On the Receptive Field of Point Convolutions

Francis Engelmann
engelmann@vision.rwth-aachen.de
Theodora Kontogianni
kontogianni@vision.rwth-aachen.de
Bastian Leibe
leibe@vision.rwth-aachen.de

Visual Computing Institute
Computer Vision Group
RWTH Aachen University
Aachen, Germany

Abstract

In this work, we propose *Dilated Point Convolutions* (DPC) which drastically increase the receptive field of convolutions on 3D point clouds. As we show in our experiments, the size of the receptive field is directly related to the performance of dense tasks such as semantic segmentation. We look at different network architectures and mechanisms to increase the receptive field size of point convolutions and propose in particular *dilated* point convolutions. Importantly, our dilation mechanism can easily be integrated into all existing methods using nearest-neighbor-based point convolutions. To evaluate the resulting network architectures, we visualize the receptive field and report competitive scores on the task of 3D semantic segmentation on the S3DIS and ScanNet datasets.

1 Introduction

The past few years have witnessed a tremendous development of 3D scene understanding methods on several tasks including semantic segmentation, part segmentation and instance segmentation of 3D point clouds. A major driving force behind this success was the development of continuous convolutional layers [21, 26, 27] that can directly operate on unstructured input data such as 3D point clouds. Continuous convolutional layers can be seen as generalizations of discrete convolutions. While the former operate on continuous data, the latter process grid-structured data, such as images or volumetric voxel-grids. From this observation, a lot of acquired knowledge from the discrete image domain can directly be transferred to continuous representations such as point clouds.

An important tool for diagnosing and understanding convolutional neural networks is the concept of receptive fields. The receptive field of a neural unit describes the region of the input data that influences the value of that unit. All input data outside of the receptive field does not contribute to it. Hence, large receptive fields are important, specifically on dense tasks such as semantic segmentation.

Most current network architectures operating on grid-like data implicitly increase the receptive field by using *deep* network architectures which have proven to work well [8, 22, 24]. However, few works directly study the influence of receptive fields in discrete CNNs [14, 16]. So far, there is no work analyzing receptive fields of continuous convolutions on 3D point clouds. Such a study is particularly challenging, since the theoretical size of receptive fields

is difficult to compute due to the non-uniform structure of the 3D point clouds. Nevertheless, the concept of receptive fields is equally important in the continuous domain as it is in the discrete one. As such, we propose to rely on a visualization of the receptive fields to motivate different network architectures and we present a thorough ablation study comparing several strategies which increase the receptive field of continuous convolutions. Specifically, we look at common strategies to increase the receptive field by stacking convolutional layers and using larger kernels. By visually analyzing the extent of the resulting receptive fields, we notice that their influence still remains rather limited. For example, the receptive field of a 7-layer point convolutional network might not be able to cover an object with the size of a chair in a dense point cloud. From these observations, we propose *Dilated Point Convolutions* as a means to further increase the receptive field size of point convolutions.

The paper is structured as follows: In the related work, we discuss current methods for 3D point cloud processing and existing works analyzing receptive fields on discrete convolutions. Then, we review *Point Convolutions* as an instance of continuous convolutions on 3D point clouds. Next, we describe and visualize well established methods for increasing receptive fields, which leads us to the derivation of *Dilated Point Convolutions*. Finally, in the experimental section, we compare the aforementioned strategies.

Contributions. (1) We evaluate most commonly used strategies to increase the receptive fields in current methods using point convolutions. (2) We propose to visualize the receptive field of point convolutions to make educated network design choices. (2) From these observations, we derive *Dilated Point Convolutions* (DPC) as an efficient way to rapidly increase the size of receptive field. (3) Using DPCs we are able to report a new state-of-the-art on 3D semantic segmentation on the S3DIS dataset and competitive scores on the ScanNet dataset.

2 Related Work

2D Projection Representation. Qi *et al*. [18] and Boulch *et al*. [4] project 3D point clouds into 2D representations, then apply 2D convolutional networks and finally fuse the results back into 3D space. These type of projections do not make use of the underlying geometric structure as they only operate on the projected appearance of the point clouds.

