FGCN: Deep Feature-based Graph Convolutional Network for Semantic Segmentation of Urban 3D Point Clouds

Saqib Ali Khan[1], Yilei Shi[2], Muhammad Shahzad[1], Xiao Xiang Zhu[3,4]
[1]School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST), Islamabad, Pakistan
[2]Chair of Remote Sensing Technology (LMF), Technical University of Munich (TUM), Munich, Germany
[3]Signal Processing in Earth Observation (SiPEO), Technical University of Munich (TUM), Munich, Germany
[4]Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Wessling, Germany
{sakhan.bscs16seecs;muhammad.shehzad}@seecs.edu.pk, yilei.shi@tum.de,xiaoxiang.zhu@dlr.de

Abstract

Directly processing 3D point clouds using convolutional neural networks (CNNs) is a highly challenging task primarily due to the lack of explicit neighborhood relationship between points in 3D space. Several researchers have tried to cope with this problem using a preprocessing step of voxelization. Although, this allows to translate the existing CNN architectures to process 3D point clouds but, in addition to computational and memory constraints, it poses quantization artifacts which limits the accurate inference of the underlying object's structure in the illuminated scene. In this paper, we have introduced a more stable and effective end-to-end architecture to classify raw 3D point clouds from indoor and outdoor scenes. In the proposed methodology, we encode the spatial arrangement of neighbouring 3D points inside an undirected symmetrical graph, which is passed along with features extracted from a 2D CNN to a Graph Convolutional Network (GCN) that contains three layers of localized graph convolutions to generate a complete segmentation map. The proposed network achieves on par or even better than state-of-the-art results on tasks like semantic scene parsing, part segmentation and urban classification on three standard benchmark datasets.

1. Introduction

With recent successes of convolutional neural network (CNN) architectures in processing 2D structured data, there is an increasingly growing interest of researchers in developing similar architectures to directly process 3D point clouds. For instance, there has been many attempts to extend the traditional CNNs [18, 22, 24, 27], that are best fit for data that lie in a structured Euclidean space to 3D point clouds. However, 3D datasets do not lie on a regular grid and thus lacks the implicit neighborhood relationship. Owing to this, there does not exist a single well-defined notion that enables convolution on unstructured 3D data. Furthermore, many approaches [18, 29, 23] transform the 3D datasets into regular 3D structures like voxels and meshes to apply convolution, but the transformed regular structures loses most of the spatial information that lies between neighbouring points and thus struggles to obtain the local feature representations that can improve the overall classification results [33].

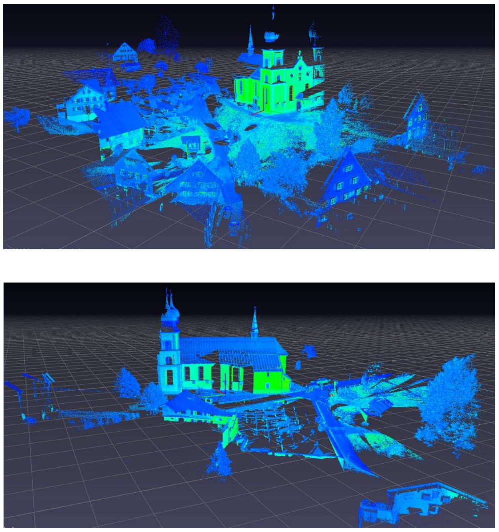

Figure 1. **Examples of outdoor scenes from Semantic3D benchmark dataset** [10]. Our architecture assigns a correct semantic label to each object with on par state-of-the-art accuracy. The results are visualized using PPTK viewer. Best viewed in color.

To encode the neighbourhood relationships, few researchers have used graph representations to capture the local features more effectively. In this context, Bronstein *et al*. [3] first used the term geometric deep learning and gave an overview of the deep learning methods for datasets that lie in non-Euclidean domain. However, the first prominent research that defines convolutional GNN in a spectral domain was given by Bruna *et al*. [4]. They have provided evidence of the possible generalizations of CNNs to signals in other domains without taking 3D translational factors into account. Defferrard *et al*. [6] proposed a generalized formulation of CNNs for spectral graphs. Their approach used the recursive form of Chebyshev polynomials to propose a fast convolution for high-dimensional unstructured datasets such as social networks or protein-interaction networks. Furthermore, it is sometimes desirable to use a kernel-based approach [17, 30]. This property of using graph-kernels is favourable because the local structure of the graph contains meaningful information. However, kernel-based approaches are computationally expensive and have quadratic training complexity.

