DualConvMesh-Net: Joint Geodesic and Euclidean Convolutions on 3D Meshes
Supplementary Material

Abstract

In the supplementary material, we provide further insights into the architectural design choices we make in order to leverage the potential of combining geodesic and Euclidean information. Furthermore, we present detailed network descriptions used for the ablation study as well as the official benchmarks. Finally, we show additional scores and qualitative results on all three benchmarks.

1. Architectural design choices

In this section, we give more details about our architectural design choices. ① By altering the filter ratio between geodesic and Euclidean convolutions for each mesh level, we further motivate the assumptions about the characteristics of Euclidean and geodesic convolutions and back them up with empirical evidence. ② We show the impact of the number of mesh levels for the DCM-Net architecture. ③ We compare activation functions in our architecture.

Ratio between geodesic and Euclidean filters. Following the intuition that geodesic convolutions mainly benefit from high-frequency mesh information in order to learn the inherent shape of objects, we want to learn more geodesic than Euclidean features in high resolution mesh levels. Contrastingly, Euclidean features are beneficial for localizing objects in the scene which is better performed in lower resolutions. In order to verify this intuition, we present the results of an experiment in which we systemically modified the ratio of geodesic and Euclidean filters per mesh level.

In Table 1, more geodesic filters in the first two levels and more Euclidean filters in later two levels bring significant performance gains over other ratio settings. We see this as a clear indicator that our assumption about the inherent properties about Euclidean and geodesic convolutions hold.

Number of mesh levels. In Table 2, we experimentally show the importance of multi-scale hierarchies for semantic segmentation for meshed point clouds. We see a clear trend that an increased number of mesh levels with different resolutions bring a significant performance gain.

Geodesic | Euclidean Ratio			
level 1-2	level 3-4	mIoU (± stdev)	Δ
25% | 75%	75% | 25%	66.0 (± 0.14)	+2.3
75% | 25%	75% | 25%	66.1 (± 0.19)	+2.2
25% | 75%	25% | 75%	66.9 (± 0.20)	+1.4
50% | 50%	50% | 50%	67.5 (± 0.13)	+0.8
75% | 25%	25% | 75%	**68.3** (± 0.12)	

Table 1: **Geodesic/Euclidean filter ratio per mesh level.** Geodesic convolutions are particularly useful in early mesh levels, when high frequency signals of the mesh are still preserved. In later mesh levels, we benefit from Euclidean convolutions for localizing objects better. This observation is materialized in a larger ratio of geodesic filters in early levels, whereas we use more Euclidean filters in later levels. (Level 1-2 use 64 and level 3-4 use 96 filters in total.)

#level	mIoU (± stdev)	Δ
2	54.4 (± 0.07)	+12.9
3	64.0 (± 0.14)	+3.3
4	**67.3** (± 0.22)	

Table 2: **Influence of the number of mesh levels.** We observed that the multi-scale architecture has a strong impact on the performance of the algorithm. With decreasing effect, more mesh levels bring performance gains. (Experiments where conducted with QEM pooling and geodesic/radius neighborhoods in our DCM-Net.)

Activation functions. Recent publications on 3D scene segmentation rely on Leaky ReLU activation functions [13]. In Table 3, we compare standard ReLU with LeakyReLU activation functions. We conclude that for our architecture LeakyReLU activations do not bring any benefits and decrease the performance by 1.6% mIoU.

2. Detailed network descriptions

In the ablation study of the main paper, we focus particularly on the comparability of our proposed networks. We

activation function	mIoU (± stdev)	Δ
Leaky ReLU	65.7 (± 0.14)	+1.4
ReLU	**67.3**(± 0.22)	

Table 3: **Comparison of activation functions.** As Leaky ReLU gains popularity, we compare it with standard ReLU activation functions. We conclude that default ReLU units work significantly better for our architecture. (Experiments are conducted with QEM pooling and geodesic/radius neighborhoods in a DCM-Net.)