3D Volumetric Grid Representation. Song *et al*. [23] and Maturana *et al*. [15] voxelize point clouds into regular volumetric grids and apply 3D convolutions. These approaches are constrained by the fixed resolution of the 3D grid. Coarse grids lead to loss of detail and fine ones are suffering from high memory and computational costs. The use of octrees [20] and kd-trees [10] offer improved grid resolutions. Recently, Graham *et al*. [7] offered a speed- and memory-efficient approach for sparse 3D convolutions which are only performed in occupied voxels. However, sparsely sampled point clouds can still lead to problems as the direct neighborhood is not occupied, hindering the flow of information.

3D Feature Learning on Point Sets. Numerous methods operate directly on 3D point clouds [17, 21, 26, 27]. They follow-up on the seminal work of *PointNet* [17]. PointNet uses multi-layer-perceptrons and a max-pooling layer to learn 3D point features. The max-pooling layer is applied across all the points in the point cloud offering global information but failing to capture local structure. Local structure is implicitly considered in 2D images and 3D voxels due to the use of spatial grids. Filters that incorporate the information of the neighboring points in the grid are then learned. Numerous methods rely on similar types of

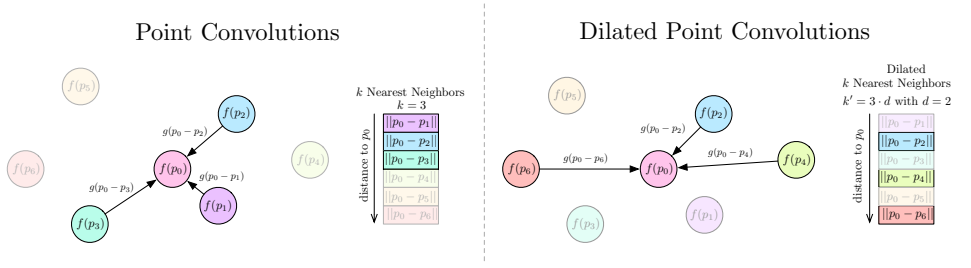

Figure 1: *(Left)* **Point Convolutions.** Schematic illustration of point convolutions. The continuous feature function $f(\cdot)$ assigns a feature value to continuous point positions p. *(Right)* **Dilated Point Convolutions.** We propose *dilated* point convolutions as a simple yet very effective way to increase the receptive field of point convolutions resulting in a significant boost in performance at almost no additional computational cost (see Table 2). Instead of computing the kernel weights $g(\cdot)$ over the k nearest neighbors, we propose to compute the kernel weights over a *dilated* neighborhood obtained by computing the sorted $k \cdot d$ nearest neighbors and preserving only every d-th point.

spatial neighborhoods in an unstructured point cloud: Hua *et al*. [9] computes nearest neighbors on the fly and bins them into spatial cells before using fully convolutional networks. Landrieu *et al*. [11] creates neighborhoods by over-segmenting 3D point clouds into superpoints. However, the most popular method used by [6, 12, 19, 26, 27] consists in computing the k nearest neighbors (KNN) of every point to represent its neighborhood. *EdgeConvs* [26] establish this neighborhood on the feature space while *PointConv* [27], *PointNet++* [19] and *PointCNN* [12] use the spatial coordinates. Engelmann *et al*. [6] use KNN in the feature space and k-means in the world coordinate system to create neighborhoods.

Receptive Field Analysis. Few works systematically study the influence of receptive fields on CNNs [14, 16]. From a practical aspect, deeper networks which stack multiple layers of 2D convolutions have proven to work better [22, 24]. Dilated convolutions (also known as *atrous* convolutions), already used in semantic 2D image segmentation [3], allow to efficiently enlarge the receptive field of filters to incorporate larger context without increasing the number of parameters or the amount of computation. In this paper, we propose a dilated version of the KNN that proves to be very effective.

3 Approach

In this section, we first revise *Point Convolutions* and the importance of large *receptive fields* in the context of point clouds and discuss several strategies to increase the receptive field size of continuous convolutions. Most notably, we propose *Dilated Point Convolutions* as a means of dramatically increasing the receptive field size at comparable computational cost.