Inspired by the idea of graph based representation to propagate local features, we have used a Graph Convolutional Network (GCN) to encode spatial information or local neighbourhood features into symmetrical graph models. In the proposed 3D representation, each point is represented by three coordinates (x, y, z). In addition to our local feature encoder or GCN, we have used a global feature extractor similar to [22], that extracts a vector of high dimensional features by taking the raw point cloud as input. Using the global features, summarizes most of the information and provides geometric invariance [22] that increases the overall performance and reliability of our network (See Section 5 for details). The graph convolution refines these high order features using the local spatial features from graph representation and outputs a global signature summarizing each point inside the graph. Therefore, our proposed architecture learns the complete local structure embedded in the graph to achieve faster convergence and better classification results. Our GCN or spatial-temporal graph neural network [33] achieves on par or even better results compared to state-of-the-art architectures. Specifically, following are the main contributions proposed in this work:

- A novel graph based convolutional network has been proposed that uses both local and global features for semantic segmentation of 3D point clouds;
- It is evidently showed how using the spatial information in the local neighbourhood of points in 3D space offers stability and increased performance;
- The proposed architecture been compared with the state-of-the-art approaches and achieved competitive performance on three standard benchmark datasets including S3DIS [1], ShapeNet [35], and Semantic3D [10] datasets. For reference, Figure 1 provides the visualization of two different outdoor scenes.

2. Related Work

Deep Learning on 3D Point Clouds Many approaches utilize 3D shapes to apply deep learning, for example *Volumetric CNNs* [23, 38, 21], is the pioneer work that applies 3D convolutions on voxelized shapes. However, Volumetric CNNs have a higher computational cost due to the sparsity of 3D data in volumetric representations. This problem has been addressed through careful engineering of CNNs [20, 31]. However, the problem still persists due to significantly sparse volumes in very large point clouds. *Multiview CNNs* [28], integrate multiple views of a 3D point cloud together and apply 2D convolution for classification. With efficient 2D convolutions, they can process very high resolution data. Furthermore, these architectures can achieve state-of-the-art results in object classification on datasets like ModelNet [38], but they cannot be extended to more complex tasks like 3D scene understanding.

Recently, many new approaches have been proposed that directly consume raw 3D point clouds and are used for tasks like semantic segmentation, object classification and detection etc. *PointNet* [22] is the pioneer work that applies deep learning on raw 3D point clouds with significant improvements in performance. However, PointNet does not generalize well on complex scenes due to its inability to capture the local structure induced by the 3D space. The local structure is exploited by *PointNet++* [24], which is an extension of PointNet. In their proposed methodology, they were able to capture the local features with increasing contextual scales. *SPLATNet* [27], sparse lattice networks, used bilateral convolutions as building blocks to apply 3D convolution only on the occupied parts of the lattice that reduces memory and computational cost. *PointConv* [32] uses dynamic filters to apply convolution on point clouds. They treat convolutional kernels as non-linear functions of the point coordinates comprised of density and weight functions.

Deep Learning on Graphs or spectral CNNs were first introduced by [4] and extended by [6]. Many approaches like ours that applies convolution in a spectral domain uses ideas from graph signal processing [26] to apply localized filters on graphs. Recently, many approaches [6, 15, 37] approximate the spectral convolution using Chebyshev polynomials, because transforming the signal back and forth between spectral domains can be expensive. Our approach uses Chebyshev polynomials for spectral convolutions in a similar way as [37, 26].

3. Proposed Methodology

Suppose, we are given a set of m training examples $\{X_m, Y_m\}$ with $X_i = \{P_j | j = 1......n\}$, where n is the number of points $P \subset \mathbb{R}^3$ in X_i, and $Y_m = \{1......n\}$ is the associated semantic label of each point P_j in the i^{th} training example. Furthermore, each point P_j in X_i consists of a vector of 3D coordinates (x, y, z).

In our proposed methodology, we extend the traditional graph based convolutions [26, 37], that works on latent graph signals to output a global signature which is then used for classification. Most of these architectures, overlook the underlying spatial information between points inside a 3D space, which plays a crucial role in identifying objects. Keeping in mind the importance of local features, we propose a unified architecture that jointly use both local and global features to give a more stable and reliable network for semantic segmentation of 3D point clouds. Using the global feature extractor before graph convolutional network summarizes most of the information and provides geometric invariance [22] which in turn increases the overall performance or our network. In the following sections, we will explain the key components of our proposed architecture and will provide evidence as to how using both local and global features can give better results.

3.1. Transforming 3D Point Sets to Weighted Graph Signals

A graph convolutional network performs convolution on input that is supported on a graph $G = \{V, E, W\}$, with a finite number of nodes $v_i \in V$, edges $e_{ij} = \{v_i, v_j\} \in E$, and $W_{i,j} \in W$ corresponding to the weighted graph signal or an entry into the adjacency matrix indicating a connection between v_i and v_j. In order to find the value of $W_{i,j}$, we find all the neighbouring nodes of node i using k-nearest neighbors, and then use a Gaussian kernel to weight the edge $e_{i,j}$ connecting node i and a neighbouring node j:

$$W_{i,j} = \begin{cases} \exp(-\frac{\|v_i - v_j\|^2}{2\sigma^2}) & \text{if } \|v_i - v_j\| < \kappa \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

for some value of $\sigma > 0$ and parameter κ. In equation 1, $\|v_i - v_j\|$ represent the Euclidean distance between two feature vectors of node $v_i = \{x_i, y_i, z_i\}$ and node $v_j = \{x_j, y_j, z_j\}$, with node v_j as a neighbor of node v_i.

