contains 3D meshed point clouds of a wide variety of indoor scenes. Each scene is provided with semantic annotations and reconstructed surfaces represented by a textured mesh. The dataset contains 20 valid semantic classes. We perform all our experiments using the public training, validation, and test split of 1201, 312, and 100 scans, respectively.

Matterport3D [4]. Matterport3D is a large RGB-D dataset of 90 building-scale scenes. Similar to ScanNet, the full 3D mesh reconstruction of each building and semantic annotations are provided. The dataset contains 21 valid semantic classes. Following previous works [51, 45, 55, 57, 9, 23], we split the whole dataset into training, validation, and test sets of size 61, 11, and 18, respectively.

Metrics. For evaluation, we use the same protocol as introduced in previous works [51, 45, 7, 17]. We report mean class intersection over union (mIoU) results for ScanNet and mean class accuracy for Matterport3D. During testing, we project the semantic labels to the vertices of the original meshes and test directly on meshes.

4.2. Implementation Details

In this section, we discuss the implementation details for our experiments. VMNet is coded in Python and PyTorch (Geometric) [14, 42]. All the experiments are conducted on one NVIDIA Tesla V100 GPU.

Data Preparation. We perform training and inference on full meshes without cropping. For the Euclidean branch of VMNet, input meshes are voxelized at a resolution of 2 cm. To compute the hierarchical mesh levels accordingly for the geodesic branch, we first apply the VC method on the input mesh with the respective cubical cell lengths of 2 cm and 4 cm for the first two mesh levels. For each remaining level, the QEM method is applied to simplify the mesh until the vertex number is reduced to 30% of its preceding mesh level. For better generalization ability, edges of all mesh levels are randomly sampled during training. We use the vertex colors as the only input features and apply data augmentation, including random scaling, rotation around the gravity axis, spatial translation, and chromatic jitter.

Training Details. We train the network end-to-end by minimizing the cross entropy loss using Momentum SGD with the Poly scheduler decaying from learning rate 1e-1.

Method	mIoU(%)	Conv Category
TangentConv [57]	43.8	2D-3D
SurfaceConvPF [68]	44.2	
3DMV [9]	48.3	
TextureNet [23]	56.6	
JPBNet [6]	63.4	
MVPNet [25]	64.1	
V-MVFusion [29]	74.6	
BPNet* [20]	**74.9**	
PointNet++ [45]	33.9	PointConv
FCPN [46]	44.7	
PointCNN [33]	45.8	
DPC [13]	59.2	
MCCN [19]	63.3	
PointConv [65]	66.6	
KPConv [58]	68.4	
JSENet [21]	69.9	
SparseConvNet [17]	72.5	SparseConv
MinkowskiNet [7]	73.6	
SPH3D-GCN [32]	61.0	GraphConv
HPEIN [26]	61.8	
DCM-Net [51]	65.8	
VMNet (**Ours**)	**74.6**	Sparse+Graph Conv

Table 1. **Mean intersection over union scores on ScanNet Test [8].** Detailed results can be found on the ScanNet benchmarking website[2]. * indicates a concurrent work.

4.3. Results and Analysis

Quantitative Results. We present the performance of our approach compared to recent competing approaches on the ScanNet benchmark [8] in Table 1. All the methods are grouped by the approaches' inherent convolutional categories as discussed in Section 2. As shown in Table 1, our method leads to a 74.6% mIoU score, achieving a significant performance gain of 8.8 % mIoU comparing to the existing best-performing graph convolutional approach, i.e., DCM-Net [51], and 1.0 % mIoU comparing to the leading sparse convolutional approach, i.e., MinkowskiNet [7]. Our method achieves results comparable to the SOTA 2D-3D method BPNet [20], which is a concurrent work on CVPR2021 utilizing both 2D and 3D data while VMNet takes as input only the 3D data. For a fair comparison, the result of OccuSeg [18] is not listed in this table, since it utilizes extra instance labels for training. We also evaluate our algorithm on the novel Matterport3D dataset [4] and report the results in Table 2. VMNet achieves overall state-of-the-art results outperforming the previous best method by 1% in terms of mean class accuracy. Since some methods only report results in one of these two datasets, the listed methods in Tables 1 and 2 are different.