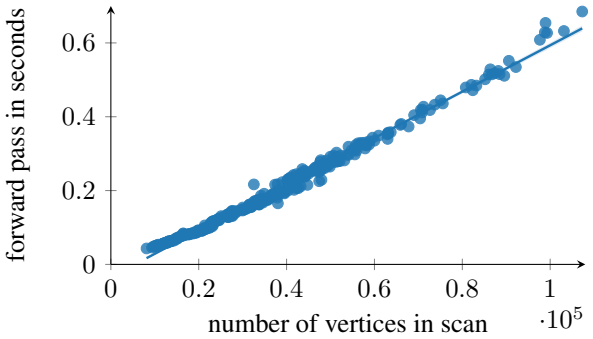


Figure 1: **Runtime wrt. the number of vertices**. We see a linear relationship between the forward pass runtime for full rooms of the ScanNet v2 validation set and the number of vertices in the input.

compare basic instantiations of SCM-Nets with its DCM-Net equivalents in the ablation study (see Table 5). Note that we ensure the same size of hidden and output channels for each edge convolution and dual convolution. That is, the 128 hidden and 64 output channels of single edge convolutions of the SCM-Nets are halved resulting in 64 hidden and 32 output features for geodesic and Euclidean filters of the dual convolutions. Thus, DCM-Nets have 15% less parameters than their SCM-Net equivalents.

However, we use extended networks for obtaining final scores on the benchmarks. Motivated by Table 1, we additionally vary the ratio of geodesic and Euclidean filters and changed the number of features in each mesh level. In the following paragraphs, we give detailed network descriptions for each benchmark.

Network architectures for ScanNet / Matterport3D. We use the DCM-Net with 75% geodesic out of 48 features in the first two mesh levels and 25% geodesic out of 96 features in the last two mesh levels. We use batch normalization and ReLU activations for the edge convolutions. In Table 6a, we show the detailed network architecture for the ScanNet and Matterport3D benchmark.

Special provisions for S3DIS. In contrast to ScanNet and Matterport3D, S3DIS is characterized by the comparably lower resolution of its underlying mesh structure. In order to use the ground truth information of the official point clouds sampled from these meshes, we artificially increase the resolution of the mesh by splitting edges exactly in the middle if the edge length does not fall under 2 cm. We create new triangles by connecting the old vertices with their adjacent vertices at the midst of the edges. Thus, we obtain 4 smaller triangles from the original triangle. We subsequently interpolate the ground truth information on this newly created mesh. In Figure 2, we provide an illustration of the preprocessing pipeline for S3DIS.

dataset	single run	majority	Δ
ScanNet [4] (*test*)	65.3	65.8	0.5
S3DIS [1] (Area-5)	63.8	64.0	0.2
S3DIS [1] (k-fold)	69.4	69.7	0.3
Matterport3D [2]	65.5	66.2	0.7

Table 4: **Majority voting.** By using majority voting with 100 runs on augmented scenes, we experience performance gains up to 0.5 % mIoU on ScanNet and S3DIS. Our scores on Matterport3D increase by 0.7 % mAcc compared to the single run variant with no test time augmentations.

Since the original resolution of the mesh is low, we do not benefit from increasing the number of geodesic filters in the early levels, as we motivate for ScanNet in Table 1. Thus, we set the ratio of geodesic convolutions in each level to 50%, similarly to the ablation study in the main paper. In Table 6b, we provide the adapted network structure.

3. Runtime

In Figure 1, we provide forward pass times for our ScanNet benchmark model with respect to the input size. We see a linear relationship between the number of input vertices and the runtime which is always well under 0.7 seconds for all scans. Overall, the mean runtime for the ScanNet validation set is 211ms with an average input size of 39161 vertices. We perform this experiment with a Tesla V100.

4. Quantitative and qualitative results

We provide additional segmentation results on Stanford Large-Scale 3D Indoor Spaces (S3DIS) to allow an in-depth comparison with competitive approaches. In Table 7 and 8, we report class-wise segmentation results on S3DIS k-fold and Area 5. Moreover, we show further qualitative results on S3DIS [1] and Matterport3D [2] in Figures 3 and 4.

Majority Voting. To obtain the final scores for the benchmarks, we leverage *majority voting* with 100 runs of the best performing model on augmented test scenes. In Table 4, we compare single runs of the models on non-augmented scenes against the majority voting method explained before.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6