Given the undirected graph with adjacency matrix $W \in \mathbb{R}^{N \times N}$, we apply graph filtering techniques [15, 37] using normalized Laplacian matrix $L = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where D corresponds to the diagonal matrix in which $D_{ij} = \Sigma_j \{W_{i,j}\}$. The normalized Laplacian matrix can also be interpreted using the eigenvectors as $L = U \Lambda U^T$, where U corresponds to the matrix of eigenvectors and Λ corresponds to the diagonal matrix of U. Let's restate our graph mapping function $f(x)$ with input x, as a linear graph filter transformation function with coefficients $\mu_1, \mu_2, ...\mu_n$ as,

$$f(x) = g_\mu(L)x = \sum_{i=0}^{K} \mu_i L^i x \quad (2)$$

The mapping function $f(x)$ can also be approximated using the eigen decomposition form of normalized Laplacian matrix with eigenvalues Λ as,

$$f(x) = g_\mu(L)x = U g_\mu(\Lambda) U^T x \quad (3)$$

Spectral based graph filtering methods [12, 7, 26] also use Chebyshev polynomials to approximate graph filters. ChebyNet [7] uses the diagonal matrix of eigen values,

$$f(x) = g_\theta(L)x = \sum_{i=0}^{K} \theta_i T_i(L)x \quad (4)$$

Additionally, equation 4 can also be defined recursively with $T_0(x) = 1$ and $T_1(x) = x$ as,

$$T_i(x) = 2xT_{i-1} - T_{i-2}(x) \quad (5)$$

The goal of graph convolutional layer is to learn a set of graph filtering coefficients $\{\mu\}$ or $\{\theta\}$ using any type of graph filtering method. However, using the normalized Laplacian with eigen decomposition has a high computational cost compared to ChebyNet [7]. Furthermore, Defferrard *et al.* [6] demonstrated the effectiveness of using Chebyshev graph filtering approximation (graph convolution) on homogeneous graphs, for tasks like image classification and 2D scene understanding. We adapt a similar approach to [7], using the Chebyshev polynomials as a graph filtering method, but in our approach we have applied convolution on heterogeneous graphs with global features (extracted from 2D convolutional layers) as input.

3.2. Model Architecture

Our segmentation network consists of three main modules: 1) Feature extraction that inputs the $N \times 3$ dimensional point coordinate vector and outputs an $N \times D$ dimensional global feature vector; 2) Graph signal processing that also takes an $N \times 3$ dimensional coordinate vector as input and outputs a weighted graph in the form of an adjacency matrix W; 3) Graph convolutional network with learnable parameter θ of order k, takes as input the $N \times D$ dimensional feature vector along with weighted graph signals W and extracts the local features corresponding to the spatial arrangement of nodes in the graph, which is then passed to fully connected layers for per-point classification. The architecture diagram can be visualized in figure 2.

3D Feature Extraction Many techniques have been developed in order to obtain global feature descriptors for 3D point sets [13, 22, 14, 8]. Johnson *et al.* [14] developed a

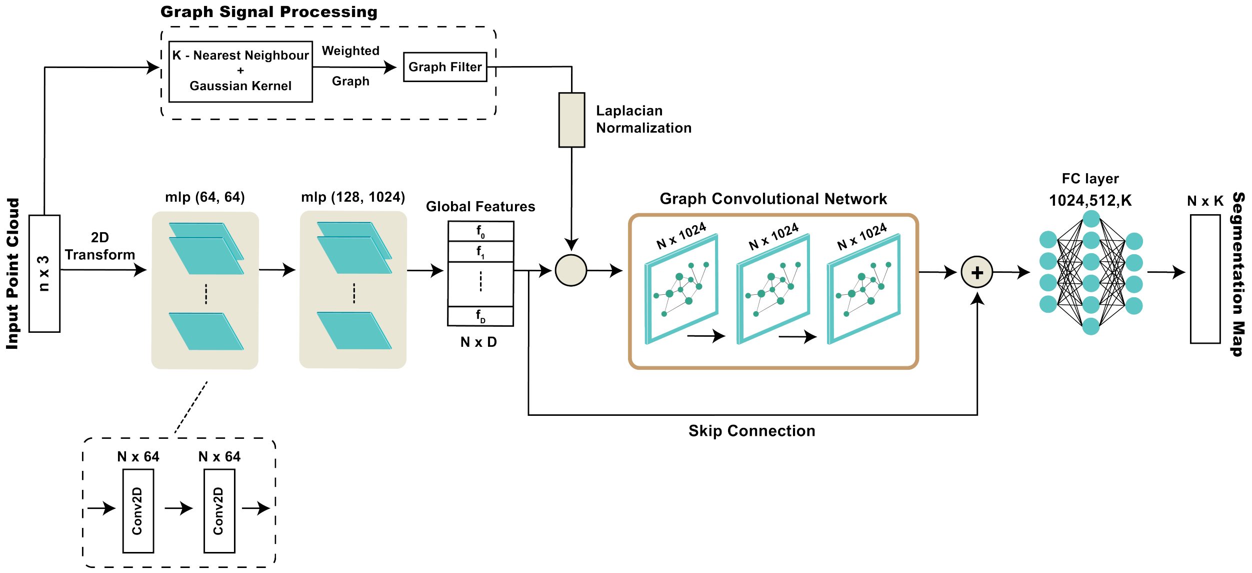

Figure 2. **Network Architecture**: The network takes as input N points with coordinates (x, y, z). The input is passed to graph signal processing module to generate a re-scaled normalized graph vector and is also passed to deep convolutional feature extraction layers to output a global feature vector $N \times D$. Both the normalized weighted graph and global features goes as input to graph convolutional network to output a global feature signature which is passed to a fully connected layer that scales down the features and assign one of k output classes to each point. GCN uses ReLU activation function and dropout regularization after each layer.