3.1 Point Convolutions

We shortly review *Point Convolutions* as an instance of continuous convolutions on 3D point clouds. Here, we present a general form and describe commonly used variations for 3D point

cloud processing. Formally, *continuous* convolutions are defined as:

$$\big(f * g\big)(p_i) = \int_{-\infty}^{+\infty} f(p_j) \cdot g(p_i - p_j)\, dp_j \tag{1}$$

with the continuous feature function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ assigning a feature-value to every D-dimensional position $p_j \in \mathbb{R}^D$ and the continuous kernel function $g : \mathbb{R}^D \rightarrow \mathbb{R}$ mapping a relative position to a kernel weight. Note that the convolution operation is commutative *i.e.* $f * g = g * f$. In most practical applications, the feature function f is not fully known since only a limited number K of point positions p_k are observed. Using Monte-Carlo integration, the continuous convolution can then be approximated as:

$$\big(f * g\big)(p_i) \approx \frac{1}{K} \sum_{k=1}^{K} f(p_k) \cdot g(p_i - p_k) \tag{2}$$

where the infinite kernel function $g(\cdot)$ is approximated using a learned parametric function, commonly implemented as a multi-layer perceptron (MLP):

$$g(p; \theta) = \mathrm{MLP}(p; \theta) \tag{3}$$

where p is the relative position and θ is a set of learned parameters.

Point Convolutions in Recent Work. The above definition of continuous convolutions is used in Wang *et al*. [21]. *PointConv* [27] additionally proposes to weight the kernel function using the inverse local density to compensate for the non-uniform distribution of point samples. In *SpiderCNN*, Xu *et al*. [28] propose to replace the MLP by a combination of step functions and Taylor expansions to capture rich spatial information. A broader interpretation of continuous convolutions is used in *EdgeConv* [26] where the kernel function $g(\cdot)$ is not only defined over relative positions but additionally over the difference of learned point features.

Local Filters using KNN. To guarantee *local* filters, discrete convolutions rely on a restricted kernel sizes *e.g.* 3×3 or 5×5 pixels on a 2D grid. With continuous convolutions, this effect is obtained by limiting the cardinality k of the local kernel support-domain. All previously mentioned methods, including *PointCNN* [12], rely on k-nearest neighbors to enforce local kernels. This is an important observation since we rely on this mechanism to propose *dilated* continuous convolutions (see Section 3.2 and Figure 1, right).

Importance of Receptive Fields. The receptive field, or *field of view*, of a neural unit inside the network describes a region of the input point cloud that influences that unit. In the context of 3D semantic segmentation, the task is to assign a semantic label to each point in a given scene. The final decision on the label for a particular point is influenced only by points inside the receptive field of that particular point, all other points outside the receptive field do not contribute to the decision, see Figure 3. It is thus essential to design network architectures with receptive fields large enough to cover relevant context around each point.

Increasing the Receptive Field. Inspired by CNN architectures, the receptive field of point convolutions can be increased by *stacking* more convolutional layers. *EdgeConvs* [26] stack 3 convolutional layers, *SpiderCNN* [28] use 4 layers and *PCCN* [21] use 8. In this work, we experiment with 3, 5 and 7 stacked convolutional layers, see Table 1.

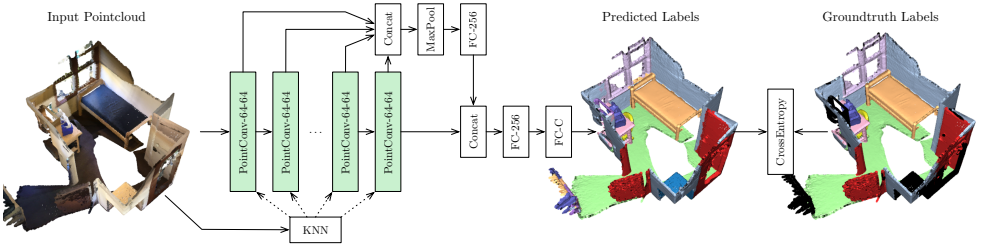

Figure 2: Model built by stacking *Point Convolutions* used in our experiments on the task of 3D semantic segmentation. Each point is assigned one out of C semantic labels.

