

Classification and Segmentation of 3d Point Clouds based on deep learning



By

Rabbia Hassan

2018-NUST-MS-CS-00000277304

Supervisor

Dr. Muhammad Shahzad

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters
in Computer Science (MS CS)

In

School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(January 2022)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Classification and Segmentation of 3d Point Clouds based on deep learning" written by RABBIA HASSAN, (Registration No 00000277304), of SEECs has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: _____ *M. SHAHZAD* _____

Name of Advisor: Dr. Muhammad Shahzad _____

Date: _____ **21-Dec-2021** _____

Signature (HOD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Approval

It is certified that the contents and form of the thesis entitled "Classification and Segmentation of 3d Point Clouds based on deep learning" submitted by RABBIA HASSAN have been found satisfactory for the requirement of the degree

Advisor : Dr. Muhammad Shahzad

Signature: M. SHAHZAD

Date: 21-Dec-2021

Committee Member 1: Dr. Muhammad Moazam Fraz

Signature: M. Moazam Fraz

23-Dec-2021

Committee Member 2: Dr. Asif Ali

Signature: Asif Ali

Date: 22-Dec-2021

Signature: _____

Date: _____


Dedication

I dedicate this dissertation to my *parents* and *grand parents* who raised me to be independent and self-actualized. I also dedicate it to three renowned authors *Stephen R. Covey*, *Viktor Frankl* and *Yasmin Mogahed* whose books helped me learn the concepts of life.

Certificate of Originality

I hereby declare that this submission titled "Classification and Segmentation of 3d Point Clouds based on deep learning" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: RABBIA HASSAN

Student Signature: 

Acknowledgments

The highest gratitude and praises to ALLAH Almighty who is Exalted in the Wisdom and might. We are truly honored and humbled for being the recipients of His utmost compassion and mercy and for all the directions and pathways He creates for us out of seemingly impossible things. In His name do we begin, and in His name do we end.

First and foremost, I would like to express my immense gratitude to my supervisor **Dr. Muhammad Shahzad** for his precious time, mentoring, guidance and computational resources. He, with his profound domain knowledge and far reaching perspective, enriched this work at each and every step. The extremely technical and insightful discussions that I had with him throughout during the course of thesis, not only helped me accomplish this current research, but it also taught me how to study concepts in greater depth for a greater impact. I am truly indebted to him for his superlative contribution to both my thesis and academic learning in general.

I extend my gratitude to my Co-supervisor **Dr. Asif Ali** for his guidance, mentoring and time. His precious tips about implementation always helped me think out of the box and his constructive criticism was profoundly helpful in laying the ground work for the research especially during initial half.

I would like to thank my honorable GEC member **Dr. Muhammad Moazam Fraz** for being a great source of inspiration with his brilliant research work and academic excellence, which motivates all the students including me to push the envelope and strive hard for better.

I am extremely grateful to my teachers **Dr. Muhammad Shahzad, Dr. Asif Ali, Dr. Imran Mehmood, Dr. Omer Arif, Dr. Safdar Abbas** and **Dr. Ali Tahir** for adding a lot of value to my MS degree with their brilliant teaching.

A special thanks to my sister **Raa'na Hassan** for all the academic guidance and support particularly during coursework, and my friend **Saim Abdullah** for his crucial tips about the write-up.

A mandatory thanks to my colleagues at Seecs **Omer, Danyal, Anum Asif, Quratul-Ain** and **Arooj** who always lent their support and swapped countless duties with me so that I could prioritize my academic deadlines besides job.

I would also like to thank ITS support department Seecs for all the efforts they put in to ensure our seamless access to the computational resources particularly during quarantine.

Contents

1	Introduction	1
1.1	what is a point cloud?	1
1.2	Deep learning on Raw pointcloud	2
1.3	Background and Motivation	2
1.4	Proposed Solution	3
1.5	Thesis Structure	4
2	Related Work	5
2.1	Volumetric Methods	5
2.2	Graph based Methods	6
2.3	Point based Methods	6
2.4	Methods based on attention mechanisim	7
3	Methodology	8
3.1	Building Blocks	8
3.1.1	Farthest point sampling	9
3.1.2	Annular convolution	9
3.1.3	Pooling	12
3.2	Proposed Architecture	13
3.2.1	Residual Block	15
3.2.2	Skip Connection	16

4 Experiments, Results And Analysis	20
4.1 Point cloud classification	20
4.1.1 Synthetic Data	20
4.1.2 Real world Data	23
4.2 Point cloud Segmentation	25
4.3 Qualitative Results	27
5 Ablation Study and Discussion	29
5.1 Ablation Experiments on Modelnet40 dataset	29
5.2 Testing with various configurations	30
6 Conclusion And Future Work	33
6.1 Conclusion	33
6.2 Future Work	33
6.2.1 Adaptive Adjustment of Neighborhood	33
6.2.2 3d Convolution	34
6.2.3 Weighted Aggregation	34
6.2.4 Use of Additional Features	34
6.2.5 To further refine the neighborhood	34
References	36

List of Abbreviations and Symbols

Abbreviations

FPS	Farthest Point Sampling
PointADNet	Annular Convolution Based Deep Residual Network
ACNN	Annularly Convolutional Neural Network
GCN	Graph Convolutional Neural Network
KNN	K Nearest Neighbors
BN	Batch Normalization
RELU	Rectified Linear Unit
RELU-GN	RELU Group Normalization
FC layers	Fully Connected Layers
MLP	Multilayer Perceptron
mIOU	mean Intersection of Union

Symbols

All the symbols have been explained in the respective sections.

List of Figures

3.1	Graphical illustration of annular convolution. Given a query point q_j , constraint based KNN search fetches K nearest neighbors i.e. $\{N_1, N_2, \dots, N_k\}$ on the rings. Given normal n_j corresponding to query point q_j , it projects the neighboring points onto a tangent plane to calculate the projections of neighboring points denoted by $\{O_1, O_2, \dots, O_k\}$. In the next step, these projections are used to order neighboring points in counterclockwise direction as per the reference direction \mathbf{c} . Lastly convolution is performed with the kernels of size 1×3 to abstract per point features.	10
3.2	Dilated rings grow in size much rapidly as compare to concentric rings due to empty spaces in between. In the deeper layers, the neighbours extracted on a concentric rings are contextually more meaningful being closer to the query points. Each ring is characterized by inner radius i.e. R_i and outer radius i.e. R_o . The term concentric points towards the common centre of each ring i.e. the query point.	11
3.3	The proposed Architecture. N, N_1 and N_2 represent the input to the first, second and third block respectively (where $N > N_1 > N_2$). Each block of annular convolution comprises of two rings with K_1 and K_2 number of neighbors in ring1 and ring2 respectively. I_{N_3} represents the indices of N_3 points of block-three. N'_3 denotes the previously computed features of points with I_{N_3} indices. C and M are the number of classification and segmentation classes.	14

LIST OF FIGURES

3.4	Simplified form of the building blocks with residual learning.Two skip connections indicate the propagation of features from Ring1 and Ring2 of the annular layer in block-one to the Ring1 and Ring2 of the annular layer in block-three.	16
3.5	The Shortcut schematics: (3.5a) Original Resnet,(3.5b) Res-RGSNet, (3.5c) Ours	17
4.1	Pre processing pipeline of ScanObjectNN dataset.	23
4.2	Visualization of objects belonging to OBJ-ONLY variant of ScanObjectNN dataset.	25
4.3	Qualitative results of part segmentation w.r.t ShapeNet-part dataset.Input and Ground Truth are plotted with around 2500 points per point cloud and predictions with 2048 points per point cloud.	27
4.4	Overall Accuracy of existing state of the art networks of point cloud classification on Modelnet40 dataset.The input to all these networks is raw pointclouds with 1024 points representing xyz coordinates.	27
4.5	Overall Accuracy of existing state of the art networks of point cloud classification on ScanobjectNN dataset.The results on real world dataset clearly lags behind their synthetic counterpart.	28
4.6	Comparison of state of the art methods on Shapenetpart dataset for part segmentation with mIOU as evaluation metric.	28

List of Tables

4.1	Classification results on ModelNet40 dataset. AAC is accuracy average class, OA is overall accuracy.xyz means 3d coordinates and norms means surface normal vector.	22
4.2	Classification accuracy on ScanObjectNN dataset.	25
4.3	Part segmentation results (instance mIOU %) on shapenet part dataset.	26
5.1	Ablation experiments of our proposed architecture on ModelNet40 dataset to reinforce the importance of proposed architectural components. Where ,AAC denotes the accuracy average class and OA denotes overall accuracy.	30
5.2	Results of experiments with CRELU,Relu only pre-activation and full pre-activation on Modelnet40 dataset.	30
5.3	The comparison of accuracy w.r.t the depth and number of skip connections.Each block processes 512,128,64,32 and 16 points respectively.Number of points can not be decreased beyond it due to hierarchical down sampling nature of the network.These experiments are done on MODELNET40 dataset where AAC denotes the accuracy average class and OA denotes the overall accuracy.	31
5.4	In this experiment,we replace the pooling layer of each block with different pooling strategies to see its impact.Max pooling outperforms all the tested techniques and hence it is kept as a final design choice in the architecture.Training and testing is conducted using MODELNET40 dataset.	31

5.5	In the first experiment,we merge the annular rings and then apply attention mechanism to assign the attention weights to neighbors in unified ring.Let’s say $N(x_i)$ denotes the K nearest neighbors of a query point x_i and together they form a group.We apply attention mechanism same as [39] to let the network assign scores to the group members as per their significance which is captured in the form of context vector.Likewise, in the second experiment attention mechanism is applied to annular rings but both the experiments evidently give underwhelming performance.	31
5.6	Comparison of accuracy in terms of different number of MLPs used post annular convolution.	32
5.7	Comparison of accuracy in terms of different number of Neurons in fully connected layers.The accuracy tends to drop due to over fitting as number of neurons increase.To manage the computational load of increasing number of neurons batch size has to be decreased accordingly.	32

Abstract

Analysis of point clouds through deep convolutional neural networks is an active area of research due to their massive real-world applications including autonomous driving, indoor navigation, robotics, virtual/augmented reality, unmanned aerial vehicles and drones technology. However, to capture fine grained geometric and semantic properties for the underlying recognition task with raw point cloud is exceedingly challenging due to their irregular and unordered nature, sparsity and lack of implicit neighborhood. In this paper, we have introduced a deep, hierarchical, 3d point based architecture to address the highly challenging problem of object classification and part segmentation using raw point cloud. The proposed architecture consists of multiple layers of Sampling, Annular convolution and Pooling, cascaded together in accordance with the principle of deep residual learning. In the skip connections of our deep residual design, we propose to use a combination of linear Projection shortcut and nonlinear Relu group normalization shortcut with batch normalization, to improve both the optimization landscape and representational power. Our network achieves on par or even better than state of the art results on synthetic and real-world benchmark datasets of object classification i.e. MODELNET40 and ScanObjectNN and part segmentation i.e. ShapeNet-part.

Keywords: *Point clouds, residual learning, group normalization, batch normalization*

CHAPTER 1

Introduction

1.1 what is a point cloud?