method to extract local feature descriptors from 3D point sets called spin images. The distance (α, β) between a feature point in spin image with coordinate p, a surface normal n and a neighbouring point q is given by $\alpha = n_q.(p-q)$ and $\beta = \sqrt{\|p-q\|^2 - \alpha^2}$. The final spin image contains the neighbors of feature points accumulated in a discontinuous 2D bin which is robust to occlusion and clutter. Flint *et al.* [8] propose a method called THRIFT that extends the feature extraction techniques applied to 2D images like SIFT and propose a 3D feature descriptor that successfully identifies keypoints in range data.

Recently, convolutional neural networks have been used in general for feature extraction in both 2D and 3D domains. The most recent work that employ CNNs to extract global features from raw 3D point clouds is PointNet [22]. PointNet architecture uses a stack of 2D convolutional layers for feature transformation and ensures invariance to permutations, geometric transformations and also considers the interaction among points using a localized convolution operation. PointNet outperformed all the existing methods used for classification of 3D points which either required conversion to other irreversible representations [23, 38, 21] or used raw 3D point clouds [18].

In this paper, we take motivation from PointNet [22] and extend our graph convolutional network to be more robust using global features. So, instead of taking the point coordinates $(x^{(i)}, y^{(i)}, z^{(i)})$ as input feature vectors [37], we use 2D convolutional layers to output an $\{x_i^{(1)}, x_i^{(2)}, ...x_i^{(D)}\} \in \mathbb{R}^{N \times D}$ global feature vector, where D represents the number of features per point.

Using the global feature extraction with graph convolutional network speeds up the training process and increases the overall performance of our network which is demonstrated in Sections 4 and 5.

Graph Convolutional Network (GCN) takes as input the feature vector $\{x_i^{(1)}, x_i^{(2)}, ...x_i^{(D)}\} \in \mathbb{R}^{N \times D}$, where D corresponds to the number of features and the weighted graph signals $W \in \mathbb{R}^{N \times N}$, and the goal of GCN is to learn a set of K trainable graph-filter coefficients. Moreover, a GCN learns a mapping function that can translate the input graph signals to capture the local features corresponding to the relative position of points in 3D space. So, a GCN can be written as a non-linear function σ of input graph signals $W^{(l)}$ and $X^{(l)}$, where l corresponds to the activations of l^{th} layer.

$$f(X^{(l)}, W) = \sigma\left(\theta^{(l)} X^{(l)} W\right) \tag{6}$$

where the learnable parameter θ is of order K. The mapping function in equation 6 contains an unnormalized graph representation W, because the range of values can vary for heterogeneous graphs, the unnormalized GCN cannot generalize well on graphs that lie in different spectral domains [33]. In order to overcome this problem, the input graph signal is to be normalized in such way that adding all the rows of W sum to one [15]. In our proposed methodology, we have used a graph Laplacian $L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ using the diagonal matrix D such that $D_{ii} = \sum_j W_{ij}$ for symmetric normalization,

$$f(X^{(l)}, W) = \sigma\,(\theta^{(l)} \widehat{D}^{-\frac{1}{2}} \widehat{W} \widehat{D}^{-\frac{1}{2}} X^{(l)})) \tag{7}$$

where $\widehat{W} = W + I$, and I is the identity matrix. Furthermore, using the Laplacian normalization, the eigenvalues of L lie in the range $[-1, 1]$.

In order to obtain the local features at each layer l, we use Chebyshev polynomials 4, and take as input the global feature vector $\{x_i^{(1)}, x_i^{(2)}, ...x_i^{(D)}\}$ for the first layer. Furthermore, in order to define a single graph convolution operation between the input feature vector x_i and a graph signal g, we use the inverse graph Fourier transform [33] as,

$$x *_G g = U(U^T x \odot U^T g) \tag{8}$$

where U is the matrix of eigenvectors and $\odot$ represents the pointwise product of inverse graph Fourier transform of x as $U^T x$ and g as $U^T g$.

In our proposed architecture, we have used the Chebyshev graph filtering representation given by equation 4, with K-neighbourhood at each point to learn the localized feature maps with three layers of graph convolutions.

3.3. Training

The architecture is trained using Adam optimizer with a learning rate that starts at 1×10^{-3} and is reduced to half after every 20 epochs, but always stays in the range $[1 \times 10^{-3}, 1 \times 10^{-7}]$. We have used a batch size of 16 and dropout regularization of 0.8 for GCN layers and 0.4 for fully connected layers to prevent overfitting. Our network uses four layers of 2D convolutional layers with kernel sizes $[64, 64, 128, 1024]$ respectively. Furthermore, to avoid additional complexity in our model, we have used a weight decay of magnitude 2×10^{-4}.