Another common technique from the field of discrete convolutions consists in increasing the *kernel size*. In the setup of point convolutions this effect is achieved by selecting a larger number k of nearest neighbors. Note however, that this does not increase the number of model parameters since the kernel weights are computed over relative point positions using the parametric kernel function $g(\cdot)$ (see Table 1). This is in stark contrast to discrete convolutions over grids where a larger kernel generally means more parameters.

Another popular technique from deep neural networks is *pooling*. Most recent point networks rely on *Farthest Point Sampling* (FPS) as used in PointNet++ [19] *e.g.* *PointConv* [27] uses 2 FPS-pooling layers. In this work, we explicitly do not use pooling. Instead, we focus on another technique: *Dilated Point Convolutions*.

3.2 Dilated Point Convolutions.

In this section, we propose *Dilated Point Convolutions* (DPC). DPCs correspond to Point Convolutions (PC), however, they differ in the way they select neighboring points: While PCs directly use the k nearest neighbors, DPCs first compute $k \cdot d$ nearest neighbors and then select every d-th neighbor, see Figure 1 (right). Note that for $d = 1$, DPCs are identical to regular PCs. This mechanism results in a vastly increased receptive field (Figure 3), however the number of parameters remains unchanged. A sublinear computational overhead results from the larger number of neighbors that needs to be computed, see Table 2. Another positive aspect about DPC is that they can directly be added – with minimal effort – into all existing point convolutional methods relying on k nearest neighbor search.

4 Experiments

Model Architecture. In all our experiments, we use a deep convolutional model as depicted in Figure 2. The main branch (shown in green) consists of stacked (Dilated) Point Convolutions. The k nearest neighbors (KNN) for each point are computed on-the-fly. The final point features are concatenated with global features obtained by max-pooling over the concatenated point features at different depth-levels. This is analogous to *EdgeConvs* [26]. Note that PCCN [21] use residual skip connections between convolutional layers. It remains to be analyzed which skip connections work better.

Task and Metrics. We evaluate our model on the task of 3D semantic segmentation. The goal is to predict a semantic label for each point in a given point cloud. This task is especially well suited to analyze the effectiveness of larger receptive fields since the label of each point

is only influenced by points in its receptive field. We adopt the commonly used metrics: mean intersection over union (mIoU), mean class accuracy (mAcc) and overall accuracy (oAcc).

Stanford Large-Scale 3D Indoor Spaces (S3DIS) [1] contains dense 3D point clouds from 6 large-scale indoor areas consisting of 271 rooms from 3 different buildings. The points are annotated with 13 semantic classes. We follow the common train/test split as used in [1, 21, 25] and train on all areas except Area 5 which we keep for test.

ScanNet v2 [4] contains 3D scans of a wide variety of indoor scenes, including apartments, hotels, conference rooms and offices. The dataset contains 20 valid semantic classes. We use the public training, validation and test split of 1201, 312 and 100 scans, respectively.

Training Details. We train our networks using the Adam Optimizer and exponential-decay learning-rate scheduling. During training we randomly sample 4092 points from blocks of 3 m side length. Points are sampled without replacement and we use zero-padding if there are less than 4092 points. We adopt this strategy to make sure the nearest neighbors are always distinct from the actual point. Note that this differs from most methods based on the reference implementation of the seminal works *PointNet* [17] and *PointNet++* [19]. Here point clouds are divided into rectangular blocks with a side length of either 1 m or 1.5 m. This setup is beneficial for PointNets which rely on global max-pooling over all points in a block to compute point features: the smaller the block size, the more details can be encoded in the point features. This effect is demonstrated in [5] which compute multi-scale features using PointNets over multiple block sizes. However, this artificial constraint becomes problematic as soon as the size of the receptive field exceeds the size of the blocks.

4.1 Results and Discussion

We perform extensive ablation studies on the previously introduced mechanisms for increasing the receptive field in Table 1 and Table 2. The ablation studies are performed on Area 5 of the S3DIS dataset [1] due to its moderate size, allowing for a faster iteration cycle.

Depth and Number of Neighbors k (Table 1). Similar to grid-convolutions, deeper point convolutional network perform better when stacking more layers. Equally, the performance increases with the number of neighbors. However, increasing the number of neighbors increases the computational cost, resulting in slower networks. Similarily, increasing the number of convolutions (going deeper) leads to additional memory consumption.