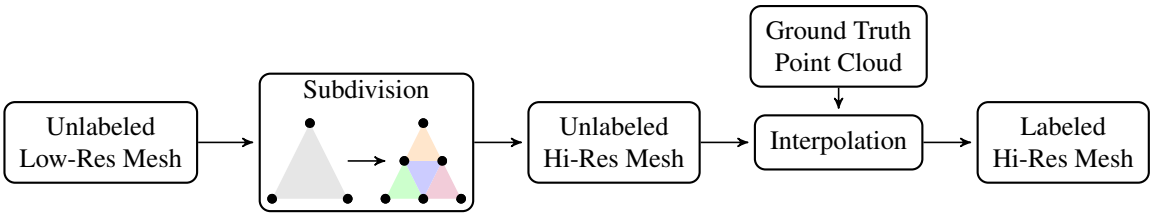


Figure 2: **Preprocessing Pipeline for S3DIS.** Our approach requires meshes as input for which the S3DIS data set does not provide an RGB + Label format. Therefore, we establish a preprocessing pipeline in order to leverage low-resolution meshes given by the dataset. Here, we perform midpoint subdivision to artificially enhance the resolution of the mesh, before interpolating RGB colors as well as labels from the ground truth point cloud onto the mesh.

#level	level type	module type	filters
1	encoder	edge+BN+ReLU	$(9, 128, 64)$
1	encoder	edge+BN+ReLU	$(128, 128, 64)$
1	encoder	edge+BN+ReLU	$(128, 128, 64)$
2	encoder	edge+BN+ReLU	$(128, 128, 64)$
2	encoder	edge+BN+ReLU	$(128, 128, 64)$
2	encoder	edge+BN+ReLU	$(128, 128, 64)$
3	encoder	edge+BN+ReLU	$(128, 128, 64)$
3	encoder	edge+BN+ReLU	$(128, 128, 64)$
3	encoder	edge+BN+ReLU	$(128, 128, 64)$
4	encoder	edge+BN+ReLU	$(128, 128, 64)$
4	encoder	edge+BN+ReLU	$(128, 128, 64)$
4	encoder	edge+BN+ReLU	$(128, 128, 64)$
3	decoder	edge+BN+ReLU	$(256, 128, 64)$
3	decoder	edge+BN+ReLU	$(128, 128, 64)$
3	decoder	edge+BN+ReLU	$(128, 128, 64)$
2	decoder	edge+BN+ReLU	$(256, 128, 64)$
2	decoder	edge+BN+ReLU	$(128, 128, 64)$
2	decoder	edge+BN+ReLU	$(128, 128, 64)$
1	decoder	edge+BN+ReLU	$(256, 128, 64)$
1	decoder	edge+BN+ReLU	$(128, 128, 64)$
1	decoder	edge+BN+ReLU	$(128, 128, 64)$
1	final	Lin+BN+ReLU	$(64, 32)$
1	final	Lin	$(32, 21)$
		# parameters:	$\mathbf{564,949}$

(a) **SCM-Net architecture.** We use SCM-Nets for the ablation study. Here, we only consider either geodesic or Euclidean neighborhood information and do not fuse this information.

#level	level type	module type	filters
1	encoder	edge+BN+ReLU	$2 * (9, 64, 32)$
1	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
1	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
3	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
3	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
3	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
4	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
4	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
4	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
3	decoder	edge+BN+ReLU	$2 * (256, 64, 32)$
3	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
3	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	decoder	edge+BN+ReLU	$2 * (256, 64, 32)$
2	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
1	decoder	edge+BN+ReLU	$2 * (256, 64, 32)$
1	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
1	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
1	final	Lin+BN+ReLU	$(64, 32)$
1	final	Lin	$(32, 21)$
		# parameters:	$\mathbf{478,933}$

(b) **DCM-Net architecture.** We perform convolutions in the geodesic and Euclidean space simultaneously and subsequently concatenate the features. Note that the total size of hidden and output features for each dual convolution equals its SCM-Net edge convolution equivalent. We are therefore able to perform fair comparisons between these two types.

Table 5: **Architectures for the ablation study.** In our ablation study, we experimentally prove the effectiveness of combining geodesic and Euclidean convolutions. We propose SCM-Nets for applying convolutions either in the geodesic or Euclidean space and DCM-Nets which jointly perform convolutions in the geodesic and Euclidean space.