The speed and stability of GCN depends heavily on the order K of Chebyshev polynomial 4. The model performs optimal at $K = 1$, and as we increase the order of K, the size of $T_i(L)$ increases which diminishes the speed and increases the time required to train the network.

4. Performance Measures

We have evaluated our architecture on a variety benchmark datasets including S3DIS containing indoor 3D scenes[1], ShapeNet part segmentation [35] and Semantic3D benchmark dataset [10]. Our methodology, outperforms the existing architectures on all the benchmarked datasets, and most of the performance gain is due to encoding the local spatial features of the 3D point cloud inside a graph model.

Method	mean IOU	mean Accuracy
PointNet [22]	47.71	48.98
SEGCloud [29]	48.92	57.35
Ours (GCN Only)	47.22	56.44
Ours (FGCN)	**52.17**	**63.22**

Table 1. **Results of Semantic scene parsing** on Stanford 3D dataset. The mIOU is calculated as an average over IOUs of all 13 classes containing indoor structural objects.

	class average
SSCNN [36]	82.0
Kd-net [16]	77.4
PointNet [22]	80.4
PointNet++ [24]	81.9
SpiderCNN [34]	82.4
SPLATNet$_{3D}$ [27]	82.0
PointConv [32]	82.8
Ours (GCN Only)	**78.2**
Ours (FGCN)	**83.1**

Table 2. **Results on ShapeNet part segmentation**: The metric is mIOU similar to the one used by PointNet [22]. We have compared our architecture with existing architectures on ShapeNet part segmentation. Our network achieves slightly better results than state-of-the-art.

4.1. Semantic Scene Parsing

In our first experiment, we have used Stanford 3D dataset [1], that contains 3D scans from 6 different areas and 271 rooms collectively acquired using an individual Matterport Scanner. The dataset contains 13 classes, so each point can be assigned 1 out of 13 semantic labels.

In order to split the data into training and testing sets, we have used the same method and statistics as used by PointNet [22]. We first divide the areas into rooms and then split points in each room using 1m by 1m blocks. Furthermore, each point contains a 9-dimensional vector containing XYZ coordinates, RGB color channels and a normal or an equirectangular projection per room.

We train our model using a point size N of 4096 per training example and a batch size of 16, where each point contains only the XYZ coordinates. The comparison between our architecture and existing architectures on S3DIS dataset is shown in table 1, and the results can be visualized in figure 3. Our methodology outperforms the existing architectures by a significant margin.

4.2. ShapeNet Part Segmentation

ShapeNet [35] provides a large-scale repository that contains richly annotated 3D shapes. The ShapeNet part dataset from [35] contains $16,881$ 3D shapes from 16 different categories, labelled with 50 parts in total. In object's part seg-

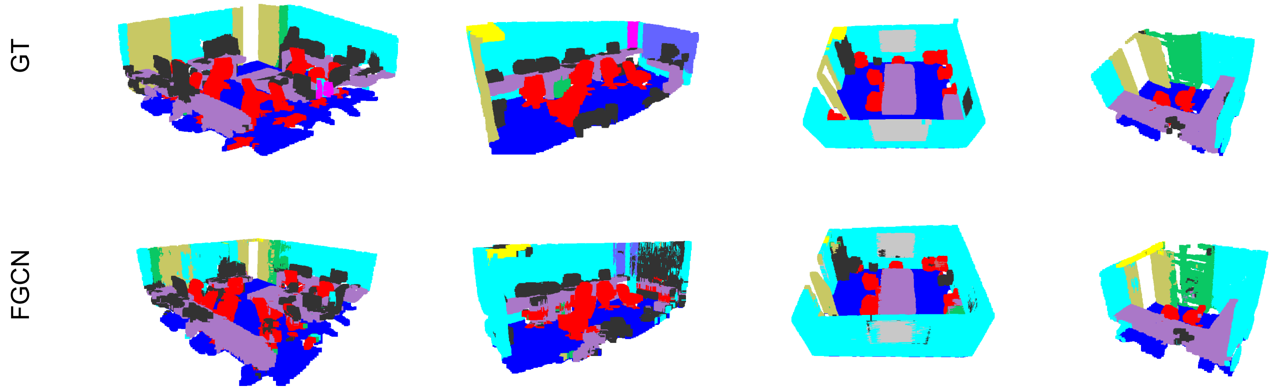

Figure 3. **Qualitative results on semantic scene parsing.** The images on the top contains the ground truth labels and on the bottom are the predictions by FGCN on the Stanford's indoor semantic scene parsing dataset [1]. The point clouds are viewed using MeshLab software. Best viewed in color.

mentation, the goal is to assign a correct semantic label to each point of the 3D shape, where the labels are salient regions or functional parts of the objects like wing, engine, tail, handle, roof etc.

In order to evaluate our model on ShapeNet part dataset we pre-compute the Graph filters using Chebyshev polynomials 4 and train our model on each of the 16 object categories. Furthermore, for a fair comparison we have used the same evaluation metric as used by PointNet [22]. We compute the intersection-over-union (IOU) over each object category and then compute the mIOU by averaging the IOUs of each individual category.