Dilation Factor d (Table 2). Dilated Point Convolutions are an efficient tool to rapidly increase the receptive field of convolutions, see Figure 3. The improved performance on the semantic segmentation tasks show indeed that a larger receptive field is important. However, the rapidly increasing receptive field resulting in large receptive fields in later layers is also responsible for sparsely sampled neighborhoods in earlier layers. We assume that this makes it harder for the network to learn high-frequency or local features. In future work, it could be interesting to investigate deep convolutional networks using Dilated Point Convolutions with a dilation rate d that increases with the depth of the network. Intuitively, such a network could learn localized signals in the earlier stages and higher level information at later stages.

Number of Point Convs	Number of Neighbors k	Time per Forward-Pass	Number of Parameters	mIoU	mAcc	oAcc
3	5	12.10 ms	$402 \cdot 10^3$	50.04	57.42	**85.01**
3	10	13.64 ms	$402 \cdot 10^3$	50.98	58.16	84.74
3	20	17.65 ms	$402 \cdot 10^3$	**52.25**	**60.83**	84.69
5	5	14.53 ms	$625 \cdot 10^3$	52.69	58.87	**85.33**
5	10	17.12 ms	$625 \cdot 10^3$	52.91	59.57	85.27
5	20	23.35 ms	$625 \cdot 10^3$	**53.27**	**60.15**	85.15
7	5	16.99 ms	$880 \cdot 10^3$	52.93	59.87	**85.62**
7	10	20.68 ms	$880 \cdot 10^3$	53.57	60.92	85.59
7	20	29.38 ms	$880 \cdot 10^3$	**53.93**	**61.73**	85.58

Table 1: **Ablation Study - Point Convolutions.** (Test Dataset: S3DIS - A5) Comparing number of convolutions, neighbors, forward-pass time, parameters and semantic segmentation scores. Deeper networks with more neighbors work better. However, they need more memory and processing time. Note that the overall accuracy (oAcc) is of limited usefulness as semantic classes with numerous points (floor, wall) are favored of smaller objects.

Number of Point Convs	Number of Neighbors k	Time per Forward-Pass	Number of Parameters	Dilation d	mIoU	mAcc	oAcc
7	20	29.38 ms	$880 \cdot 10^3$	1	53.93	61.73	85.58
7	20	31.57 ms	$880 \cdot 10^3$	2	55.83	61.76	85.68
7	20	35.36 ms	$880 \cdot 10^3$	8	**61.28**	**68.38**	**86.78**
7	20	51.65 ms	$880 \cdot 10^3$	16	58.79	65.84	86.41

Table 2: **Ablation Study - Dilated Point Convolutions** (Test Dataset: S3DIS - A5) Comparing different dilation factors d. Using dilation, the receptive field can be increased significantly (Figure 3) at constant memory requirements and marginal increment in processing time. Initially, the segmentation performance increases with the dilation factor d. However, it decreases after a certain point, see Section 4.1 for a discussion.

Model Size. Note that, since the kernel function $g(p)$ is defined over relative positions p, the number of trainable parameters is independent of the number of neighbors k (and the dilation factor d). As such, increasing the number of neighbors k (or the dilation factor d) increases the receptive field without additional memory requirements.

Competing Approaches. We report scores of our best performing model on S3DIS [1] and are able to outperform other recent methods by a significant margin, see Table 3. We also evaluated the model on ScanNet v2 [4] and we obtain 59,45 mIoU on the validation set.

Quantitative Results. In Figure 4, we show qualitative results on the ScanNet validation dataset. Additional qualitative results both S3DIS and ScanNet can be found in the supplementary material.