Pointclouds also known as point sets are irregular and unordered collection of points scattered in 2D or 3D space.

Mathematically:-

$$P = \{P_i | i = 1, 2, \dots, n\}$$

where P_i represents a 3d point comprises of x,y and z coordinate.it can also include added features such as normals, RGB or color values etc depending upon the nature of the task and n represents the number of points in a pointcloud i.e. its size.

Being comprised of only raw coordinates, pointclouds form a simplest and fundamental representation of 3d shape. The remarkable advancement in sensing technology such as Kinect, Google Tango,LIDAR, MEMS sensors and RGB-D cameras has made point clouds readily available as a dense representation of the real world. Both of the above mentioned factors together with the availability of well-defined and richly annotated benchmark datasets e.g. MODELNET[48], SHAPENET [79] and SEMANTIC3D [80] have led to the rise in interests of researchers to directly process 3d point clouds in the applications related to computer vision and graphics.

1.2 Deep learning on Raw pointcloud

Although deep learning has managed to bring impressive results on 1D and 2D data, yet its adaptation to perceptual tasks related to 3d point clouds is a fiercely challenging problem. Standard deep neural networks consume input with a regular structured format such as 2d images, multi view images and volumetric grids etc. while point cloud is inherently irregular and sparse in nature. Moreover, the lack of implicit notion of neighborhood relationship between points in the space in case of point clouds adds to the difficulty of defining convolution operator which is the real essence of conventional neural networks. One obvious solution is to transform raw point cloud into an intermediate structured representation e.g. Multiview images [81,82,85,83,84] or volumetric grids [9,10,11,12,13,86,88,92] and then process them using Multiview CNNs and volumetric/grid CNNs respectively. These transformations, however, not only incur huge computational cost and memory expense but also lead to the loss of inherent geometric info embedded in raw point clouds, which obstructs the performance of neural architectures by making it difficult for them to capture fine grained features.

1.3 Background and Motivation

As point clouds are often sampled with non-uniform densities therefore multiscale architectures such as pointnet++[2] learn local features at various contextual scales and then progressively formulate a global signature of each point. Although this approach elegantly addresses the issues incurred by variation in density of input such as the inability of network to learn fine grained contextual features from under sampled input and the difficulty in generalization of features learnt in denser regions to sparsely sampled regions. But at the same time, it redundantly includes the neighboring points of one scale in the other which prevents the network from learning the discriminative features and hence reduces the performance. To address the afore mentioned limitation Komarichev et al. [32] proposed annular convolution operator which uses multi ring strategy to avoid the duplication of neighboring points at various scales. To wisely utilize the notion of scales, they apply constraint based KNN search within the region spanned in different rings to capture unique neighboring points. It further refines the neighboring relation between points by ordering them using surface normals and then performs convolution

on ordered neighbors using kernels of arbitrary size. Inspired by its exquisite properties such as invariance to orientation of local patches, ability to adapt to geometric variability and robustness towards the direction of surface normals etc. [32] we use annular convolution to encode the local neighborhood features.

1.4 Proposed Solution

To this end, we propose a deep, hierarchical 3d point based architecture for point cloud classification and segmentation which exploits deep residual learning [33] to integrate features at various levels/blocks. The proposed architecture processes pointsets in hierarchical manner to capture fine grained contextual geometric information in local regions. It extracts local features from smaller neighborhoods in various blocks using annular convolution [32] and then group them together into larger units and process them further to formulate higher level features. The architecture includes skip connections to deal with the infamous vanishing gradient problem. Instead of using linear projection shortcuts as in [33], we formulate the shortcuts for the skip connections in our deep architecture by adding the linear projection shortcut and nonlinear Relu group normalization shortcut [46] followed by batch normalization to assist in optimization further by stabilizing the gradient behaviors. This design reaps the benefits of both the inherent linear characteristics of projection shortcut and better representational power induced by RG shortcut and hence boosts the performance. The key contributions of the proposed work are summarized as follows: -

- The proposed Annular convolution based Deep Residual Network (PointADNet) architecture comprises of cascaded combination of sampling, annular convolution and pooling layers to learn and aggregate the geometric point features.
- To exploit the residual learning, skip connections with nonlinear shortcuts have been designed in accordance with the requirements of hierarchical networks in which points are down sampled while feature space expands as it gets deeper.
- It is showed through experimentation that our proposed architecture demonstrates comparable performance to existing state of the art approaches on three benchmark datasets including MODELNET40 [48], ScanObjectNN[52] and ShapeNet-part [61] datasets.

1.5 Thesis Structure

The thesis is structured as follows.

- Chapter 2: Related work
- Chapter 3: Methodology.
- Chapter 4: Experiments, Results and Analysis.
- Chapter 5: Ablation Study.
- Chapter 6: Conclusion and Future Work.

Related Work

2.1 Volumetric Methods

3d data is available in so many significantly different formats ,structured formats such as multi view images and voxelized volumes and unstructured formats like point cloud are few to mention.With the remarkable success of deep learning methods for 2d and 1d data the quest to apply it for3d data came into being.One of the renowned way followed by voxnet[9] and subsequent architectures [10,11,12,13] is to convert 3d point cloud into volumetric occupancy grids and advocate the use of 3d convolution for feature learning.However voxel resolution is an inevitable parameter which makes these methods computationally intensive. To cater for this issue Gernot et al.[86] proposed OctNet. Essentially built on sparsity property of point cloud, OctNet constructs octrees of occupied voxels along a regular grid. The proposed data structuring decreases the memory footprint and manages computation somewhat but still its hard to keep the data granularity intact using volumetric approaches.Qiangneg et al.[59] introduced Grid-GCN which benefits both from the computational efficiency of point based methods as well as effective data structuring of volumetric methods. The processing of points through proposed GridConv layers is twofold. After voxelizing the Input space it computes group centers and neighboring node points and then projects them onto to a graph for context aggregation which greatly facilitates learning by capturing the edge relations between group centers and neighboring nodes.

2.2 Graph based Methods

GCNs have created a recent surge in the point cloud recognition tasks. Being based on graph formalism they offer a quite elegant way to capture geometric properties of non-Euclidian points. The underlying graph convolution method segregates GCNs into two categories. Spatial graph convolution operates on local neighborhood and tends to learn a node's features based on its neighboring node's features. Being a local operation it can conveniently share kernel weights across different locations. Edge conditioned convolution (EEC) proposed by Simonovsky et al.[14] was a breakthrough work in this domain which exploited edge labels as information channel with an effect same as that of rotational invariance enforced by regular convolution on images. Subsequent noteworthy architectures [15,16,17,18,19,20,21] exploited different renditions of spatial graph convolution with interleaved spatial graph pooling layers to coarsen graph formalism into high level representations.[26,27,28,29,30,31] uses Spectral graph convolution which characterizes convolution as spectral filtering between signals on the graph and eigenvectors of Laplacian matrix[23,24]. Spectral approach has to process complex graphs with billions of nodes and edges simultaneously which restricts it to take the advantage of parallel processing. It together with the signal transformation across difference domains entails high computational complexity [25].

2.3 Point based Methods

Pointnet[1] is a seminal paper which applies deep learning techniques on raw point cloud. It processes points from metric space individually by passing them through consecutive layers of multi-layer perceptron due to which it fails to capture local context. Pointnet++[2] addresses this issue by processing points in hierarchical manner at different scales. However, it tends to learn redundant features due to the overlapping nature of different scales. ACNN[32] alleviates this problem by introducing annular convolution which restricts the inclusion of features learnt at one scale in the other by imposing ring shaped local regions. Our deep residual architecture exploits annular convolution to learn the local geometric features. However the dilated rings proposed by ACNN have empty spaces between them which leads to wastage of the region lying distance wise closer to the query point. Due to which the neighbors captured from the last ring lie far

away from the query point and hence context is compromised. Our proposed method makes use of this empty space and captures the neighborhood in a better way. There are various other noteworthy architectures which applied deep learning on point cloud. PointCNN[4] exploits canonical ordering of points using operator X-conv to weight the input points and features and then process them using conventional method of convolution. Wenxuan et al.[5] proposed an operation PointConv which uses relative positions of points as input and projects them as weight to convolution using MLP. Pointweb[6] explores the relationship among points in local neighborhood by densely connecting each point with the other which facilitates a point to learn features from all other points. Qingyong et al.[7] introduced a method to process large scale point clouds which uses random sampling to find query points. However random sampling is susceptible to drop important points so to cater for this problem they propose two powerful modules named as LocSE which learns an augmented feature vector corresponding to each point and attentive pooling to perform feature aggregation using attention mechanism. It stacks these modules together in the form of dilated residual block using skip connections.

2.4 Methods based on attention mechanism

The prime focus of all of above mentioned point based methods is to learn local context and then acquire global context by successive aggregation of these locally learnt features in hierarchical fashion. On the other hand there are various methods which directly learn global context from local features using attention mechanism. A-SCN[8] combines the idea of shape context with global self-attention which embodies both the selection and aggregation operations into a single alignment process. But its results suffer due to the lack of support of local features. PointASNL[39] on the other hand achieves much superior performance by incorporating attention mechanism both on local as well as global level. It learns to weight the points lying in the neighborhood of initially sampled point obtained as a result of farthest point sampling using self-attention and then use these weights to adjust the coordinates of sampled points. It uses attention on global level in its PNL(point non local)cell for global context aggregation and fuses features learnt in both local and non local cells which greatly facilitates the recognition tasks.

Methodology

We propose an end to end framework for raw point cloud classification and part segmentation which exploits deep residual learning to integrate features at various levels. Our method belongs to the paradigm of hierarchical feature learning in which points are down sampled as we go deeper while features belong to the higher dimensional space.

3.1 Building Blocks

The proposed architecture is composed of multiple blocks to process the points in hierarchical manner and encode features. These blocks are stacked together following the framework of residual learning [33] which facilitates the propagation of information along the hierarchy. Fundamental building blocks comprise of three component layers. First layer is sampling layer to select subset of points as query points which serve as candidates around which local neighborhood is abstracted using constraint based KNN search. The second layer is a baseline layer of annular convolution proposed by Komarichev et al. [32] which is used as a mechanism of feature extraction. Although annular convolution tends to capture discriminative features by leveraging the use of multiscale ring shaped regions but the empty spaces between proposed dilated rings miss out the potential neighbors lying contextually closer to query points. Moreover, in our architecture as the deeper layer of annular convolution contains rings lying potentially far from the query point so empty spaces of dilated rings adversely affect the quality of learnt features. To overcome this issue, we use concentric rings instead of dilated rings. Third layer is pooling layer which serves as a mechanism of feature aggregation to summarize the relation

between candidate points and their local neighborhood. In the subsequent section we will illustrate the component layers in detail.