#level	level type	filters	
		geodesic	Euclidean
1	encoder	$(9, 96, 48)$	$(9, 32, 16)$
1	encoder	$(128, 96, 48)$	$(128, 32, 16)$
1	encoder	$(128, 96, 48)$	$(128, 32, 16)$
2	encoder	$(128, 96, 48)$	$(128, 32, 16)$
2	encoder	$(128, 96, 48)$	$(128, 32, 16)$
2	encoder	$(128, 96, 48)$	$(128, 32, 16)$
3	encoder	$(128, 48, 24)$	$(128, 144, 72)$
3	encoder	$(192, 48, 24)$	$(192, 144, 72)$
3	encoder	$(192, 48, 24)$	$(192, 144, 72)$
4	encoder	$(192, 48, 24)$	$(192, 144, 72)$
4	encoder	$(192, 48, 24)$	$(192, 144, 72)$
4	encoder	$(192, 48, 24)$	$(192, 144, 72)$
3	decoder	$(384, 48, 24)$	$(384, 144, 72)$
3	decoder	$(192, 48, 24)$	$(192, 144, 72)$
3	decoder	$(192, 48, 24)$	$(192, 144, 72)$
2	decoder	$(320, 96, 48)$	$(320, 32, 16)$
2	decoder	$(128, 96, 48)$	$(128, 32, 16)$
2	decoder	$(128, 96, 48)$	$(128, 32, 16)$
1	decoder	$(256, 96, 48)$	$(256, 32, 16)$
1	decoder	$(128, 96, 48)$	$(128, 32, 16)$
1	decoder	$(128, 96, 48)$	$(128, 32, 16)$
1	final	$(64, 32)$	
1	final	$(32, C)$	

ScanNet	# parameters:	$\mathbf{761,333}$
Matterport3D	# parameters:	$\mathbf{761,366}$

(a) **ScanNet/Matterport architecture.** We use more filters in the later two mesh levels and the best performing filter ratio from Table 1. We obtain different numbers of parameters for ScanNet and Matterport3D since they differ in their number of semantic classes ($C_{\text{scannet}} = 21$ and $C_{\text{matterport}} = 22$).

#level	level type	module type	filters
1	encoder	edge+BN+ReLU	$2 * (9, 64, 32)$
1	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
1	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	encoder	edge+BN+ReLU	$2 * (128, 64, 32)$
3	encoder	edge+BN+ReLU	$2 * (128, 96, 48)$
3	encoder	edge+BN+ReLU	$2 * (192, 96, 48)$
3	encoder	edge+BN+ReLU	$2 * (192, 96, 48)$
4	encoder	edge+BN+ReLU	$2 * (192, 96, 48)$
4	encoder	edge+BN+ReLU	$2 * (192, 96, 48)$
4	encoder	edge+BN+ReLU	$2 * (192, 96, 48)$
3	decoder	edge+BN+ReLU	$2 * (384, 96, 48)$
3	decoder	edge+BN+ReLU	$2 * (192, 96, 48)$
3	decoder	edge+BN+ReLU	$2 * (192, 96, 48)$
2	decoder	edge+BN+ReLU	$2 * (320, 64, 32)$
2	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
2	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
1	decoder	edge+BN+ReLU	$2 * (256, 64, 32)$
1	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
1	decoder	edge+BN+ReLU	$2 * (128, 64, 32)$
1	final	Lin+BN+ReLU	$(64, 32)$
1	final	Lin	$(32, 13)$

	# parameters:	$\mathbf{728,045}$

(b) **S3DIS architecture.** Unlike the ablation study, we use more filters in the final two mesh levels.

Table 6: **Architectures for benchmarks.** We present two slightly different architectures for S3DIS and ScanNet/Matterport, respectively. This is due to the comparably lower mesh quality of S3DIS.

[2] 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. 2, 6

[3] Christopher Choy, Jun Young Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[5] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[6] 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 *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[7] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical Kernel for Efficient Graph Convolution on 3D Point Clouds. *arXiv preprint arXiv:1909.09287*, 2019. 5