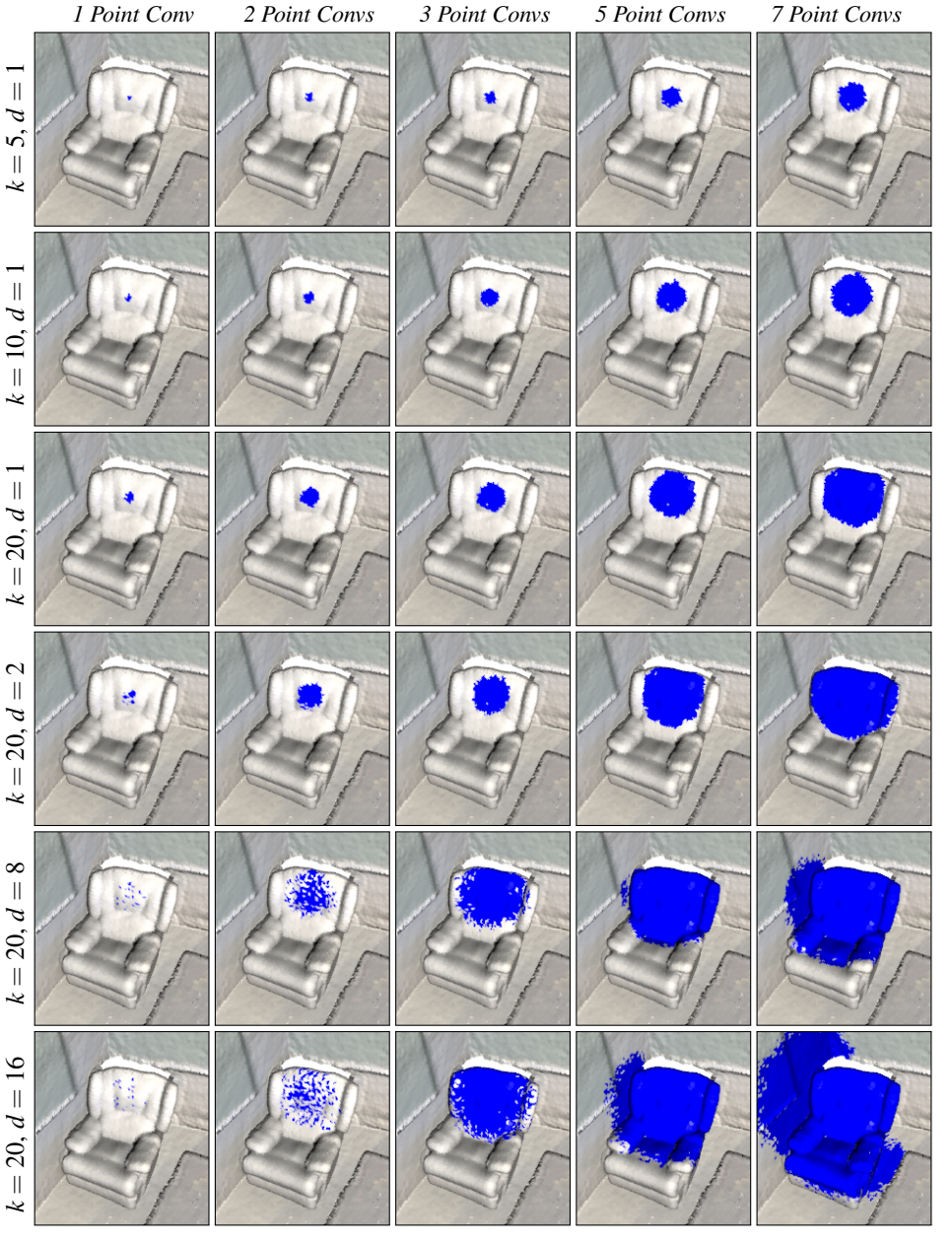

Figure 3: **Receptive field** visualized in blue for different network architectures using an increasing number of *Point Convolutions* (columns) and increasing kernel sizes (rows) based on the number of nearest neighbors k and dilation factor d. The receptive field sizes of point convolutions without dilation ($d = 1$) are substantially smaller than convolutions with dilation. However, for large dilations, *e.g.* $d = 16$ the receptive field is sparse in early stages of the deep network (bottom left).