3.1.1 Farthest point sampling

FPS algorithm has been widely used to generate smaller sub set point cloud for various tasks related to surface processing from graph clustering [34], to progressive image sampling [35], to curved manifold and point cloud sampling [36]. Pointnet++ [2] and subsequent architectures [4,32,5,39,40] continued to use it to make the computation feasible for the architecture. It incurs great advantage as far as computational efficiency is concerned and tends to minimize the information loss by picking the candidate points from all over the input point cloud. It iteratively operates over entire point cloud and picks up the farthest point from the already selected points [36]. E.g. given $M \subset \mathbb{R}^n$ pointsets, it finds $\{m_1, \dots, m_k, \dots, m_K\}$, which represents a reordering of metric space such that the k th selected point i.e. m_k is lying farthest out of $\{m_1, \dots, m_k\}$ points [7]. The purpose to give preference to FPS over other contemporary techniques [39,41] is its ability to produce relatively uniform and original points i.e. the generated subset is always a part of original point cloud.

3.1.2 Annular convolution

The essence of CNNs is the convolution operator which captures correlational characteristics of points in their local neighborhood. Deep learning on 3d data can be either on structured representation such as volumetric grids and multi-view images or on unstructured representation such as point clouds. Convolution on former can be easy to implement but it comes with the expense of memory and computation. However, convolution on latter is a challenging task due to irregular and sparse nature of raw point cloud.

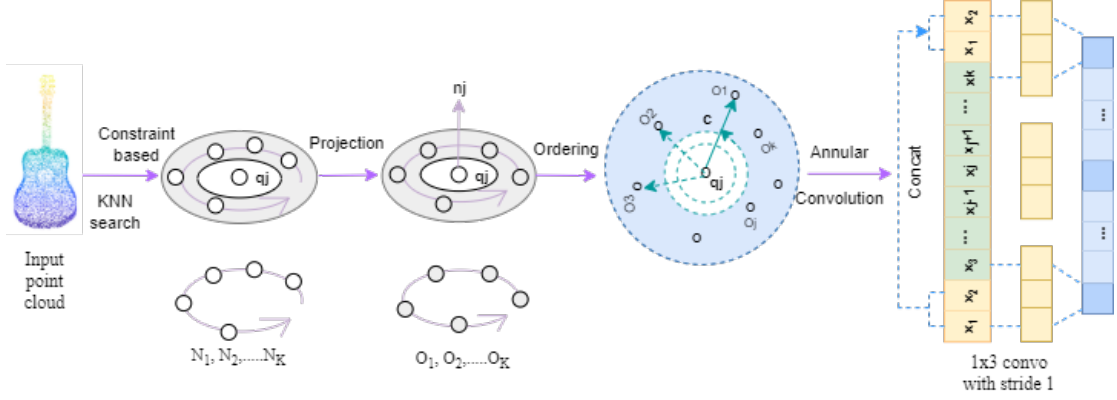


Figure 3.1: Graphical illustration of annular convolution. Given a query point q_j , constraint based KNN search fetches K nearest neighbors i.e. $\{N_1, N_2, \dots, N_k\}$ on the rings. Given normal n_j corresponding to query point q_j , it projects the neighboring points onto a tangent plane to calculate the projections of neighboring points denoted by $\{O_1, O_2, \dots, O_k\}$. In the next step, these projections are used to ordered neighboring points in counterclockwise direction as per the reference direction c . Lastly convolution is performed with the kernels of size 1×3 to abstract per point features.

We employ annular convolution operator (introduced in [32]) to capture the local geometric representation of points. Annular convolution is a four fold process whose graphical illustration is given in figure 3.1. Unlike images, the notion of neighborhood is not implicit for point clouds due to their unordered and irregular nature. Since the marvel of CNNs lies in the ability of convolution operator to learn the abstraction of points in their local neighborhood so it has to be captured explicitly using K-NN search or ball query algorithm [2]. Annular convolution uses multi ring strategy to restrict the infusion of neighbors of one scale in the other. The regular and dilated rings impose the constraint on search area of the points to materialize the constraint based K-NN search which actually guarantees to find the closest and unique neighbors of a point.

3.1.2.1 Concentric Rings

Let's say (M, d_m) is a point-set and metric on the set pair, where $M \subset \mathbb{R}^n$ represents the underlying 3 dimensional points and d_m defines a notion of distance between these points which is a non-negative and real number. Together they constitute the Y metric space which implies $Y = (M, d_m)$. This metric space induces very interesting properties

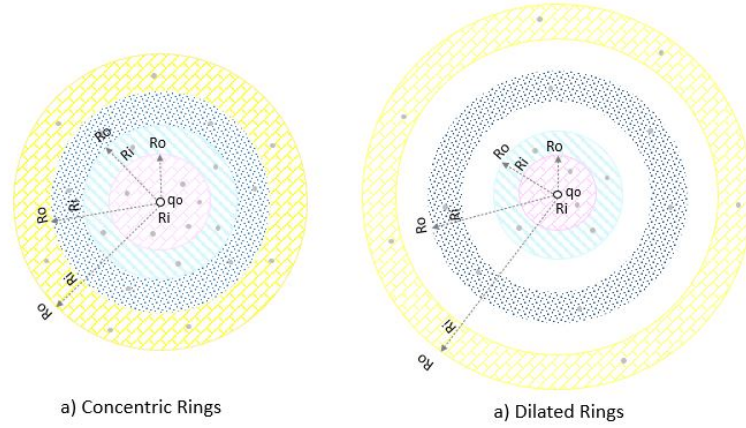


Figure 3.2: Dilated rings grow in size much rapidly as compare to concentric rings due to empty spaces in between. In the deeper layers, the neighbours extracted on a concentric rings are contextually more meaningful being closer to the query points. Each ring is characterized by inner radius i.e. R_i and outer radius i.e. R_o . The term concentric points towards the common centre of each ring i.e. the query point.

in point-sets. Implicit notion of open sphere and ball, neighborhood and nice geometric properties are few to mention. The point-sets lying in 3d metric space satisfy the equation $x^2 + y^2 + z^2 < r^2$ which means that these points are actually contained by an open sphere of radius r centered at origin. Likewise the neighborhood of a point in 3d space corresponds to actually the interior points of a sphere which is centered exactly at that point. Ball query algorithm used by pointnet++ considers a neighborhood within a ball of a particular radius centered at query points, while [32] proposed a ring shaped strategy which splits a ball into multiple dilated and regular rings by imposing a constraint of distance in the form of radius. Dilated rings used in annular convolution inspired by dilated convolution [37] span over larger area with same kernel size.

As our deeper architecture tends to stack more layers of annular convolution than [32] so the choice of dilated rings induces a huge amount of wastage of space lying contextually closer to the query point which actually contains the potentially meaningful neighbors. To make use of that empty space we propose to use concentric rings by fusing both the regular and dilated rings together. Each concentric ring is characterized by inner and outer radius as depicted in figure 3.2. Although Next ring starts where the previous ends so outer radius of inner ring becomes the inner radius of the next one but the term concentric points towards the common center of all rings which is the query point. This strategy not only manages to improve the count of unique number of neighbors by

exploiting the left over space around point sets but also restricts the inclusion of features learnt at one scale into the other. Both of these factors benefit the multiscale feature aggregation for our deeper architecture.

3.1.2.2 Projection and Ordering

Annular convolution [32] further refines the notion of neighborhood in a local region by ordering the neighbors w.r.t angle. Normal being an extremely important geometric property of point-clouds, is used as a helping hand to find projections which facilitates the ordering process. The Constraint based K-NN search gives $N_i, i \in \{1, 2, \dots, K\}$ neighbors corresponding to a query point q_j , where K represents the total number of neighbors. The points belonging to the neighborhood set are projected onto a tangent plane characterized by a unit normal n_j , to compute the orthogonal projections on rings and use them in the dot and cross product to compute the angle between query points and neighboring points w.r.t a reference direction \mathbf{c} as given below[32]:

$$O_i = N_i - ((N_i - q_j) \cdot n_j) \cdot n_j, i \in \{1, 2, \dots, K\} \quad (3.1.1)$$

$$\cos(\Theta_{oi}) = \frac{\mathbf{c} \cdot (O_i - q_j)}{\|\mathbf{c}\| \|O_i - q_j\|} \quad (3.1.2)$$

In third step it sorts the neighboring points based upon ascending or descending values of Θ_{oi} to obtain the clockwise or counterclockwise order. This technique elevates the quality of neighborhood by exploiting angle between the points besides distance of course. It yields the neighbors both distance wise and angle wise. Lastly convolution is performed on these ordered point sets with kernels of arbitrary choice to abstract the per point features.

3.1.3 Pooling

Annular convolution encodes features from query point based upon neighborhood point set. As we have used ring based convolution scheme as a mechanism of feature learning so feature aggregation has to be applied across all neighbors in each ring individually [32]. We tested various approaches of feature aggregation including average pooling, max pooling, exponential softmax aggregation [38] and attentive pooling [7]. Our idea behind using attentive pooling was to let the network learn how to assign weights to multi-scale

features from different rings based upon their proximity to the query point by incorporating attention mechanism. Max pooling stood tall out of all the tested approaches and manages to aggregate the distinctive features from concentric rings.

Attentive pooling underperformed max pooling in this scenario because network suffers from overfitting which affects its generalizability. Since in our deeper architecture the number of filters in deeper layers of annular convolution increases to upheave the dimensions of feature space and attentive pooling adds to the load of number of parameters with its MLP and convolution layer. The overall load incurred by both annular convolution and attentive pooling in deeper layers cause the network to over-fit which undermines its performance hence max pooling is the better choice.

3.2 Proposed Architecture

Our deep residual hierarchical architecture is composed of three blocks connected to each other using the residual framework[33] which facilitates the training of deeper networks.it takes raw point cloud as an input and assigns a category label to a complete object or a part category label (such as airplane wing, table leg) to each point in the input. Workflow begins by applying set of operations such as sampling, annular convolution and pooling in each block to compute point features which uniquely describe the local regions. To allow the stacking of these blocks , theory of residual networks has been employed which is explained at great length in the subsequent part of this section. Features extracted from three blocks are concatenated and processed again in the last convolutional layer with kernel size 1×1 followed by batch normalization and ReLU layer to compute high level features. These high level concatenated features are passed through fully connected layers followed by dropout and ReLU activation layers in the end. Although in case of segmentation, a mechanism for feature propagation from all three blocks and an interpolation technique has to be placed before fully connected layers to predict the per point label. We use the same strategy as used in [32] to compute the segmentation class distribution for each point.The architecture diagram can be visualized in figure 3.3.