The results are shown in table 2, we have used a batch size of 16 and the Chebyshev order or $K = 1$ for graph filtering. We have compared our methodology with existing architectures that directly consume raw 3D point clouds, and have achieved a class average of 83.1 which is on par with state-of-the-art.

4.3. Semantic3D Benchmark

There has been a long tradition of benchmark evaluation in the geospatial dataset domain particularly ISPRS. For example, the *ISPRS-EuroSDR benchmark on High Density Aerial Image Matching*, which evaluates dense matching algorithms [9, 5] on aerial imagery. The ISPRS *Benchmark on Urban Object Detection and Reconstruction* that contains a variety of challenges including object detection, semantic segmentation and 3D reconstruction of geospatial aerial imagery [25].

In this paper, we have used the Semantic3D benchmark dataset [10] for evaluating our architecture. This dataset is the most recent, and by far the largest labeled 3D point cloud dataset of outdoor scenes containing both urban and rural environments like villages, churches, railway roads, squares, streets etc. It contains nearly 4 billion points collected with 30 terrestrial laser scanners across Central Europe depicting the European architecture in most of its scenes. The results shown in table 3 are on the *reduced-8* dataset of the benchmark that has the following 8 classes: 1) natural terrain; 2) buildings; 3) low vegetation; 4) high vegetation; 5) man made terrain; 6) scanning artifacts; 7) cars and trucks; and 8) remaining hard scape.

Additionally, Semantic3D [10] benchmark proposed a baseline 3D-CNN architecture for 3D point cloud classification that takes as input 3D voxel-grids per scan point at 5 different resolutions. Their pipeline uses VGG-like architecture that uses 3D convolutions with softmax layers to output per-point classifications of the 3D point cloud. However, their approach converts the original 3D point clouds to voxel representations that renders the input dataset highly voluminous and increases the overall computation cost of the network. Following this approach, many architectures were evaluated on Semantic3D dataset, SEGCloud [29] is an end-to-end architecture that jointly uses the advantages of fully Convolutional Neural networks (FCN), trilinear interpolation (TI), and fully connected Conditional Random Fields (FC-CRF) to provide fine grained semantics per point inside a 3D point cloud. SEGCloud outperformed existing architectures on Semantic3D benchmark by a significant margin, nearly 2.2 mIOU points and 2.28% increase in accuracy.

We evaluate our architecture on Semantic3D benchmark dataset using the similar intersection over union (IOU) metric as defined in [10], and the results are shown in table 3. The mIOU is calculated as an average over IOUs of all 8 classes. Our architecture achieves on par state-of-the-art

category	TMLC-MSR [11]	DeePr3SS [19]	SnapNet [2]	SEGCloud [29]	Ours (GCN only)	FGCN
man-made terrain	89.80	85.60	82.00	83.90	79.20	**90.30**
natural terrain	74.50	**83.20**	77.30	66.00	62.10	65.20
high vegetation	53.70	74.20	79.70	86.00	82.30	**86.20**
low vegetation	26.80	32.40	22.90	**40.50**	36.20	38.70
buildings	88.80	89.70	**91.10**	**91.10**	86.20	90.10
hard scape	18.90	18.50	18.40	30.90	**34.70**	31.60
scanning artefacts	36.40	25.10	**37.30**	27.50	29.00	28.80
cars	44.70	59.20	64.40	64.30	66.40	**68.20**
mean IOU	54.20	58.50	59.10	61.30	59.50	**62.40**
mean Accuracy	68.95	88.90	70.80	73.68	76.80	**89.30**

Table 3. **Results on Semantic3D *reduced-8* dataset:** The mIOU and mAcc are calculated as mean over all categories of Semantic3D dataset. Our approach achieves state-of-the-art results on Semantic3D benchmark dataset.

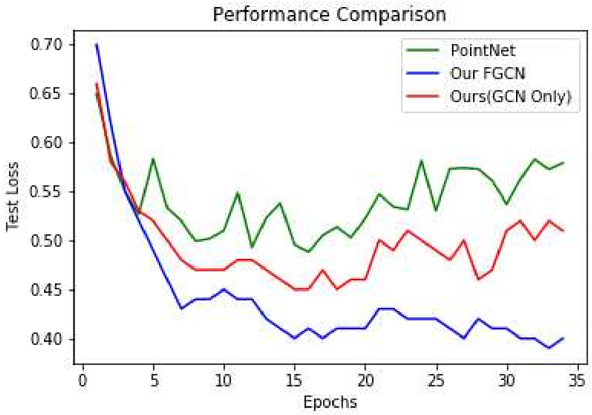

Figure 4. **Test Loss comparison on S3DIS dataset [1].** The comparison is between PointNet [22] and our proposed architectures. The graph indicates a faster convergence rate and a more stable learning curve for our approach. Best viewed in color.

performance on Semantic3D benchmark dataset and combining the local spatial information with global features inside a GCN accounts for most of the performance increase. The results can be visualized in figure 5.

5. Architecture Design Goals

In this section, we evaluate the performance of our architecture with respect to speed and stability using S3DIS [1] dataset. We also show the effect of using local feature extraction and how adding the global features to our network gives best performance for our network.