Method	mIoU	mAcc	ceil.	floor	wall	beam	col.	wind.	door	chair	table	book.	sofa	board	clut.
Pointnet [10]	41.1	49.0	88.8	97.3	69.8	0.1	3.9	46.3	10.8	52.6	58.9	40.3	5.9	26.4	33.2
SegCloud [12]	48.9	57.4	90.1	96.1	69.9	0.0	18.4	38.4	23.1	75.9	70.4	58.4	40.9	13.0	41.6
Eff 3D Conv [16]	51.8	68.3	79.8	93.9	69.0	0.2	28.3	38.5	48.3	71.1	73.6	48.7	59.2	29.3	33.1
RSNet [5]	51.9	59.4	93.3	98.4	79.2	0.0	15.8	45.4	50.1	65.5	67.9	22.5	52.5	41.0	43.6
TangentConv [11]	52.6	62.2	90.5	97.7	74.0	0.0	20.7	39.0	31.3	69.4	77.5	38.5	57.3	48.8	39.8
PointCNN [8]	57.3	63.9	92.3	98.2	79.4	0.0	17.6	22.8	62.1	80.6	74.4	66.7	31.7	62.1	56.7
RNN Fusion [15]	57.3	63.9	92.3	**98.2**	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
ParamConv [14]	58.3	67.1	92.3	96.2	75.9	**0.3**	6.0	**69.5**	63.5	66.9	65.6	47.3	68.9	59.1	46.2
MinkowskiNet [3]	65.4	71.7	91.8	98.7	86.2	0.0	34.1	48.9	62.4	89.8	81.6	74.9	47.2	74.4	58.6
KPConv [13]	**67.1**	**72.8**	**92.8**	97.3	**82.4**	0.0	23.9	58.0	69.0	**91.0**	81.5	**75.3**	**75.4**	**66.7**	**58.9**
SPGraph [9]	58.0	66.5	89.4	96.9	78.1	0.0	**42.8**	48.9	61.6	84.7	75.4	69.8	52.6	2.1	52.2
SPH3D-GCN* [7]	59.5	65.9	93.3	97.1	81.1	0.0	33.2	45.8	43.8	79.7	86.9	33.2	71.5	54.1	53.7
HPEIN [6]	61.9	68.3	91.5	98.2	81.4	0.0	23.3	65.3	40.0	75.5	87.7	58.5	67.8	65.6	49.4
DCM Net (**Ours**)	64.0	71.2	92.1	96.8	78.6	0.0	21.6	61.7	54.6	78.9	**88.7**	68.1	72.3	66.5	52.4

Table 7: **Semantic segmentation IoU scores on S3DIS Area 5.** We furthermore provide mean class accuracy scores. Among all approaches, we perform third best only outperformed by KPConv [13] and MinkowskiNet [3]. Among graph convolutional approaches, we clearly report state-of-the-art with a gap of 2.1% to HPEIN [6].

Method	mIoU	mAcc	ceil.	floor	wall	beam	col.	wind.	door	chair	table	book.	sofa	board	clut.
Pointnet [10]	47.6	66.2	88.0	88.7	69.3	42.4	23.1	47.5	51.6	42.0	54.1	38.2	9.6	29.4	35.2
RSNet [5]	56.5	66.5	92.5	92.8	78.6	32.8	34.4	51.6	68.1	60.1	59.7	50.2	16.4	44.9	52.0
PointCNN [8]	65.4	75.6	**94.8**	**97.3**	75.8	63.3	51.7	58.4	57.2	71.6	69.1	39.1	61.2	52.2	58.6
KPConv [13]	**70.6**	79.1	93.6	92.4	**83.1**	**63.9**	54.3	66.1	**76.6**	57.8	64.0	**69.3**	**74.9**	61.3	60.3
SPGraph [9]	62.1	73.0	89.9	95.1	76.4	62.8	47.1	55.3	68.4	73.5	69.2	63.2	45.9	8.7	52.9
HPEIN [6]	67.8	76.3	-	-	-	-	-	-	-	-	-	-	-	-	-
SPH3D-GCN* [7]	68.9	77.9	93.3	96.2	81.9	58.6	**55.9**	55.9	71.7	72.1	**82.4**	48.5	64.5	54.8	60.4
DCM Net (**Ours**)	69.7	**80.7**	93.7	96.6	81.2	44.6	44.9	**73.0**	73.8	71.4	74.3	63.3	63.9	**63.0**	**61.9**

Table 8: **Semantic segmentation IoU scores on S3DIS k-fold.** We furthermore provide mean class accuracy scores. Among all approaches, we perform second best only outperformed by KPConv [13]. Among graph convolutional approaches, we report state-of-the-art with a gap of 0.8% to the concurrent work SPH3D-GCN [7].