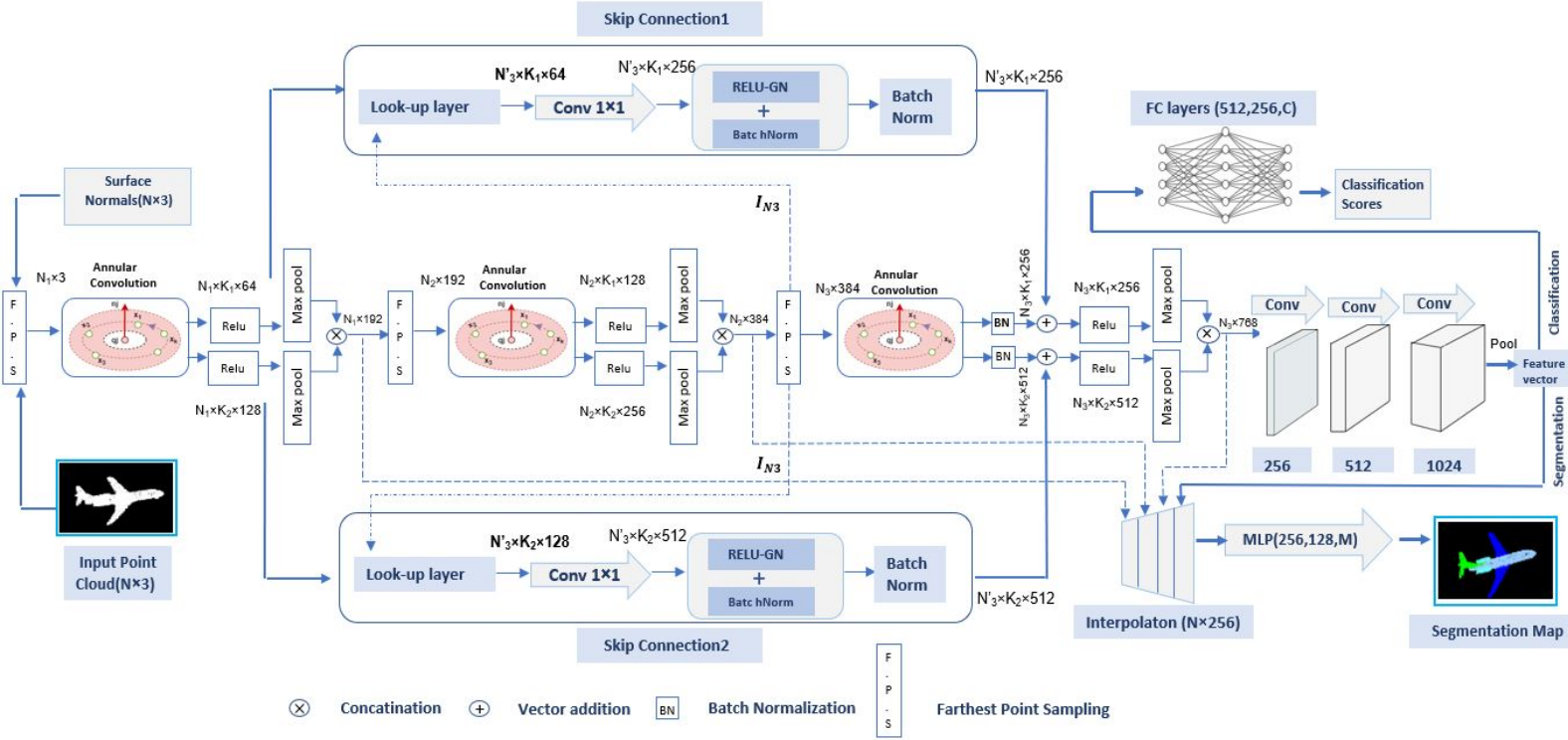


Figure 3.3: The proposed Architecture. N, N_1 and N_2 represent the input to the first, second and third block respectively (where $N > N_1 > N_2$). Each block of annular convolution comprises of two rings with K_1 and K_2 number of neighbors in ring1 and ring2 respectively. I_{N_3} represents the indices of N_3 points of block-three. N'_3 denotes the previously computed features of points with I_{N_3} indices. C and M are the number of classification and segmentation classes.

We took the inspiration to exploit the network depth from [33] which states that the depth of representations being a crucial aspect of networks causes them to achieve a gain in accuracy and ease of optimization if used in accordance with the residual learning principle. In hierarchical feature learning paradigm point cloud usually down samples, while the size and dimensions of feature space increase drastically as we go deeper. The down sampling strategy helps the network to manage computational load which is incurred by the processing of features belonging to much higher dimensional space. The size of neighborhood employed in annular convolution [32] is fixed for both rings so if the total number of neighbors fall short of a fixed count, it appends the nearest neighbor to the query point in the list of neighbors to keep the count consistent. Following the down sampling approach, the size of input point cloud is decreased considerably in deeper layers due to which the neighborhood captured by constraint based KNN search

contains less unique points and more redundantly replicated nearest neighbor points which adversely affects the quality of features learnt in deeper layers. Both the difficulty in optimization faced by deeper networks in general [33] and non-discriminative features learnt due to redundant neighborhood cause the network’s performance to suffer. To address this issue, we bring skip connections into the picture which not only helps alleviate the optimization difficulty of deeper network but also improve the quality of features by propagating the features of initial layers to deeper ones. The Superior results shown by deeper network with skip connection than its plain deeper counterpart consolidates our claim (as explained in next section).

3.2.1 Residual Block

The general form of residual block is given in figure 3.4 which illustrates the propagation of feature maps from first annular layer to third. Second layer is skipped to keep the residual learning principle intact. Its worth noting that we use two skip connections because of two rings in annular convolutional layer to add the feature maps from both rings in block-one to that of block-three individually. To facilitate the argument, we present the general mathematical form of residual block given in [33]:

$$y_l = F(x_l, W_l) + f(x_l) \quad (3.2.1)$$

$$f(x_l) = \sigma_1(\sigma_2(W_s x_l) + \sigma_3(W_s x_l)) \quad (3.2.2)$$

$$x_{l+1} = \sigma_4(y_l) \quad (3.2.3)$$

Where F denotes the residual function which is the nonlinear, W_l depicts the learnable weights. x_l and x_{l+1} represent the input of the l_{th} and $(l+1)_{th}$ unit respectively i.e. the skip connection or the shortcut. $f(x_l)$ depicts the mapping function which facilitates the addition of both the residual and skip parts. The residual part F is learnt by the stack of annular convolution layers. We use two rings in each layer. Annular convolution can support kernels of arbitrary size but we use kernels of size 1×3 for each ring. To compensate for the reduction in size due to valid convolution we pad two neighbors from original list to the neighbor’s list to restore the actual size.

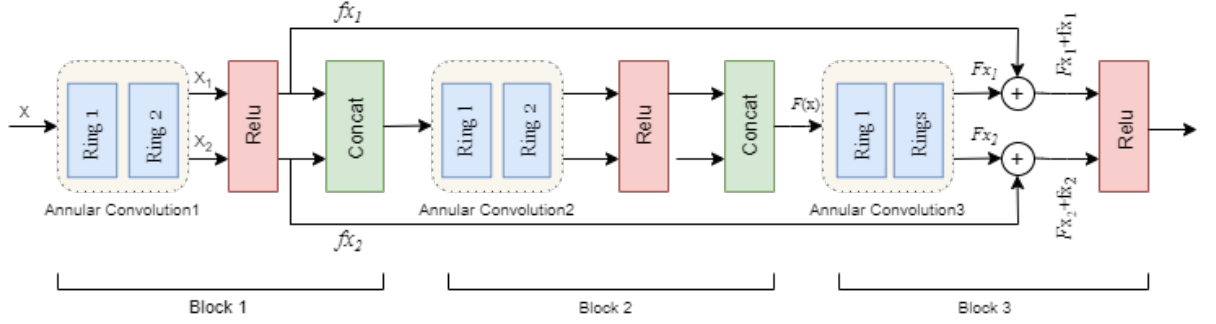


Figure 3.4: Simplified form of the building blocks with residual learning. Two skip connections indicate the propagation of features from Ring1 and Ring2 of the annular layer in block-one to the Ring1 and Ring2 of the annular layer in block-three.

3.2.2 Skip Connection

$f(x_i)$ represents the mapping function over here which has been exploited in different ways in the literature. E.g. it was identity mapping in case of Resnet [33] and pre-activation Resnet [42].

The whole idea of pre activation Resnet was to ensure the unimpeded flow of information from the beginning till end and it is particularly prevalent in rather deeper networks. In Resnet[33] although the Relu activation applied after addition operation becomes dormant after a while but still it modifies the info being propagated which hinders the direct path. Pre activation Resnet proposes to apply Relu activation before, instead of after summation which facilitates the info to have a direct path. Motivated by the transformer [44] Fenglin et al. [43] proposed to use skip connection and layer normalization in combination to build a skip connection based architecture which uses a modulating scalar to adjust the weighting between the residual part and the skip part. Although layer normalization somewhat eases out the hindrance in optimization due to gradient distortion caused by modulating scalar, but to stabilize the gradient further, they assign equal weightage to the skip and residual part and add them recursively with layer normalization which improves the expressive power of the model.

We propose a mapping function which can be visualized in figure 3.5.c. As we are applying the residual principle over point based hierarchical network in which the input i.e. point cloud is successively down sampled as it passes through different blocks unlike images in Resnet[33]. Due to this operation of down sampling, the feature maps of block-one cannot be directly added into the feature maps of block-three. Block-one

down samples the received input containing 1024 points to 512 points and then generates the feature maps for down sampled point cloud comprised of 512 points. These feature maps are propagated as it is to the block-three as skip connection. Likewise block-three down samples the input comprised of 128 points to 64 points and generates the feature maps accordingly. Feature maps computed in both the blocks cannot be added together unless they are made dimensionally compatible. To cater for this issue, we pass the volume generated by block-one through a look-up layer which searches for the previous feature maps of 64 points of block-three in it. This nonlinear look-up operation as shown in figure 3.3 fetches the tensor containing previous feature maps w.r.t the indices of 64 points of block-three from the volume propagated by block-one to facilitate the addition operation latter.

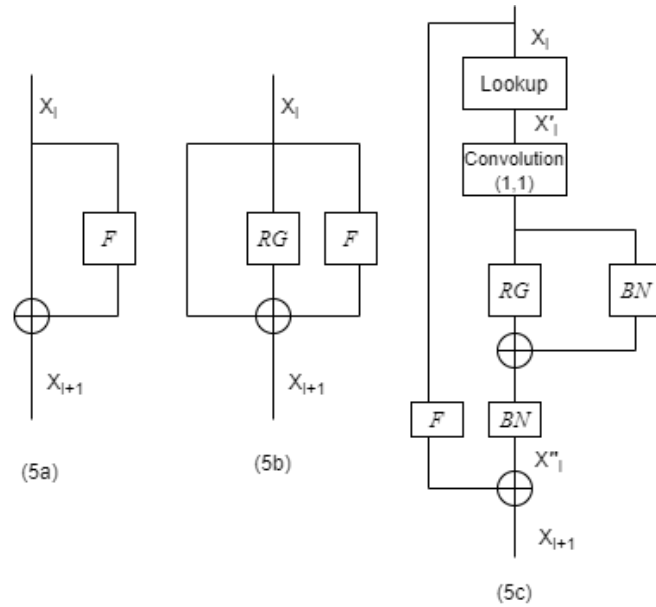


Figure 3.5: The Shortcut schematics: (3.5a) Original Resnet,(3.5b) Res-RGSNet, (3.5c) Ours

3.2.2.1 Projection shortcut

Equation 4 gives the mathematical illustration of the proposed mapping function in which σ_1 and σ_2 depict batch normalization. In spite of the look-up operation both the volumes i.e. the one propagated from block-one and the one computed in block-three are not compatible to be added together yet, due to the mismatch of last channel dimension. Number of input and output channels vary due to increase in the number

filters as it gets deeper. The channel dimension is increased significantly in latter layers due to increase in the number of filters applied by annular convolution. This issue can be resolved by using techniques such as zero padding or interpolation, but we employed a more plausible solution suggested by [33] which multiplies the identity mapping by a linear projection W_s . The linear projection is implemented using a convolutional layer with filters of size 1×1 which expands the channel dimension of the shortcut to match it to the residual part learnt in block-three. The convolutional layer is followed by a batch normalization layer which adds the extra stability to the gradients by smoothing out the optimization landscape[47].