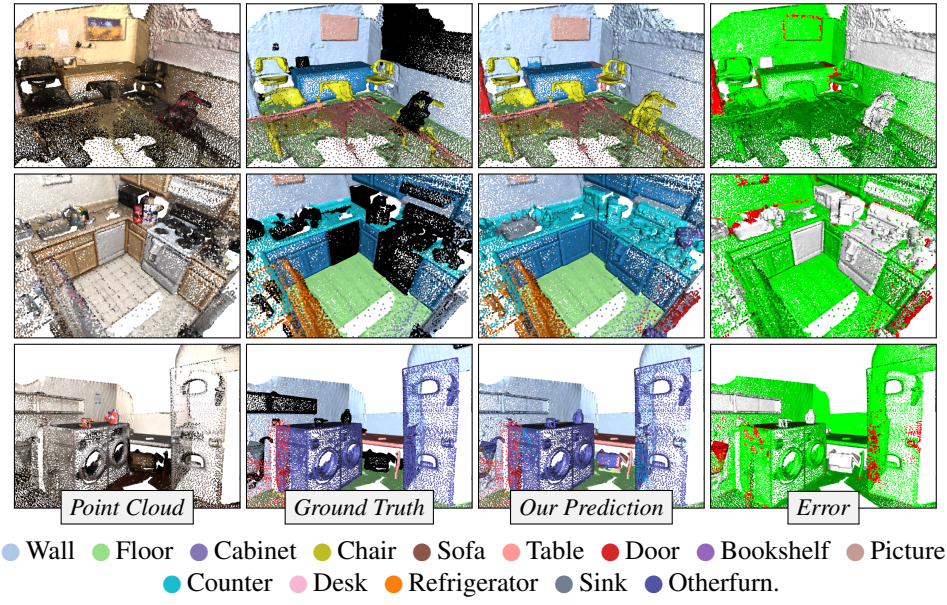

Figure 4: **Results of our method on ScanNet v2 dataset [4] validation.** Left to right: Input RGB point cloud, semantic segmentation ground truth, semantic segmentation prediction and the error where green shows correct predictions, red shows wrong predictions and white indicates that no ground truth label is available. More examples can be found in the appendix.

5 Conclusion

In this work, we presented several mechanisms to increase the receptive field size of 3D point convolutions. Specifically, we have proposed *Dilated Point Convolutions* as an intuitive and effective technique to increase the receptive field size of point convolutions. As a result, we were able to report remarkable improvements over well-known baseline methods. More importantly, our dilation mechanism is straightforward to incorporate into existing point convolutional networks based on KNN. We hope the community and fellow researchers can develop better performing algorithms using these insights.

Method	mIoU	mAcc	oAcc
PointNet [17]	41.1	49.0	-
Engelmann et al. [6]	52.2	59.1	84.2
PointCNN [12]	57.3	63.9	85.9
SPG [13]	58.0	66.5	86.4
PCNN [21]	58.3	67.0	-
DPC (Ours)	**61.28**	**68.38**	**86.78**

Table 3: **Semantic segmentation scores on S3DIS (Area 5).**

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] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics*, 2017.

[3] 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. *CoRR*, abs/1606.00915, 2016.

[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.

[5] Francis Engelmann*, Theodora Kontogianni*, Alexander Hermans, and Bastian Leibe. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In *IEEE International Conference on Computer Vision Workshops (ICCV'W)*, 2017.

[6] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In *IEEE European Conference on Computer Vision (ECCV'W)*, 2018.

[7] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Point-wise convolutional neural network. *CoRR*, abs/1712.05245, 2017.

[10] Roman Klokov and Victor Lempitsky. Escape from cells: Deep dkd-networks for the recognition of 3d point cloud models. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[11] Loic Landrieu and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. *arXiv preprint arXiv:1904.02113*, 2019.

[12] 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.

[13] 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.

[14] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016.

[15] 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)*, 2015.

[16] Dmytro Mishkin, Nikolay Sergievskiy, and Jiri Matas. Systematic evaluation of convolution neural network advances on the imagenet. *Computer Vision and Image Understanding*, 161:11–19, 2017.

[17] 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.

[18] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. *CoRR*, abs/1604.03265, 2016.

[19] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Neural Information Processing Systems (NIPS)*, 2017.

[20] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning Deep 3D Representations at High Resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[23] S. Song, A. Khosla, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] C. Szegedy, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[25] 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.

[26] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *Computing Research Repository CoRR*, abs/1801.07829, 2018.

[27] Wenxuan Wu, Zhongang Qi, and Fuxin Li. PointConv: Deep Convolutional Networks on 3D Point Clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters. In *IEEE European Conference on Computer Vision (ECCV)*, 2018.