[8] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution On X-Transformed Points. In *Neural Information Processing Systems (NIPS)*, 2018. 5

[9] Landrieu Loic and Martin Simonovsky. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[11] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent Convolutions for Dense Prediction in 3D. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

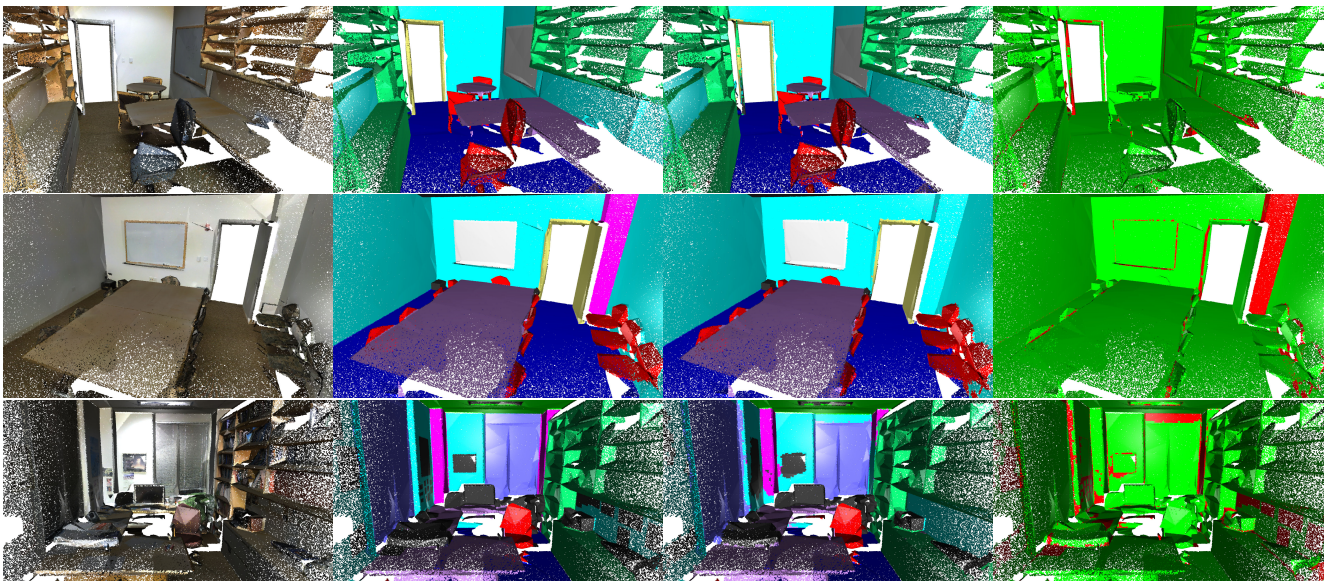
[12] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *International Conference on 3D Vision (3DV)*, 2017. 5

[13] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 5

[14] S. Wang, S. Suo, W.C. Ma, A. Pokrovsky, and R. Urtasun. Deep Parametric Continuous Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[15] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3D Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation. In *IEEE European Conference on Computer Vision (ECCV)*, 2018. 5

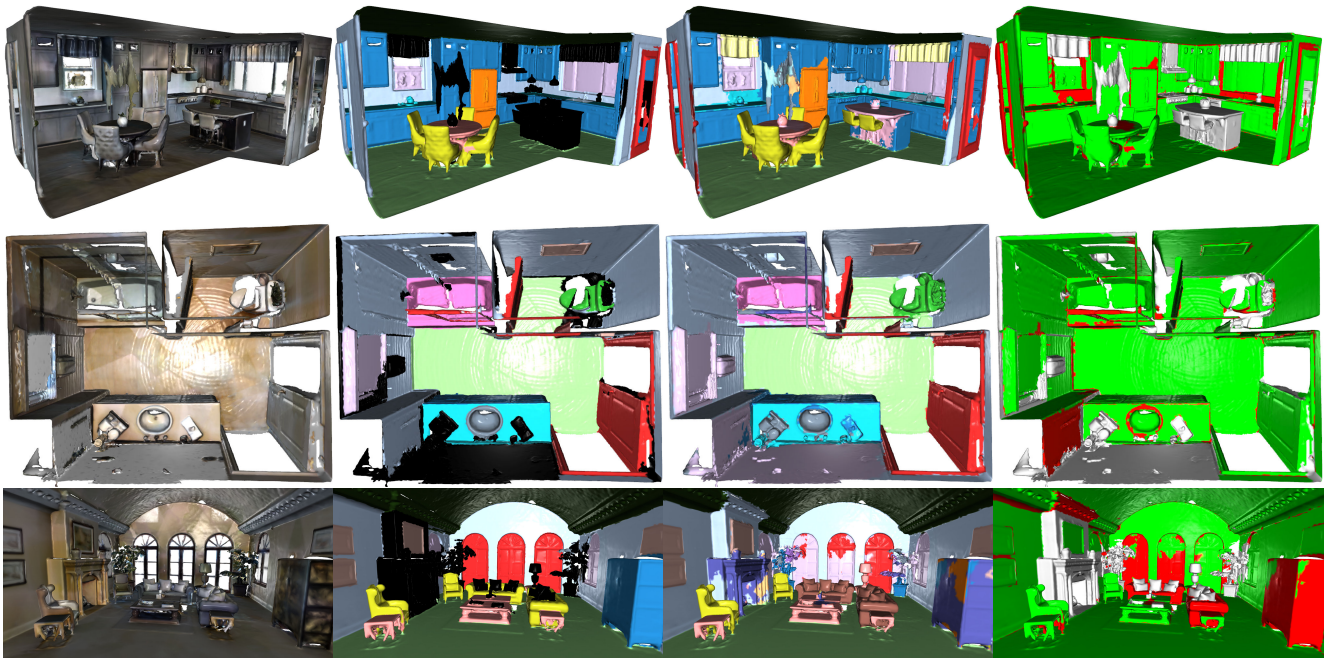
[16] Chris Zhang, Wenjie Luo, and Raquel Urtasun. Efficient convolutions for real-time semantic segmentation of 3d point clouds. In *International Conference on 3D Vision (3DV)*, 2018. 5