3.2.2.2 Relu group normalization shortcut

σ_3 in in equation 5 depicts the non linear RG operation[46] which is a combination of Relu and group normalization. There has to exist a trade-off between the gradient stability and the representational power in deep networks [45]. Though identity shortcuts are a strength of deep residual networks being helpful in their training by fixing the infamous vanishing/exploding gradient problems but is also a weakness at the same time in terms of representational power. Gradient is not forced to go through the weights of residual blocks during its flow through the network, so it can possibly learn very little meaningful representation through some residual blocks which affects the overall representational power [50]. To circumvent this problem Zhang et al.[46] propose a nonlinear RG shortcut in which Relu induces non linear characteristics and the group normalization applied along the channel direction adds the stability.

3.2.2.3 Mapping function

Our mapping function (represented in equation 3.2.2) is inspired by Res-RGSNet proposed in [46] which is intuitively based upon the claim made in [45] that the gradient stability and representational power both being extremely important aspects can contribute significantly towards the superior performance if somehow a trade-off can be reached between the both ,while degrading either one of them can abysmally impair the training process which can lead to decline in performance. Res-RGSNet combines both the identity shortcut and non linear RG shortcut as shown in figure 3.5.b and

claims that it manages to achieve a boost in performance as compare to the individual shortcuts at different depths. We got the motivation from this design to add our projection shortcut and RG shortcut together to exploit the advantages of inherent linear characteristics of projection shortcut and better representational power induced by RG shortcut. Batch normalization depicted by σ_1 entails effective and faster optimization by stabilizing the behavior of gradients [47]. It prevents the activation magnitude from exploding and acts as an important regularizer by maintaining non vanishing and non-exploding model parameters[51].

We also tested our network with both the projection shortcut and RG shortcut individually, but it encounters the decline in accuracy as depicted in table 4. The superior results brought by this design choice than both of the independently used shortcuts validates the claim made in [46] and reinforces the argument presented in [45] .

Experiments, Results And Analysis

We evaluate the proposed model on variety of tasks such as classification on both the synthetic as well as real world dataset and part segmentation. We will share the precise details of the experiments and comparisons of our model with the state of the art in the next subsection.

4.1 Point cloud classification

4.1.1 Synthetic Data

In point cloud classification, the goal is to assign a correct label to the point cloud of the 3d shape. We evaluate our architecture on MODELNET40[48] dataset which is comprised of 12,311 CAD models out of which 9843 models correspond to the train split and 2468 correspond to the test split. Being synthetic dataset, MODELNET40 contains well segmented, free from noise and complete objects which are grouped into 40 categories.

We sample 1024 points with normals as input from each mesh surface and normalize them into unit sphere as given in [2] where normals are only used for the ordering of neighboring points in local regions. Similarly, 512 and 128 points are sampled as input in subsequent blocks using farthest point sampling as explained earlier. Data augmentation being a prevalent strategy these days helps improve the generalizability of network and

induce diversification in the training data without getting to increase the actual quantity of data. As a measure of augmentation, we apply techniques such as random scaling to diversify the size of objects, shift the object’s locations, apply jittering to point positions with Gaussian noise and shuffle the order of points to help FPS produce different query points.

Table 4.1 summarizes the quantitative comparison of our method with state-of-the-art point-based methods in terms of classification. Our method evidently outperforms existing state of the art point-based methods in the category of 1K input points while it gives slightly worse performance than PAN and SO-Net which use rather denser point clouds with 5k points and normals as input. RS-CNN manages to improve from 92.9% to 93.6% by incorporating a voting mechanism with various transformations which is different from one time vote setting therefore we exclude its results from comparison.

Point based methods with 1K points				
Method name	Input type	# points	AAC	OA
Pointnet	xyz	1K	–	89.2
Pointnet++	xyz	1K	–	90.7
SO-Net	xyz	2K	87.3	90.9
PAT	xyz+norms	1K	–	91.7
3DGCN	xyz	1K	–	92.1
PointWeb	xyz	-	89.4	92.3
WCP-Net	xyz	1K	90.53	92.41
PointConv	xyz+norms	1K	–	92.5
FPCConv	xyz+norms	–	–	92.5
ACNN	xyz	1K	90.3	92.6
Point2Sequence	xyz	1K	90.4	92.6
DensePoint	xyz	1K	–	92.8
DensePoint(vote)	xyz	1K	–	93.2
RS-CNN	xyz	1K	–	92.9
RS-CNN(Vote)*	xyz	1K	–	93.6
InterpCNN	xyz	1K	–	93.0
PointGLR	xyz	1K	–	93.0
PAN	xyz	1K	–	93.1
ShellNet	xyz	1K	–	93.1
DRNet	xyz	1K	–	93.1
Grid-GCN	xyz	1K	91.3	93.1
PointASNL	xyz+norms	1K	–	93.2
ours	xyz	1K	90.9	93.27
Point based methods with more points				
Pointnet++	xyz+norms	5K	–	91.9
ψ -CNN	xyz	10K	88.7	92.0
KPConv rigid	xyz	6.8K	–	92.9
PAN	xyz+norms	5K	–	93.4
SO-Net	xyz+norms	5K	90.8	93.4

Table 4.1: Classification results on ModelNet40 dataset. AAC is accuracy average class, OA is overall accuracy. xyz means 3d coordinates and norms means surface normal vector.

4.1.2 Real world Data

To inspect the performance of our architecture on real world point cloud based data we test it on ScanObjectNN[52] which is a benchmark dataset of real world classification. Uy et.al [52] formulated this dataset by initially segmenting objects from mesh based scene datasets SceneNN[49] and ScanNet[53] then pre-processing and grouping them into 15 categories of household objects. Dataset has several variants to offer the difficulty levels of various kinds, but we use **OBJ-ONLY** variant which is by far the best real world counter part of MODELNET dataset.

4.1.2.1 Pre processing

We pre process ScanObjectNN[52] dataset to make it suitable for training and testing on our architecture. Several variants of this dataset are available which offer variety of challenges to the neural network. The first step of pre processing pipeline is to separate the pointclouds without background from those with background. The dataset contains a flag to indicate the occurrence of background instance in the pointcloud we used that flag to group both the categories separately with the help of a script in python. Calculation of surface normals is explained at length in the next paragraph. There are few more conversions in the pipeline such as conversion from H5 to xyz or txt to H5 which are actualized with the help of python scripts.

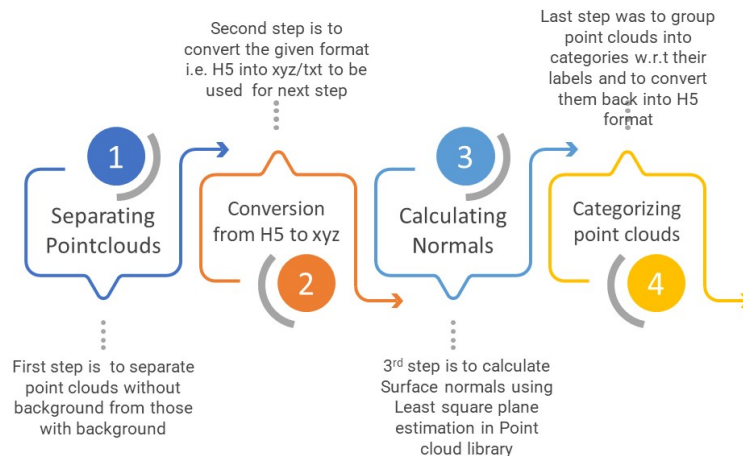


Figure 4.1: Pre processing pipeline of ScanObjectNN dataset.

Estimation of Surface Normals

We compute surface normals of ScanObjectNN dataset for annular convolution by using first order 3D plane fitting method [54]. To compute the normal of a point lying on 3D surface, a plane has to be fitted at that point which is tangent to the surface. Then the normal to that Tangent plane corresponds to the surface normal of the point [55]. So instead of finding normal directly, it becomes a least square plane fitting estimation problem [56]. Plane is characterized by a point q_j and normal \vec{n}_j . Neighborhood $X = \{x_i | i = 1, 2, \dots, K\}$ of the point q_j has to be considered to compute any geometric feature such as normal. The solution for normal \vec{n}_j lies in the eigenvalues and eigenvectors of covariance matrix C of the neighborhood X which is given as [56]:

$$C = 1/K \sum_{i=1}^K \zeta_j (x_i - q_j) \cdot (x_i - q_j)^T, C \in \mathbb{R}^n \quad (4.1.1)$$

$$C \cdot \vec{v}_\alpha = \lambda_\alpha \cdot \vec{v}_\alpha, \alpha \in \{0, 1, 2\} \quad (4.1.2)$$

where λ_α and \vec{v}_α represent the α_{th} eigenvalue and eigen vector respectively out of the total three eigenvalues and eigenvectors produced by solving Covariance matrix. Eigenvalues produced by C are always real numbers because of its symmetric, positive and semi definite nature. As the correct eigenvector for least square solution is the one which corresponds to the smallest eigenvalue [56] so if $\lambda_2 \geq \lambda_1 \geq \lambda_0 \geq 0$ then eigenvector corresponding to smallest eigenvalue i.e. \vec{v}_0 gives the approximation of normal vector \vec{n}_j .

Out of total 2902 objects we use the default train and test split percentage (training 80%, test 20%) given by [52]. we sample 1024 points with normals as input and normalize them into unit sphere. Network uses (x,y,z) coordinates as input for training and normals for the ordering of neighboring points. Table 4.2 gives the comparison of performance between our architecture and existing state of the art architectures on ScanObjectNN dataset. Our technique outperforms the existing state of the art except PointGLR[57] which belongs to the un-supervised learning paradigm.

It can be clearly inferred from Table 4.1 and Table 4.2 that the results of existing architectures on real world dataset lag behind their synthetic counter part by a significant margin. The reason behind this lag is the fierce challenges offered by real world data to the networks such as incomplete and partial objects due to scanning or reconstruction

Method	Accuracy
3DmFV [58]	73.8
PointNet[1]	79.2
SpiderCNN[60]	79.5
PointNet++[2]	84.3
PointCNN[4]	85.5
DGCNN[15]	86.2
Ours	86.4
PointGLR[57]	87.2

Table 4.2: Classification accuracy on ScanObjectNN dataset.

errors or occlusion, lack of definite and accurate boundaries around the objects and presence of low frequency noise etc. as shown in figure 4.2.