Qualitative Comparison. Fig. 7 shows our qualitative re-

[2] http://kaldir.vc.in.tum.de/scannet_benchmark/

Method	mAcc(%)	Cat	wall	floor	cab	bed	chair	sofa	table	door	wind	shf	pic	cntr	desk	curt	ceil	fridg	show	toil	sink	bath	other
TangentConv [57]	46.8	I	56.0	87.7	41.5	73.6	60.7	69.3	38.1	55.0	30.7	33.9	50.6	38.5	19.7	48.0	45.1	22.6	35.9	50.7	49.3	56.4	16.6
3DMV [9]	56.1	I	79.6	95.5	59.7	82.3	70.5	73.3	48.5	64.3	55.7	8.3	55.4	34.8	2.4	**80.1**	94.8	4.7	54.0	71.1	47.5	76.7	19.9
TextureNet [23]	63.0	I	63.6	91.3	47.6	82.4	66.5	64.5	45.5	69.4	60.9	30.5	**77.0**	**42.3**	44.3	75.2	92.3	49.1	66.0	80.1	**60.6**	86.4	27.5
SplatNet [55]	26.7	II	**90.8**	95.7	30.3	19.9	77.6	36.9	19.8	33.6	15.8	15.7	0.0	0.0	0.0	12.3	75.7	0.0	0.0	10.6	4.1	20.3	1.7
PointNet++ [45]	43.8	II	80.1	81.3	34.1	71.8	59.7	63.5	**58.1**	49.6	28.7	1.1	34.3	10.1	0.0	68.8	79.3	0.0	29.0	70.4	29.4	62.1	8.5
ScanComplete [11]	44.9	III	79.0	**95.9**	31.9	70.4	68.7	41.4	35.1	32.0	37.5	17.5	27.0	37.2	11.8	50.4	**97.6**	0.1	15.7	74.9	44.4	53.5	21.8
DCM-Net [51]	66.2	IV	78.4	93.6	**64.5**	**89.5**	70.0	**85.3**	46.1	**81.3**	**63.4**	43.7	73.2	39.9	**47.9**	60.3	89.3	**65.8**	43.7	**86.0**	49.6	87.5	**31.1**
VMNet (**Ours**)	**67.2**	V	85.9	94.4	56.2	**89.5**	**83.7**	70.0	54.0	76.7	63.2	**44.6**	72.1	29.1	38.4	79.7	94.5	47.6	**80.1**	85.0	49.2	**88.0**	29.0

Table 2. **Mean class accuracy scores on the Matterport3D Test [4].** The same network definition as for the ScanNet benchmark is used. Conv Category: (I) 2D-3D, (II) PointConv, (III) VoxelConv, (IV) GraphConv, (V) Sparse+Graph Conv.

Method	Params (M)	Latency (ms)		mIoU(%)
		Ori	TS	
SparseConvNet [17]	30.1	712	102	72.5
MinkowskiNet [7]	37.8	629	105	73.6
VMNet (**Ours**)	17.5	-	107	**74.6**

Table 3. **Comparison of run-time complexity against SOTA sparse voxel-based methods.** For a fair comparison, we report the latencies of both their original versions (Ori) and our implementations using the same type of sparse convolution (TS) as VMNet.

Information	mIoU(%)
Geo Only	58.1
Euc Only	71.0
VMNet(Geo+Euc)	**73.3**

Baseline	Intra	Inter	mIoU(%)
✓			70.2
✓	✓		72.1
✓	✓	✓	**73.3**

Table 4. **Ablation study**: **(Left)** Euclidean and geodesic information; **(Right)** Network components.

sults on the ScanNet validation set. Compared to the SOTA sparse voxel-based method SparseConvNet, which operates in the Euclidean domain solely, VMNet generates more distinctive features for close-located objects and better handles complex geometries thanks to the combined Euclidean and geodesic information. More qualitative results can be found in **Supplementary Section C**.

Complexity. We compare our method with two SOTA sparse voxel-based methods, i.e., SparseConvNet [17] and MinkowskiNet [7], for their run-time complexity. We randomly select a scene from the ScanNet validation set and compute the latency results by averaging the inference time of 100 forward passes. Although the accuracies of sparse voxel-based methods are not dependent on the implementation of sparse convolution, the latencies of these methods are highly dependent on the implementation. Therefore, we re-implement SparseConvNet and MinkowskiNet using the same version of sparse convolution (torchsparse [56]) as VMNet for a fair comparison. As shown in Table 3, VMNet achieves the highest mIoU score with the least number of parameters. It implies that, compared to extracting features in the Euclidean domain alone, combining Euclidean and geodesic information leads to more effective aggregation of features, even with a simpler network structure. The latency of VMNet is slightly higher than our new implementations of the other two methods. This is caused by the unoptimized projection operations, which are left for future improvement. More complexity comparisons can be found in **Supplementary Section D**.