Input Point Cloud **Ground Truth** **Prediction** **Error**

ceiling floor wall beam column window door chair table bookshelf sofa board clutter

Figure 3: **Results on Stanford Large-Scale 3D Indoor Spaces [1].** Our method correctly predicts challenging classes such as board, while maintaining clear boundaries for most of the classes. In the second row, our method confuses the similar classes column and wall. In the last example, it becomes evident that our method tends to produce unclear boundaries for diverse clutter regions.



Input Mesh **Ground Truth** **Prediction** **Error**

unlabeled wall floor cabinet chair sofa table door window picture counter desk fridge toilet sink bathtub otherfurn ceiling

Figure 4: **Results on Matterport3D [2].** Our method correctly predicts even unlabeled regions. However, reasonable errors occur, such as confusing windows extending down to the floor as doors. In the last row, our algorithm correctly predicts sofa even though the ground truth is falsely labeled as chair.

ScanNet [8] Test	mIoU	Data Representation					Features
		Points	Voxel	Mesh	2D	Texture	
PointNet [39]	-	✓	-	-	-	-	XYZ-RGB
PointNet++ [40]	33.9	✓	-	-	-	-	XYZ
FCPN [11]	44.7	✓	✓	-	-	-	XYZ-RGB-N
3DMV [9]	48.3	✓	✓	-	✓	-	XYZ-RGB-N
JPBNet [6]	63.4	✓	-	-	✓	-	XYZ-RGB-N
MVPNet [26]	64.1	✓	-	-	✓	-	XYZ-RGB-N
Tangent Conv [48]	43.8	✓	-	-	-	-	XYZ-RGB-N
SurfaceConvPF [20]	44.2	-	-	✓	-	-	XYZ-RGB-N
TextureNet [25]	56.6	-	-	✓	✓	✓	XYZ-RGB-N
PointCNN [33]	45.8	✓	-	-	-	-	XYZ-RGB-N
ParamConv [52]	-	✓	-	-	-	-	XYZ-RGB
MCCN [22]	63.3	✓	-	-	-	-	XYZ-RGB-N
PointConv [56]	66.6	✓	-	-	-	-	XYZ-RGB-N
KPConv [50]	68.4	✓	-	-	-	-	XYZ-RGB
SparseConvNet [17]	72.5	-	✓	-	-	-	XYZ-RGB
MinkowskiNet [7]	**73.4**	-	✓	-	-	-	XYZ-RGB
DeepGCN [31]	-	✓	-	-	-	-	XYZ-RGB-N
SPGraph [34]	-	✓	-	-	-	-	XYZ-RGB
SPH3D-GCN [30]	61.0	✓	-	-	-	-	XYZ-RGB-N
HPEIN [27]	61.8	✓	-	-	-	-	XYZ-RGB-N
DCM Net **(Ours)**	65.8	✓	-	✓	-	-	XYZ-RGB-N

Table 9: **Data representations and input features.** We show the data representation and input features of each approach.