Effect of Using Local Feature Extraction: Many approaches transform the input 3D point cloud to a structured 3D form [18, 29, 23, 38, 21] losing most of the spatial information that is beneficial for identifying objects inside a 3D space. Recently, the interest is towards consuming the point clouds directly [22, 24, 32, 27], but many of these architectures try hard to improve the local feature extractor by applying convolution directly to the unstructured point cloud. Consider figure 4, which shows the fluctuations in test loss during training on S3DIS dataset [1], because of the sensitivity to initial weights. This problem is especially severe for PointNet [22] that only seeks the latent representations with more emphasis on the overall global signature of the 3D object, without considering the meaningful local features that exists between points. On the other hand, our final architecture uses both global features (that also provides geometric invariance [22]) and local point features and thus has a relatively faster convergence rate and is more stable towards the unstructured nature of 3D point clouds. However, using only the local features is not sufficient because of varying geometry of 3D objects in different 3D scenes. (See figure 4, GCN Only)

Effect of Using Global Feature Extraction: One of the key problems in working with unstructured 3D data is the geometry of the 3D object. The semantic segmentation should be able to generalize well on all the possible permutations of 3D objects, and in order to do so we have used a stack of 2D convolutional layers to extract high order features given the 3D point coordinates [22]. Therefore, our final architecture reforms the raw 3D point cloud to a vector of high dimensional features before passing it on to the graph convolutional network. This adds to the overall stability and reliability of our model across different scenes with objects of varying geometries. (See Figure 4, FGCN). Additionally, our architecture also preserves the spatial position of points since we combine information using k-nearest neighbor with K hops and encode this information into graphs which are symmetric by default.

6. Conclusion

In this paper, we have presented *FGCN*, a novel feature based graph convolutional network for semantic segmentation of 3D point clouds. The proposed architecture achieves on par or even better performance than state-of-the-art approaches on tasks like semantic scene parsing, part segmen-