Operator	mIoU(%)
Vector Attention	72.3
EdgeConv	72.6
Scalar Attention	**73.3**

Method	mIoU(%)
VC only	72.3
QEM only	72.9
VC + QEM	**73.3**

Table 5. **Ablation study**: **(Left)** Attentive operators; **(Right)** Mesh simplification.

4.4. Ablation Study

In this section, we conduct a number of controlled experiments that demonstrate the effectiveness of building modules in VMNet, and also examine some specific decisions in VMNet design. Since the test set of ScanNet is not available for multiple tests, all experiments are conducted on the validation set, keeping all hyper-parameters the same.

Euclidean and Geodesic Information. In Section 3, we advocate the combination of Euclidean and geodesic information. To investigate their impacts, we compare VMNet to two baseline networks: "Euc only" is a U-Net structure based on sparse convolutions operating on voxels and "Geo only" has the same structure but is based on the proposed intra-domain attention layers operating on meshes. For a fair comparison, we keep the layer numbers of these baselines the same as the Euclidean branch of VMNet but increase their channel numbers to make sure all the compared methods have similar parameter sizes. As shown in Table 4 (Left), VMNet outperforms the two baselines showcasing the benefit of combining information from the two domains.

Network Components. In Table 4 (Right), we evaluate the effectiveness of each component of our method. "Baseline" represents the Euclidean branch of VMNet, which is a U-Net network built on voxel convolutions. "Intra" refers to the intra-domain attentive aggregation module and "Inter" refers to the inter-domain attentive fusion module. As shown in the table, by combining the intra-domain attentive aggregation module with the baseline, we can improve the performance by 1.9%. This improvement is brought by the introduction of geodesic information through feature refinement on meshes. From the inter-domain attentive fusion module, we further gain about 1.2% improvement in performance by adaptive fusion of features from the two domains.

Attentive Operators. In Sections 3.3 and 3.4, we adopt the standard scalar attention [59] to build the intra-domain attentive aggregation module and the inter-domain attentive fusion module. In Table 5 (Left), we evaluate the influ-

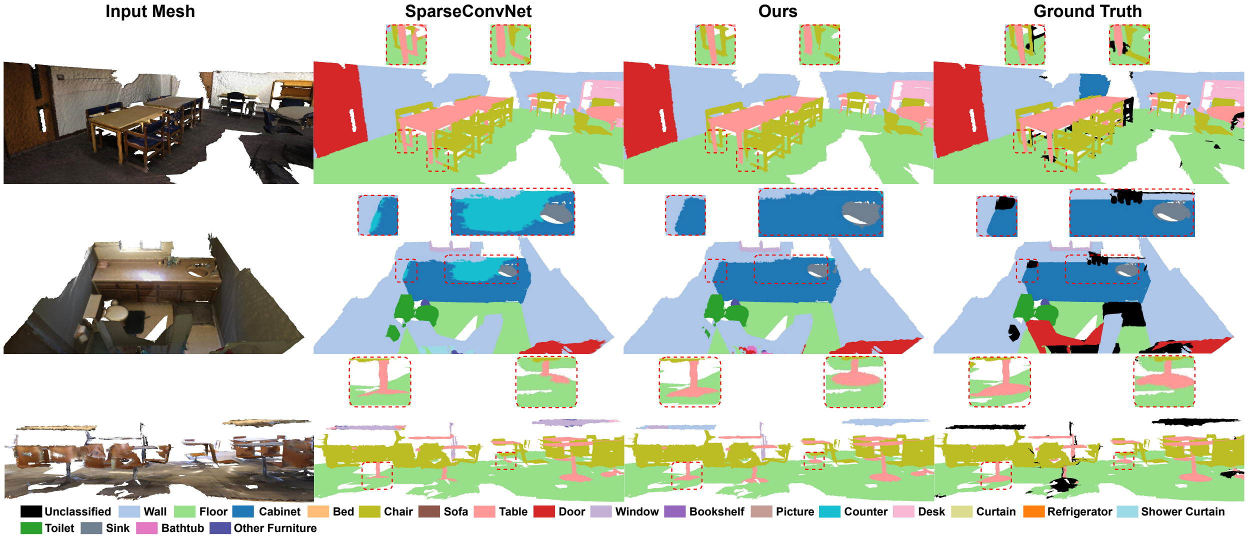

Figure 7. **Qualitative results on ScanNet Val [8].** The key parts for comparison are highlighted by dotted red boxes.