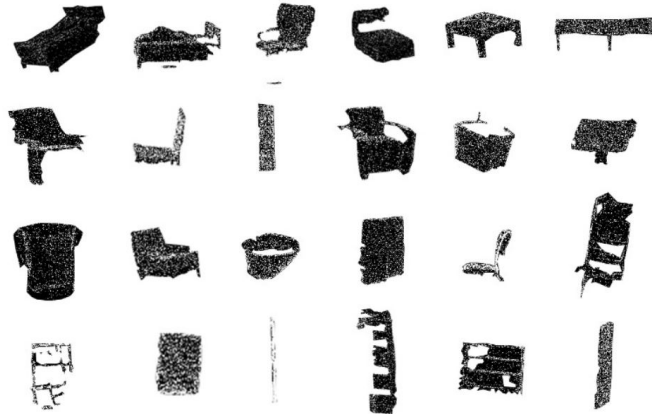


Figure 4.2: Visualization of objects belonging to OBJ-ONLY variant of ScanObjectNN dataset.

4.2 Point cloud Segmentation

We test the proposed architecture on ShapeNet-part [61] dataset which is a renowned dataset for part segmentation. In part segmentation the goal is to assign a correct part category label such as airplane wing, table leg etc. to each point of the 3d shape. ShapeNet-part [61] dataset is comprised of 16,881 richly annotated 3d shapes belonging to 16 categories labelled with 50 parts in total. It contains 14,007 and 2874 shapes in

Method	mIOU	aero	bag	cap	car	chair	ear	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	Skate	table
PointNet[1]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
SO-Net[62]	84.6	81.9	83.5	84.8	78.1	90.8	72.2	90.1	83.6	82.3	95.2	69.3	94.2	80.0	51.6	72.1	82.6
PointNet++ [2]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
3D-GCN[64]	85.1	83.1	84.0	86.6	77.5	90.3	74.1	90.9	86.4	83.8	95.6	66.8	94.8	81.3	59.6	75.7	82.8
P2Sequence[67]	85.2	82.6	81.8	87.5	77.3	90.8	77.1	91.1	86.9	83.9	95.7	70.8	94.6	79.3	58.1	75.2	82.8
DGCNN[15]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
SRN[77]	85.3	82.4	79.8	88.1	77.9	90.7	69.6	90.9	86.3	84.0	95.4	72.2	94.9	81.3	62.1	75.9	83.2
SFCNN[78]	85.4	83.0	83.4	87.0	80.2	90.1	75.9	91.1	86.2	84.2	96.7	69.5	94.8	82.5	59.9	75.1	82.9
PAN[71]	85.7	82.9	81.3	86.1	78.6	91.0	77.9	90.9	87.3	84.7	95.8	72.9	95.0	80.8	59.6	74.1	83.5
PointConv[5]	85.7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ACNN [32]	85.9	83.9	86.7	83.5	79.5	91.3	77.0	91.5	86.0	85.0	95.5	72.6	94.9	83.8	57.8	76.6	83.0
PointASNL[39]	86.1	84.1	84.7	87.9	79.7	92.2	73.7	91.0	87.2	84.2	95.8	74.4	95.2	81.0	63.0	76.3	83.2
RS-CNN[69]	86.2	83.5	84.8	88.8	79.6	91.2	81.1	91.6	88.4	86.0	96.0	73.7	94.1	83.4	60.5	77.7	83.6
InterpCNN[70]	86.3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PointADRNet (Ours)	86.3	84.0	84.8	85.2	80.1	91.3	78.8	91.6	86.9	84.6	95.3	73.6	95.5	84.6	60.5	76.7	84.1
KPConv deform[74]	86.4	84.6	86.3	87.2	81.1	91.1	77.8	92.6	88.4	82.7	96.2	78.1	95.8	85.4	69.0	82.0	83.6
DensePoint[68]	86.4	84.0	85.4	90.0	79.2	91.1	81.6	91.5	87.5	84.7	95.9	74.3	94.6	82.9	64.6	76.8	83.7
DRNet[73]	86.4	84.3	85.0	88.3	79.5	91.2	79.3	91.8	89.0	85.2	95.7	72.2	94.2	82.0	60.6	76.8	84.2
ψ -CNN [75]	86.8	84.2	82.1	83.8	80.5	91.0	78.3	91.6	86.7	84.7	95.6	74.8	94.5	83.4	61.3	75.9	85.9

Table 4.3: Part segmentation results (instance mIOU %) on shapenet part dataset.

train and test split, respectively. We sample 2048 points with normals for each shape as input same as given in [32], where normal are not as additional features but just as helping hand for the ordering of points in annular convolution. We compute mIOU (mean intersection over union) of the shapes as well as overall categories and use it as evaluation metric.

In table 4.3 we compare our PointADRNet with other state of the art methods which consume raw point cloud as input. Our method achieves 86.30 mIOU which is evidently on par with state of the art. For fair comparison we exclude the specialized networks of segmentation [3,91,93] and only include joint networks of classification and segmentation.

4.3 Qualitative Results

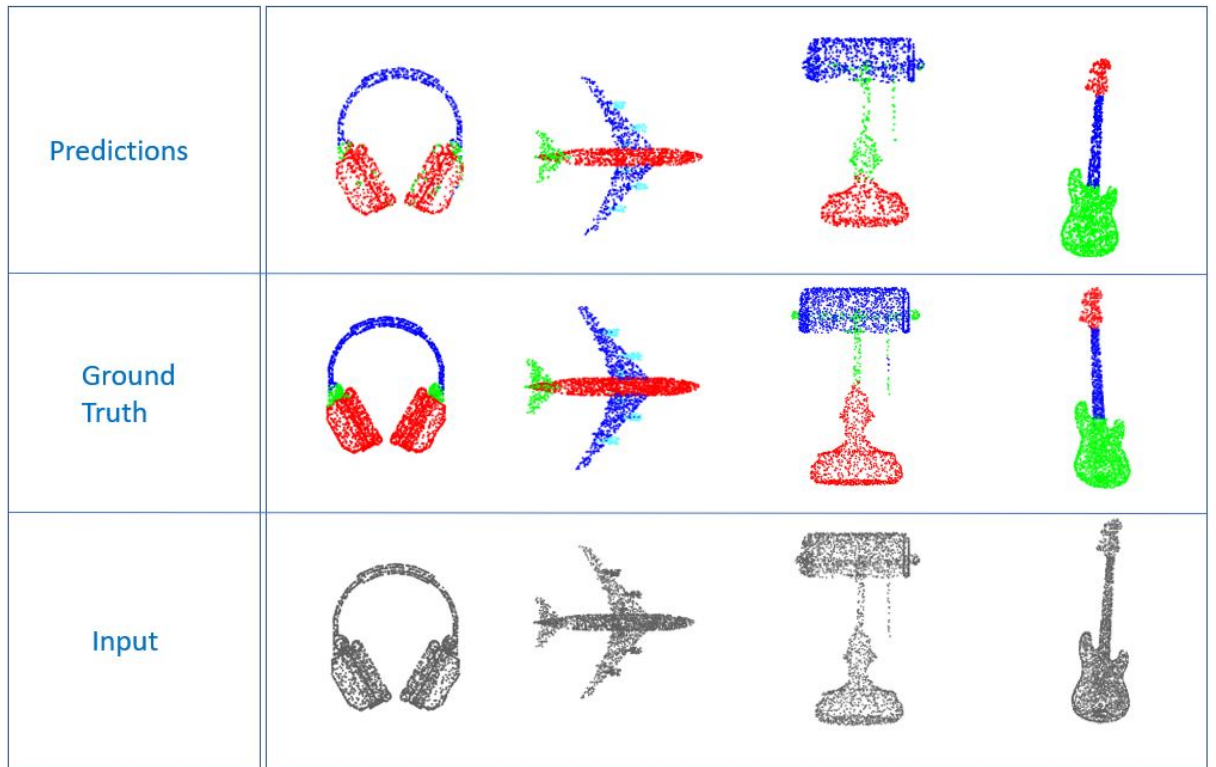


Figure 4.3: Qualitative results of part segmentation w.r.t ShapeNet-part dataset. Input and Ground Truth are plotted with around 2500 points per point cloud and predictions with 2048 points per point cloud.

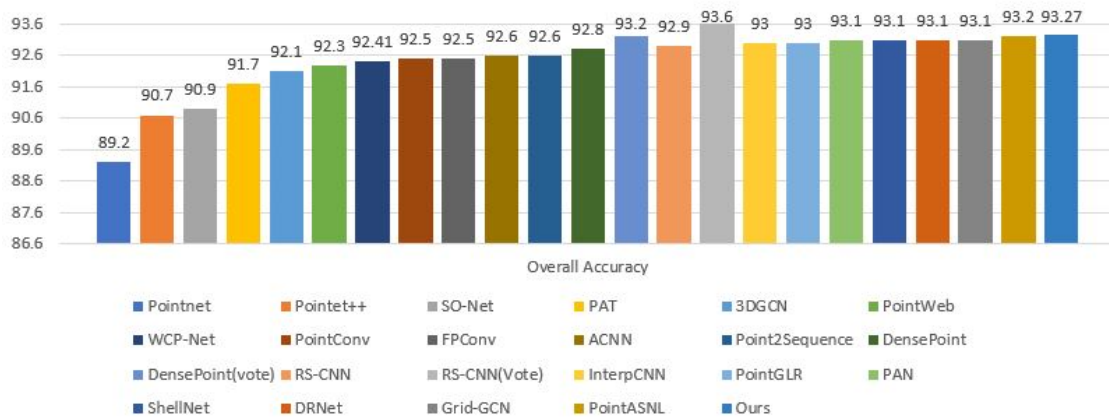


Figure 4.4: Overall Accuracy of existing state of the art networks of point cloud classification on Modelnet40 dataset. The input to all these networks is raw pointclouds with 1024 points representing xyz coordinates.

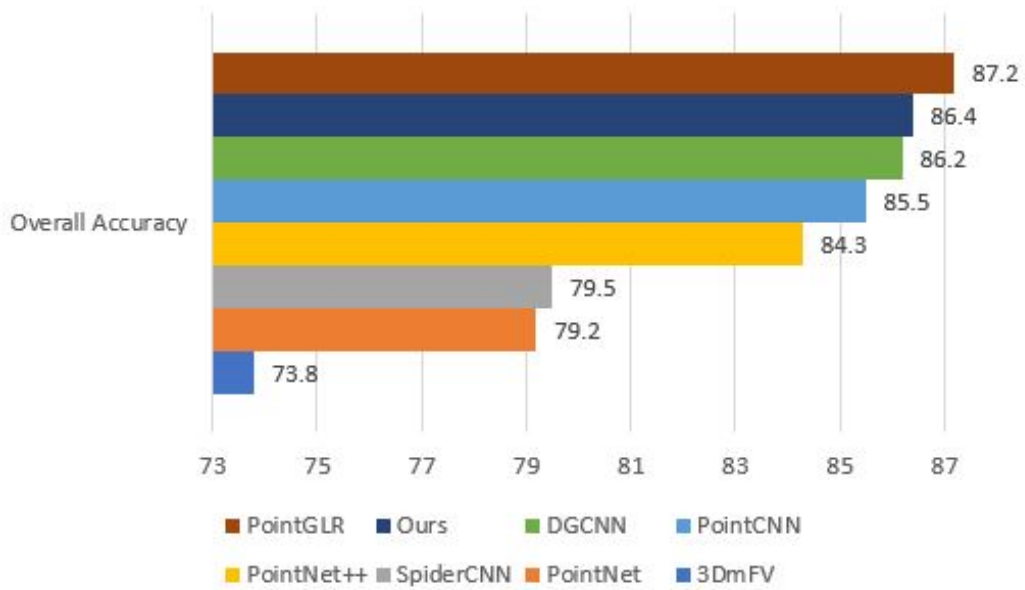


Figure 4.5: Overall Accuracy of existing state of the art networks of point cloud classification on ScanobjectNN dataset. The results on real world dataset clearly lags behind their synthetic counterpart.