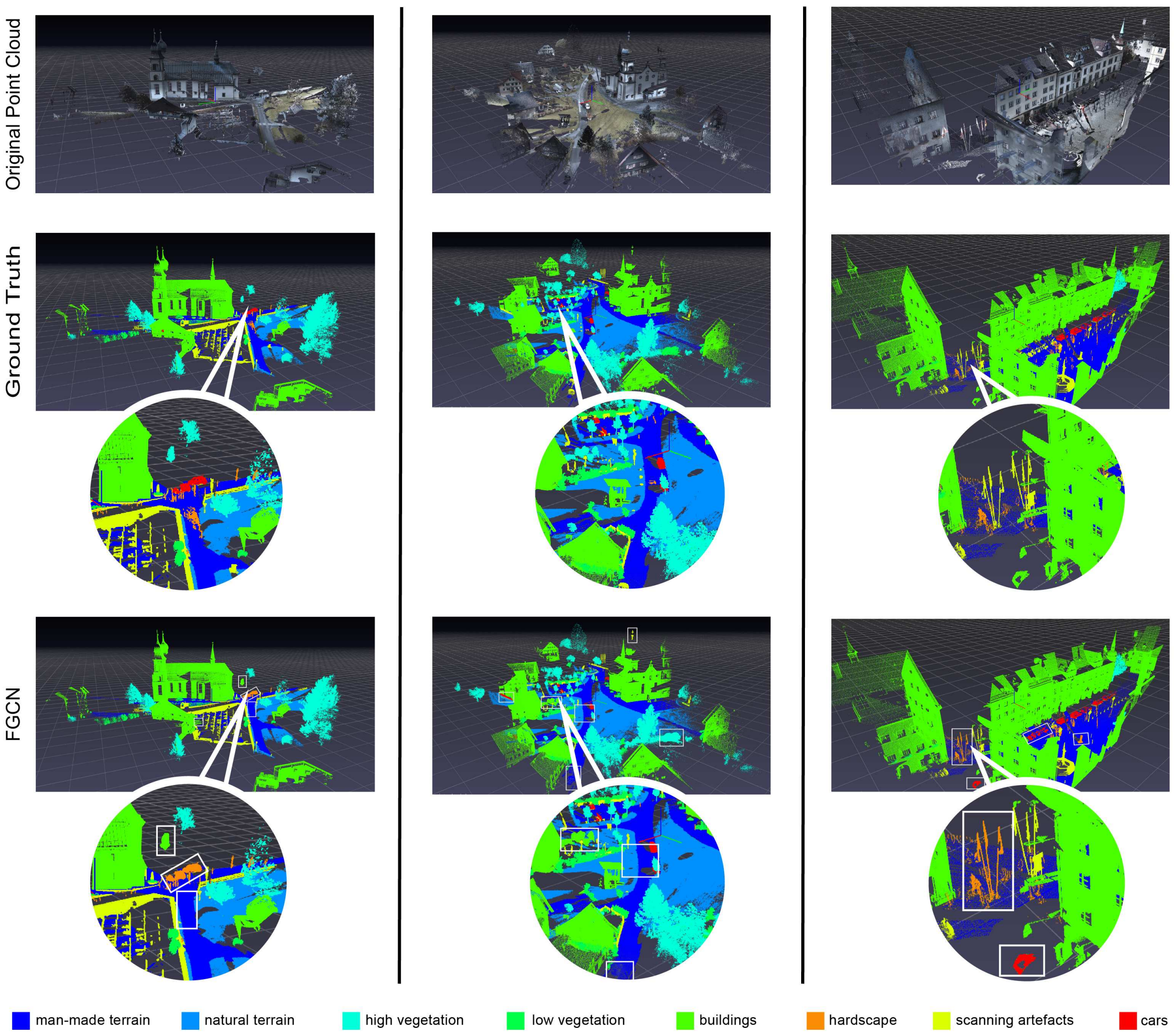

Figure 5. **Qualitative results of testing on Semantic3D benchmark dataset.** The point clouds are visualized using the PPTK viewer. In order to produce these visualizations, we have further reduced the training set by three examples (scenes). Furthermore, the network outputs a sparse prediction which we interpolate to produce a dense point cloud prediction using Open3D's k-NN hybrid search with radius of 0.2. The most prominent classification errors are indicated by the rectangles drawn on FGCN output. Best viewed in color.

tation and classification of objects in natural scenes such as in Semantic3D benchmark dataset. In this work, we have shown the importance of using local features and how using the spatial position of points can increase the overall performance of the segmentation task when it comes to identifying objects in 3D scenes. In addition to increased performance, the proposed architecture is invariant to geometric distortions and preserves the local structures of objects using the graph models. Although the proposed network achieves better results in terms of accuracy but requires more memory footprint compared to the existing architectures. In future, we intend to optimize the memory usage by using subgraphs (or using dynamic construction of subgraphs during training). In addition to optimizing the memory, we would develop a module for propagating global information between subgraphs which could add an extra boost in accuracy.

7. Acknowledgements

The authors would like to thank German Academic Exchange Service (DAAD) for supporting this work through Deutsch-Pakistanische Forschungskooperationen. 2j ab 19 grant number: 57459164.

References

[1] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints*, Feb. 2017. 2, 5, 6, 7

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

[3] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34:18–42, 2017. 2

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs, 2013. 2

[5] Stefan Cavegn, Norbert Haala, Stephan Nebiker, Mathias Rothermel, and Patrick Tutzauer. Benchmarking high density image matching for oblique airborne imagery. volume XL-3, 09 2014. 6

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016. 2, 3

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3844–3852, Red Hook, NY, USA, 2016. Curran Associates Inc. 3

[8] A. Flint, A. Dick, and A. v. d. Hengel. Thrift: Local 3d structure recognition. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, pages 182–188, Dec 2007. 3, 4

[9] Norbert Haala. The landscape of dense image matching algorithms. 2013. 6

[10] 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. 1, 2, 5, 6

[11] Timo Hackel, Jan Wegner, and Konrad Schindler. Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3:177–184, 06 2016. 7

[12] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, Mar. 2011. 3

[13] Xian-Feng Han, Jesse Jin, Juan Xie, Ming-Jie Wang, and Wei Jiang. A comprehensive review of 3d point cloud descriptors. 02 2018. 3

[14] Andrew Edie Johnson and Martial Hebert. Surface matching for object recognition in complex 3-d scenes. 1998. 3

[15] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. 2, 3, 5

[16] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. 04 2017. 5

[17] Risi Imre Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, page 315–322, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. 2

[18] Sudhakar Kumawat and Shanmuganathan Raman. Lp-3dcnn: Unveiling local phase in 3d convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 4, 7

[19] Felix Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Khan, and Michael Felsberg. Deep projective 3d semantic segmentation. pages 95–107, 05 2017. 7

[20] Yangyan Li, Sören Pirk, Hao Su, Charles R. Qi, and Leonidas J. Guibas. Fpnn: Field probing neural networks for 3d data. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 307–315, Red Hook, NY, USA, 2016. Curran Associates Inc. 2

[21] D. Maturana and S. 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, Sep. 2015. 2, 4, 7

[22] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3, 4, 5, 6, 7

[23] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 1, 2, 4, 7

[24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5099–5108. Curran Associates, Inc., 2017. 1, 2, 5, 7

[25] Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sébastien Bénitez, and U Breitkopf. The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3, 07 2012. 6

[26] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013. 2, 3

[27] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz.

SPLATNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2530–2539, 2018. 1, 2, 5, 7

[28] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*, 2015. 2

[29] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. *CoRR*, abs/1710.07563, 2017. 1, 5, 6, 7

[30] Matteo Togninalli, M. Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten M. Borgwardt. Wasserstein weisfeiler-lehman graph kernels. *CoRR*, abs/1906.01277, 2019. 2

[31] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, 2015. 2

[32] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 2018. 2, 5, 7

[33] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *ArXiv*, abs/1901.00596, 2019. 1, 2, 4, 5

[34] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. 03 2018. 5

[35] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016. 2, 5

[36] Li Yi, Hao Su, Xingwen Guo, and Leonidas Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. pages 6584–6592, 07 2017. 5

[37] Yingxue Zhang and Michael Rabbat. A graph-cnn for 3d point cloud classification. pages 6279–6283, 04 2018. 2, 3, 4

[38] Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, June 2015. 2, 4, 7