ence of different forms of attentive operators in our architecture. “Scalar Attention” refers to the operators used in VMNet as presented in Equations 2 and 3. “Vector Attention” represents a variant of Scalar Attention, in which attention weights are not scalars but vectors, which can modulate individual feature channels. It is widely adopted in previous attention-based methods operating on 3D point clouds [60, 71]. Moreover, we implement a non-attention baseline building on the popular EdgeConv [61], which is originally proposed to operate on kNN graphs of 3D point clouds. As shown in the table, the scalar attention used in VMNet achieves the best result outperforming the non-attention baseline “EdgeConv” by 0.7% and the attentive variant “Vector Attention” by 1.0%. Interestingly, the non-attention baseline “EdgeConv” performs slightly better than the attention-based baseline “Vector Attention”. A possible reason is that “Vector Attention” adaptively modulates each individual feature channel and this property appears to be overfitting in our case.

Mesh Simplification. In Section 3.5, we discuss two mesh simplification methods Vertex Clustering (VC) and Quadric Error Metrics (QEM) for multi-level feature learning. We apply the VC method on the first two mesh levels to remove high-frequency signals in noisy areas, and then apply the QEM method on the remaining mesh levels for its better topology-preserving property. To justify our choice, we train three models with the same network definition but performing on different mesh hierarchies, and compare their performances in Table 5 (Right). “VC+QEM” refers to the mesh hierarchy simplified by the combination of the VC and QEM methods as described in Section 4.2. For “VC only”, at each mesh level $\mathcal{M}^l$, we set the cubical cell lengths of the VC method to the same size as the lengths of voxels in the corresponding voxel level $\mathcal{S}^l$. For “QEM only”, at each mesh level $\mathcal{M}^l$, the QEM method simplifies the mesh until the vertex number is reduced to 30% of its preceding mesh level $\mathcal{M}^{l-1}$. As shown in the table, we witness a significant performance gap of 1.0% between the results of “VC+QEM” and “VC only”. We assume that the more faithful geodesic information provided by meshes simplified through the QEM method leads to the performance gain. We also notice that the performance of “QEM only” is slightly lower than the one of “VC+QEM”. It may be caused by the resulting high-frequency noises of directly applying the QEM method on the original meshes.

5. Conclusion

In this paper, we have presented a novel 3D deep architecture for semantic segmentation of indoor scenes, named Voxel-Mesh Network (VMNet). Aiming at addressing the problem of lacking consideration for the geodesic information in voxel-based methods, VMNet takes advantages of both the semantic contextual information available in voxels and the geometric surface information available in meshes to perform geodesic-aware 3D semantic segmentation. Extensive experiments show that VMNet achieves state-of-the-art results on the challenging ScanNet and Matterport3D datasets, significantly improving over strong baselines. We hope that our work will inspire further investigation of the idea of combining Euclidean and geodesic information, the development of new intra-domain and inter-domain modules, and the application of geodesic-aware networks to other tasks, such as 3D instance segmentation.

Acknowledgements. This work was partially supported by a Strategic Research Grant from City University of Hong Kong (Project No. 7005729).

References

[1] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. *IEEE Robotics and Automation Letters*, 3(4):3145–3152, 2018. 3

[2] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in neural information processing systems*, pages 3189–3197, 2016. 3

[3] Alexandre Boulch, Bertrand Le Saux, and Nicolas Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. *3DOR*, 2:7, 2017. 3

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1, 2, 6, 7

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[6] Hung-Yueh Chiang, Yen-Liang Lin, Yueh-Cheng Liu, and Winston H Hsu. A unified point-based framework for 3d segmentation. In *International Conference on 3D Vision (3DV)*, pages 155–163. IEEE, 2019. 6