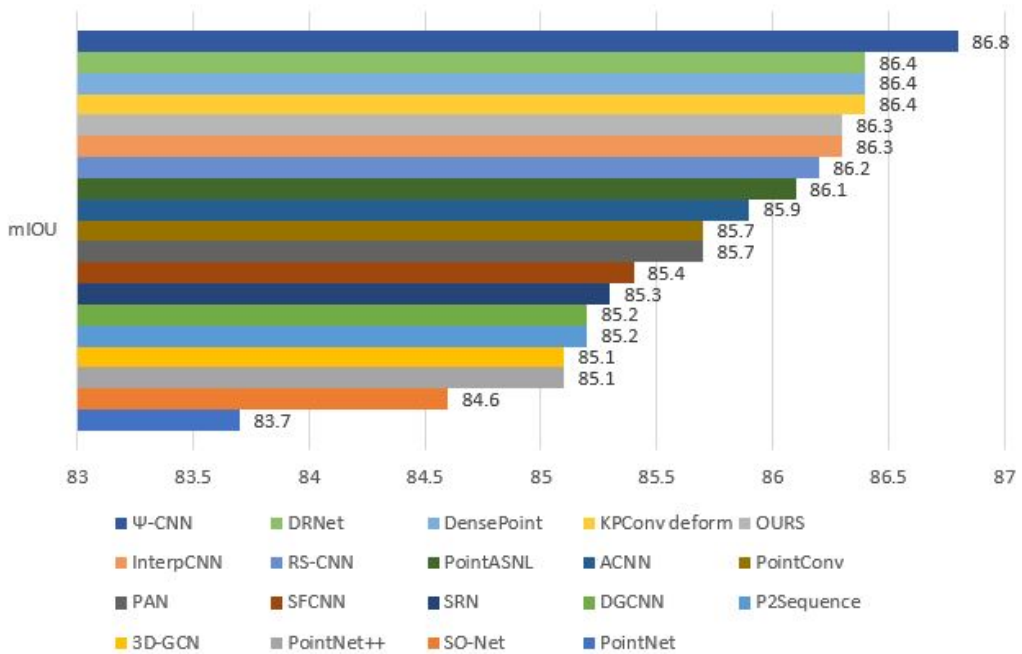


Figure 4.6: Comparison of state of the art methods on Shapenetpart dataset for part segmentation with mIOU as evaluation metric.

Ablation Study and Discussion

5.1 Ablation Experiments on Modelnet40 dataset

We conduct the ablation study of our network w.r.t modelnet40 dataset to study the effectiveness of architectural components. We conduct detailed experimentation to evaluate the proposed skip connection (Sec 3.2.2) and its components such as projection shortcut (Sec 3.2.2.1), Relu group normalization shortcut (Sec 3.2.2.2) and our proposed mapping function which exploits the sum of both these shortcuts (Section 3.2.2.3).

In first experiment we remove the skip connections i.e., we train and test a three blocks deeper plane network of classification on Modelnet40 dataset and report the results in table 5.1. A clear decline in accuracy can be witnessed in plan deeper network i.e., without skip connections. Due to hierarchical nature of our network, the quality of features learnt in last block suffer because of redundantly replicated neighboring points from the successively down sampled point cloud. We intuitively claim that the skip connections not only help alleviate the optimization difficulties faced by the gradient like it does in any deep network, but they also boost up the overall quality of features by adding the features of block-one to otherwise poorly learnt features in block-three. The comparison between the results of three blocks deeper plane network and three blocks deeper network with skip connections reinforces our claim.

In the second and third experiments, we incorporate the skip connections with simple projection shortcuts and Relu group normalization shortcuts respectively but both of them clearly under perform our final choice of shortcut which is formulated by the addition of both the projection and ReLU group normalization shortcuts.

Method	AAC	OA
1): -Remove skip connections	90.29	92.56
2):-Skip connections with Projection shortcuts	90.55	93.11
3):-Skip Connections with RG shortcuts	90.24	92.97
4):- Skip connections with RG+ projection shortcuts followed by batch norm	90.9	93.27

Table 5.1: Ablation experiments of our proposed architecture on ModelNet40 dataset to reinforce the importance of proposed architectural components. Where ,AAC denotes the accuracy average class and OA denotes overall accuracy.

5.2 Testing with various configurations

We also test the proposed architecture by replacing the ReLU activation with CReLU as well as with various configurations given in [42] such as ReLU only pre-activation and full pre-activation and report results in table 5. Although pre-activation configuration [42] ensures the direct flow of information from beginning till the very end but it mostly benefits the networks which are extremely deep. Moreover, in our case information cannot flow unimpeded anyway due to the convolution followed by batch normalization operation in the projection shortcuts to overcome the dimensional mismatch. Hence it gives underwhelming performance in the current setting as evident from table 5.2.

Network	AAC	OA
1): - Replace ReLU with CReLU	89.89	93.07
2):- ReLU only pre-activation[42]	90.05	92.99
3):- Full pre-activation[42]	89.81	92.91

Table 5.2: Results of experiments with CRELU,Relu only pre-activation and full pre-activation on Modelnet40 dataset.

Depth	No of Skip connections	AAC	OA
3 blocks deep network	2	90.9	93.27
5 blocks deep network	4	89.99	92.48

Table 5.3: The comparison of accuracy w.r.t the depth and number of skip connections. Each block processes 512,128,64,32 and 16 points respectively. Number of points can not be decreased beyond it due to hierarchical down sampling nature of the network. These experiments are done on MODELNET40 dataset where AAC denotes the accuracy average class and OA denotes the overall accuracy.

Network	Training Accuracy	Test Accuracy
1):- Attentive pooling [7]	92.99	92.03
2):- Exponential Softmax aggregation [38]	92.7	92.4
3):- Average pooling	92.55	93.07
4):- Max pooling	92.54	93.27

Table 5.4: In this experiment, we replace the pooling layer of each block with different pooling strategies to see its impact. Max pooling outperforms all the tested techniques and hence it is kept as a final design choice in the architecture. Training and testing is conducted using MODELNET40 dataset.

Network	AAC (Accuracy Average Class)	OA (Overall Accuracy)
1):- Merged Rings plus attention Mechanism	90.23	92.34
2):- Annular Rings plus attention Mechanism	90.1	92.38

Table 5.5: In the first experiment, we merge the annular rings and then apply attention mechanism to assign the attention weights to neighbors in unified ring. Let's say $N(x_i)$ denotes the K nearest neighbors of a query point x_i and together they form a group. We apply attention mechanism same as [39] to let the network assign scores to the group members as per their significance which is captured in the form of context vector. Likewise, in the second experiment attention mechanism is applied to annular rings but both the experiments evidently give underwhelming performance.

No of MLPs	Accuracy Average Class (AAC)	Overall Accuracy (OA)
[256,512,1024]	90.9	93.27
[256,512,1024,2048]	90.88	93.25
[64,128,256,512,1024,2048]	90.9	93.25
[16,32,64,128,256,512,1024,2048]	90.62	92.887

Table 5.6: Comparison of accuracy in terms of different number of MLPs used post annular convolution.

No of MLPs	Accuracy Average Class (AAC)	Overall Accuracy (OA)
[512,256,40]	90.9	93.27
[1024,512,256,64,40]	90.56	93.0325
[2048,1024,512,256,64,40]	90.53	92.92

Table 5.7: Comparison of accuracy in terms of different number of Neurons in fully connected layers. The accuracy tends to drop due to over fitting as number of neurons increase. To manage the computational load of increasing number of neurons batch size has to be decreased accordingly.

Conclusion And Future Work

6.1 Conclusion

In this work, we have proposed PointADRNet, a novel 3d point based deep, hierarchical architecture for object classification and part segmentation of raw point clouds. Despite of its simpler and modular design it achieves on par or even better performance in comparison to existing state of the art architectures. This simple methodology and its superior results are expected to offer a new perspective to potentially explore the network depth and its relevant aspects such as skip connections and shortcut designs to improve the performance of convolutional architectures for raw point cloud.

6.2 Future Work

In this section, we propose some future recommendations for the design of proposed network and few design parameters.

6.2.1 Adaptive Adjustment of Neighborhood

Currently, the size of neighborhood employed in annular convolution is fixed so if the total number of neighbors fall short of a fixed count, it appends the nearest neighbor to the query point in the list of neighbors to keep the count consistent. However, we have empirically observed that the performance of architecture suffers due to these redundantly replicated nearest neighbor points. Therefore, in future, we intend to exploit Recurrent neural networks to adaptively adjust the neighborhood size individually for

each query point which can possibly give a boost in accuracy.

6.2.2 3d Convolution

Currently, the proposed architecture employs annular convolution as a mechanism of feature extraction. As a future recommendation, we propose to incorporate other state of the art point based convolution methods such as FPConv [94] PAConv [95] in the existing backbone architecture. PAConv dynamically builds up the kernels by using point positions as weights of coefficients. FPConv flattens the surface encompassed by pointsets and softly projects the points onto a 2d grid by learning a weight map and then perform 2d convolution on them. It would be an interesting insight to see how network performs with these methods of convolution.

6.2.3 Weighted Aggregation

In annular convolution, the ring based strategy ensures that features are learnt from all the rings. We apply max pooling to aggregate the features from those rings. An interesting future direction would be to apply weighted aggregation instead of max aggregation. A Gaussian function can be used to assign decaying weights to the rings based upon their proximity to the query point and then aggregation can be applied on those Gaussian weighted features.

6.2.4 Use of Additional Features

So far, we have only used features computed as a result of annular convolution for the underlying recognition task but an other credible future direction can be to concatenate conventional 3d point cloud features such as surface normals, eigen based features or plane residuals etc with the features computed as a result of annular convolution. Such a combination of features can possibly elevate the accuracy.

6.2.5 To further refine the neighborhood

Currently, the annular convolution captures the neighbors of query points by using euclidean distance as a metric, then it further sorts those neighbors w.r.t angle by calculating projections of points. These points can be categorized as both distance wise and

angle wise neighbors. An other future direction can be to further refine the notion of neighborhood w.r.t angle. The idea is to compute the dot product of surface normals of all the neighboring points with query point and then sort the neighbors in decreasing order of the dot product. It will possibly refine the notion of neighborhood and will contribute towards the improvement in accuracy.

References

- [1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In CVPR, 2017.
- [2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413, 2017.
- [3] H.-Y. Meng, L. Gao, Y.-K. Lai, and D. Manocha, “VV-Net: Voxel vae net with group convolutions for point cloud segmentation,” in ICCV, 2019
- [4] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution On XTransformed Points. Proceedings of Advances in Neural Information Processing Systems (NeuralIPS), 2018.
- [5] W. Wu, Z. Qi and L. Fuxin, "PointConv: Deep Convolutional Networks on 3D Point Clouds," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9613-9622, doi: 10.1109/CVPR.2019.00985.
- [6] 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
- [7] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, “RandLA-Net: Efficient semantic segmentation of large-scale point clouds,” CVPR, 2020
- [8] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4606–4615, 2018

- [9] 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
- [10] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In Robotics: Science and Systems, volume 1, pages 10–15607, 2015
- [11] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236, 2016
- [12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5939–5948, 2019.
- [13] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. VV-net: Voxel vae net with group convolutions for point cloud segmentation. In ICCV, 2019
- [14] M. Simonovsky and N. Komodakis, “Dynamic edge-conditioned filters in convolutional neural networks on graphs,” in CVPR, 2017
- [15] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. arXiv preprint arXiv:1801.07829, 2018.
- [16] K. Zhang, M. Hao, J. Wang, C. W. de Silva, and C. Fu, “Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features,” arXiv preprint arXiv:1904.10014, 2019
- [17] Y. Yang, C. Feng, Y. Shen, and D. Tian, “FoldingNet: Point cloud auto-encoder via deep grid deformation,” in CVPR, 2018.
- [18] K. Hassani and M. Haley, “Unsupervised multi-task feature learning on point clouds,” in ICCV, 2019
- [19] J. Liu, B. Ni, C. Li, J. Yang, and Q. Tian, “Dynamic points agglomeration for hierarchical point sets learning,” in ICCV, 2019.
- [20] Y. Shen, C. Feng, Y. Yang, and D. Tian, “Mining point cloud local structures by kernel correlation and graph pooling,” in CVPR, 2018

REFERENCES

- [21] M. Dominguez, R. Dhamdhere, A. Petkar, S. Jain, S. Sah, and R. Ptucha, “General-purpose deep point cloud feature extractor,” in WACV, 2018
- [22] Chen, G. Li, R. Xu, T. Chen, M. Wang, and L. Lin, “ClusterNet: Deep hierarchical cluster network with rigorously rotationinvariant representation for point cloud analysis,” in CVPR, 2019
- [23] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, “Spectral networks and locally connected networks on graphs,” ICLR, 2014
- [24] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in NeurIPS, 2016
- [25] H. Lei, N. Akhtar, and A. Mian, “Seggcn: Efficient 3d point cloud segmentation with fuzzy spherical kernel,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11 611–11 620.
- [26] G. Te, W. Hu, A. Zheng, and Z. Guo, “RGCNN: Regularized graph CNN for point cloud segmentation,” in ACM MM, 2018
- [27] R. Li, S. Wang, F. Zhu, and J. Huang, “Adaptive graph convolutional neural networks,” in AAAI, 2018
- [28] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, “Hypergraph neural networks,” in AAAI, 2019
- [29] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning. arXiv preprint arXiv:1803.05827, 2018.
- [30] Y. Zhang and M. Rabbat, “A Graph-CNN for 3D point cloud classification,” in ICASSP, 2018
- [31] G. Pan, J. Wang, R. Ying, and P. Liu, “3DTI-Net: Learn inner transform invariant 3D geometry features using dynamic GCN,” arXiv preprint arXiv:1812.06254, 2018
- [32] Artem Komarichev, Zichun Zhong, and Jing Hua. A-CNN: Annularly convolutional neural networks on point clouds. In CVPR, 2019.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016

- [34] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293-306, 1985.
- [35] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. The farthest point strategy for progressive image sampling. *Transactions on Image Processing*, 6(9):1305-1315, 1997.
- [36] C. Moenning and N. A. Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003
- [37] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016
- [38] J. Salamon, B. McFee, and P. Li, “DCASE 2017 submission: Multiple instance learning for sound event detection,” *DCASE2017 Challenge*, Tech. Rep., 2017.
- [39] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, “Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling,” in *CVPR*, 2020.
- [40] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng Ann Heng. PU-Net: Point Cloud Upsampling Network. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2799, 2018.
- [41] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. *arXiv preprint arXiv:1812.01659*, 2018.
- [42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.
- [43] Liu, F., Ren, X., Zhang, Z., Sun, X., and Zou, Y., “Rethinking Skip Connection with Layer Normalization in Transformers and ResNets”, 2021.
- [44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 4-9 December 2017, Long Beach, CA, USA, pages 6000–6010

- [45] George Philipp, Dawn Song, and Jaime G. Carbonell. Gradients explode-deep networks are shallow-resnet explained. In ICLR Workshop, 2018.
- [46] Chaoning Zhang, Francois Rameau, Seokju Lee, Junsik Kim, Philipp Benz, Dawit Mureja Argaw, Jean-Charles Bazin, and In So Kweon. Revisiting residual networks with nonlinear shortcuts. In BMVC, 2019
- [47] Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. How does batch normalization help optimization? In Advances in Neural Information Processing Systems, pp. 2488–2498, 2018.
- [48] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In CVPR, pages 1912–1920, 2015.
- [49] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, MinhKhoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In International Conference on 3D Vision (3DV), 2016. <http://www.scenenn.net>.
- [50] S. Zagoruyko and N. Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016
- [51] Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In Advances in Neural Information Processing Systems, pages 2413–2421, 2015.
- [52] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In ICCV, 2019.
- [53] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richlyannotated 3d reconstructions of indoor scenes. In CVPR, 2017.
- [54] Berkman, J. and T. Caelli. “Computation of Surface Geometry and Segmentation Using Covariance Techniques.” IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994): 1114-1116.

- [55] R. Rusu. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009
- [56] Shakarji, Craig. (1998). Least-Squares Fitting Algorithms of the NIST Algorithm Testing System. *Journal of Research of the National Institute of Standards and Technology*. 103. 633. 10.6028/jres.103.043.
- [57] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5376–5385, 2020.
- [58] Y. Ben-Shabat, M. Lindenbaum and A. Fischer, "3DmFV: Three-Dimensional Point Cloud Classification in Real-Time Using Convolutional Neural Networks," in *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3145-3152, Oct. 2018, doi: 10.1109/LRA.2018.2850061.
- [59] Q. Xu, X. Sun, C.-Y. Wu, P. Wang, and U. Neumann, "Grid-gcn for fast and scalable point cloud learning," in *CVPR*, 2020
- [60] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters
- [61] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, et al. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics*, 35(6):210, 2016.
- [62] Jiaxin Li, Ben M Chen, and Gim Hee Lee. SO-Net: SelfOrganizing Network for Point Cloud Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018
- [63] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. *CVPR*, 2019
- [64] Z.-H. Lin, S.-Y. Huang, and Y.-C.-F. Wang, "Convolution in the cloud: Learning deformable kernels in 3D graph convolution networks for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1800–1809.

- [65] E. Nezhadarya, E. Taghavi, R. Razani, B. Liu and J. Luo, "Adaptive Hierarchical Down-Sampling for Point Cloud Classification," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 12953-12961, doi: 10.1109/CVPR42600.2020.01297.
- [66] points. In NeurIPS. 2018. 2, 4, 5, 6, 15, 16 [26] Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. Fpconv: Learning local flattening for point convolution. CVPR, 2020.
- [67] Liu, Xinhai et al. "Point2Sequence: Learning the Shape Representation of 3D Point Clouds with an Attention-based Sequence to Sequence Network." AAAI (2019).
- [68] Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., Pan, C.: Densepoint: Learning densely contextual representation for efficient point cloud processing. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (October 2019)
- [69] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In CVPR, 2019.
- [70] J. Mao, X. Wang and H. Li, "Interpolated Convolutional Networks for 3D Point Cloud Understanding," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1578-1587, doi: 10.1109/ICCV.2019.00166.
- [71] L. Pan, P. Wang and C. -M. Chew, "PointAtrousNet: Point Atrous Convolution for Point Cloud Analysis," in IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 4035-4041, Oct. 2019, doi: 10.1109/LRA.2019.2927948.
- [72] Z. Zhang, B. Hua and S. Yeung, "ShellNet: Efficient Point Cloud Convolutional Neural Networks Using Concentric Shells Statistics," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1607-1616, doi: 10.1109/ICCV.2019.00169.
- [73] Shi Qiu, Saeed Anwar, and Nick Barnes. Dense-resolution network for point cloud classification and segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 3813–3822, January 2021
- [74] 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 ICCV, 2019.

- [75] H. Lei, N. Akhtar and A. Mian, "Octree Guided CNN With Spherical Kernels for 3D Point Clouds," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9623-9632, doi: 10.1109/CVPR.2019.00986.
- [76] K. Fujiwara and T. Hashimoto, "Neural Implicit Embedding for Point Cloud Analysis," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11731-11740, doi: 10.1109/CVPR42600.2020.01175.
- [77] Y. Duan, Y. Zheng, J. Lu, J. Zhou and Q. Tian, "Structural Relational Reasoning of Point Clouds," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 949-958, doi: 10.1109/CVPR.2019.00104.
- [78] Y. Rao, J. Lu and J. Zhou, "Spherical Fractal Convolutional Neural Networks for Point Cloud Recognition," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 452-460, doi: 10.1109/CVPR.2019.00054.
- [79] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015.
- [80] Timo Hackel, N. Savinov, L. Ladicky, Jan D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pages 91–98, 2017.
- [81] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multiview convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, 2015.
- [82] C. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. 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.
- [83] H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. G. Kim, and E. Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Transactions on Graphics*, 37(1):6, 2018.
- [84] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao. GVCNN: Group-view convolutional neural networks for 3D shape recognition. In *CVPR*, pages 264–272, 2018.

REFERENCES

- [85] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu. Multi-view 3D object retrieval with deep embedding network. *IEEE Trans. Image Processing*, 25(12):5526–5537, 2016.
- [86] G. Riegler, A. Ulusoy, and A. Geiger. OctNet: Learning deep 3D representations at high resolutions. In *Proceedings of the 9 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017.
- [87] Shiyi Lan, Ruichi Yu, Gang Yu, and Larry S Davis. Modeling local geometric structure of 3D point clouds using GeoCNN. In *CVPR*, 2019
- [88] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong. Adaptive OCNN: A patch-based deep representation of 3D shapes. *ACM Transactions on Graphics*, 37(6), 2018.
- [89] Wenxiao Zhang and Chunxia Xiao. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *CVPR*, 2019
- [90] B.-S. Hua, M.-K. Tran, and S.-K. Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.
- [91] F. Yu, K. Liu, Y. Zhang, C. Zhu, and K. Xu, “PartNet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation,” in *CVPR*, 2019.
- [92] M. Gadelha, R. Wang, and S. Maji. Multiresolution tree networks for 3D point cloud processing. In *ECCV*, pages 105– 122, 2018.
- [93] Yecheng Lyu, Xinming Huang, and Ziming Zhang. Learning to segment 3D point clouds in 2D image space. In *CVPR*, 2020.
- [94] Y. Lin, Z. Yan, H. Huang, D. Du, L. Liu, T. Chinese, and H. Kong. FPConv: Learning Local Flattening for Point Convolution. *CVPR*, 2020.
- [95] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. *arXiv preprint arXiv:2103.14635*, 2021.