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2, 3, 6, 7

[8] 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*, 2017. 1, 2, 6, 8

[9] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018. 3, 6, 7

[10] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 1

[11] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018. 7

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 3

[13] Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9463–9469. IEEE, 2020. 6

[14] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 6

[15] Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems*, 2020. 3

[16] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. 5

[17] 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. 1, 2, 3, 6, 7

[18] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2940–2949, 2020. 6

[19] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 6

[20] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3, 6

[21] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-lan Tai. Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 222–239, 2020. 3, 6

[22] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018. 3

[23] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas J Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4440–4449, 2019. 3, 6, 7

[24] Shi-Sheng Huang, Ze-Yu Ma, Tai-Jiang Mu, Hongbo Fu, and Shi-Min Hu. Supervoxel convolution for online 3d semantic segmentation. *ACM Transactions on Graphics (TOG)*, 2021. 3

[25] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 6

[26] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction

network for point cloud semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10433–10441, 2019. 3, 6

[27] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE transactions on visualization and computer graphics*, 21(11):1241–1250, 2015. 1

[28] Artem Komarichev, Zichun Zhong, and Jing Hua. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7421–7430, 2019. 3

[29] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–535. Springer, 2020. 3, 6

[30] Felix Järemo Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Deep projective 3d semantic segmentation. In *International Conference on Computer Analysis of Images and Patterns*, pages 95–107. Springer, 2017. 3

[31] Huan Lei, Naveed Akhtar, and Ajmal Mian. Octree guided cnn with spherical kernels for 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9631–9640, 2019. 3

[32] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 3, 6

[33] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 6

[34] Yingwei Li, Xiaojie Jin, Jieru Mei, Xiaochen Lian, Linjie Yang, Cihang Xie, Qihang Yu, Yuyin Zhou, Song Bai, and Alan L Yuille. Neural architecture search for lightweight non-local networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10297–10306, 2020. 2

[35] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, pages 965–975, 2019. 1, 4

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[37] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3d point cloud understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1578–1587, 2019. 3

[38] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015. 3

[39] 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. 1, 3

[40] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017. 3

[41] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 2

[42] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[43] 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. 3

[44] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5648–5656, 2016. 1, 3

[45] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 3, 6, 7

[46] Dario Rethage, Johanna Wald, Jurgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 596–611, 2018. 6

[47] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017. 3

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[49] Jarek Rossignac and Paul Borrel. Multi-resolution 3d approximations for rendering complex scenes. In *Modeling in computer graphics*, pages 455–465. Springer, 1993. 5

[50] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Classification of point cloud scenes with multiscale voxel deep network. *arXiv preprint arXiv:1804.03583*, 2018. 3

[51] Jonas Schult, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dualconvmesh-net: Joint geodesic and

euclidean convolutions on 3d meshes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8612–8622, 2020. 3, 5, 6, 7

[52] Yunsheng Shi, Zhengjie Huang, Shikun Feng, and Yu Sun. Masked label prediction: Unified massage passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020. 5

[53] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3693–3702, 2017. 3

[54] 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. 1

[55] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018. 3, 6, 7

[56] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 685–702, 2020. 3, 4, 7

[57] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018. 3, 6, 7

[58] 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 International Conference on Computer Vision*, pages 6411–6420, 2019. 3, 6

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2, 4, 7

[60] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019. 3, 8

[61] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019. 3, 8

[62] Zongji Wang and Feng Lu. Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes. *IEEE transactions on visualization and computer graphics*, 2019. 1, 3

[63] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016. 1

[64] Chi-Chong Wong and Chi-Man Vong. Efficient outdoor 3d point cloud semantic segmentation for critical road objects and distributed contexts. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 499–514. Springer International Publishing, 2020. 3

[65] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 3, 6

[66] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 1

[67] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020. 3

[68] Yuqi Yang, Shilin Liu, Hao Pan, Yang Liu, and Xin Tong. Pfcnn: convolutional neural networks on 3d surfaces using parallel frames. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13578–13587, 2020. 3, 6

[69] Chris Zhang, Wenjie Luo, and Raquel Urtasun. Efficient convolutions for real-time semantic segmentation of 3d point clouds. In *2018 International Conference on 3D Vision (3DV)*, pages 399–408. IEEE, 2018. 3

[70] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5565–5573, 2019. 3

[71] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020. 8

[72] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 1, 3