AGCN: Adversarial Graph Convolutional Network for 3D Point Cloud Segmentation

Seunghoi Kim
edshkim98@gmail.com
Daniel C. Alexander
d.alexander@ucl.ac.uk

Department of Computer Science
University College London
London, UK

Abstract

3D point cloud segmentation provides a high-level semantic understanding of object structure that is valuable in applications such as medicine, robotics and self-driving. In this paper, we propose an Adversarial Graph Convolutional Network for 3D point cloud segmentation. Many current networks encounter problems such as low segmentation accuracy and high complexities due to their crude network architectures and local feature aggregation methods. To overcome these problems, we propose a) a graph convolutional network (GCN) in an adversarial learning scheme where a discriminator network provides a segmentation network with informative information to improve segmentation accuracy and b) a graph convolution, GeoEdgeConv, as a means of local feature aggregation to improve segmentation accuracy and space and time complexities. By using an embedding L_2 loss as an adversarial loss, the proposed network is learned to reduce noisy labels by enforcing the consistency between neighbouring labels. Preserving geometric structures over convolution layers by using both point and relative position features, GeoEdgeConv helps learn fine details of complex structures, and thus improves segmentation accuracy in boundaries and reduces label noise inside a class without increased computational complexity. Experiments on ShapeNet Part demonstrate that our model outperforms the state-of-the-art (SOTA) with lower complexity and it has strong prospects in applications requiring low power but high segmentation performance.

1 Introduction

3D point clouds are one of the most popular 3D representations as they can preserve the original geometric representation with minimal information loss and thus have applicability in various fields such as orthodontics[30], robotics[13] and self-driving [9].

Recently, there has been considerable success in applying deep learning in many areas such as computer vision and natural language processing, where it has outperformed traditional approaches, becoming a trendy area of research. In contrast, deep learning on point clouds faces several challenges due to their unstructured and irregular nature, precluding the use of grid convolutions directly onto raw point clouds. While previous research transformed point clouds into other grid forms such as 3D voxel [17] to overcome this, they are inefficient and can lead to a loss of geometric information due to quantisation effects.

PointNet [1], a recently proposed pioneering work, solves the problem by performing point convolution directly onto raw point clouds while achieving permutation invariance. Since then, many papers[14, 19, 25, 26] have proposed different point convolutions, but most of them still provide limited segmentation performance due to their crude learning architectures or local aggregation methods. These works consist of a single segmentation network that predicts labels for each point independently, and thus it tends to produce inconsistent labels such as misclassification in boundaries, noisy label within a class or misclassification of a class. Moreover, their convolution techniques are not designed to aggregate sufficient local features to learn complex structures, resulting in poor segmentation accuracy. These limit applicability in many practical tasks, such as orthodontics[30] and self-driving[9], that require high segmentation performance.

This paper presents a novel neural network approach for 3D point cloud segmentation to improve segmentation accuracy without extra computational burden. The proposed Adversarial Graph Convolution Network (AGCN) trains two networks, a segmentation network and a discriminator network, in an adversarial manner where the discriminator network calculates a difference between two respective embedding features of ground truth map and predicted label map from the segmentation network in its last convolution layer to train the segmentation network. This adversarial training helps improve the segmentation accuracy as well as the training stability of segmentation network by enabling the network to learn high level features of ground truth labels that are smooth and consistent. Additionally, we propose a new graph convolution, which is a geometry-preserving edge convolution, abbreviated as GeoEdgeConv. It is designed to aggregate rich local geometric features such as geometric shape or structure by explicitly incorporating both edge features and relative positions between a point and neighbouring points. This allows our network to reduce noisy labels as well as improve segmentation accuracy in boundaries by enabling the network to learn fine-detailed geometry of complex structures. The proposed network is evaluated on ShapeNet Part [29], and the results show that it outperforms the SOTA with lower complexity. Our contributions are:

- We propose a novel neural network approach for 3D point cloud segmentation by using an adversarial learning scheme underpinning our new AGCN, where unlike previous works, we present an embedding L_2 loss as an adversarial loss to provide more informative feedback to segmentation network.

- We introduce GeoEdgeConv as local feature aggregation method, which helps our network to efficiently and effectively learn complex local geometric structures. It enables a large receptive field by adopting dilated convolution and the space and time complexities are reduced by using additional group convolution.

- We propose a smaller AGCN, AGCN-S, that achieves the smallest time complexity and second lowest space complexity compared to the SOTA, but still outperforming them.

2 Related work

2.1 3D Point Cloud Segmentation

Traditionally, many researchers used hand-crafted features, especially local feature descriptors such as inner-distance [15] and geometry-based features [22, 23] for high-resolution

tasks such as segmentation. However, these methods are computationally expensive and are dependent on domain knowledge, being slow and difficult to process large point clouds.

Until [1] was introduced, many approaches transformed point clouds into grid forms such as images [24] or voxels [17] to apply grid convolutions due to irregular and unordered properties of points clouds. However, they are constrained by data resolution and can encounter problems such as quantisation effects, leading to a loss of information. The pioneering work [1] performs convolution on raw point clouds directly by using permutation invariance modules such as shared MLP and max-pooling. However, it lacked in capturing local features, so subsequent studies primarily focused on capturing local features using various methods such as KNN or ball query. In [14, 26], KNN was used to construct local graphs and calculated relative point features or positions respectively to extract local features. However, [26] dynamically updates graphs, increasing the computational cost, and [14] is not permutation invariant. Many studies [7, 11, 31] developed networks based on [26] by using channel attention, multiple receptive fields or residual connections, respectively, but the performances were sub-par to SOTA. In [19, 25], ball query was used, which set a fixed size of radius to create local graphs and select points within the radius. These studies somewhat improved performance, but they either have a large model size or lack fine-grained segmentation. To alleviate such problems, [9] focused on sampling strategy to increase inference speed and used various handcrafted features for rich local feature aggregation, while [6, 33] used transformers, which have a self-attention mechanism to place more weights on valuable features. However, [9] used a raw point coordinate vector as one of the handcrafted features, lacking in translation invariance, and [6, 33] have low inductive bias that makes the model to perform poorly on small size dataset.

2.2 Generative Adversarial Networks

Since [4], there have been many studies applying GANs to different applications. While many studies[10, 20, 34] focused on improving image generations, some studies[28, 32] proposed hybrid loss functions where a discriminator network is used to improve segmentation performance. Although there has been very little research that uses GANs for point cloud segmentation, a few studies such as [30] and [12] proposed adversarial training for point cloud segmentation. [30] constructed a discriminator inspired from T-Net [1] and fed statistical data calculated from each segmented tooth into the discriminator to reduce training time. However, their architectures are inspired from either [14] or [19], lacking in extracting fine local features. Additionally, as [3] stated, they used binary cross entropy (BCE) for adversarial training, which is not sufficient to train the network in stable and effective manners as it only provides a single binary prediction. Since then, there has not been any other work using different adversarial loss functions in 3D point cloud segmentation.

3 Methods

This section will first describe the general architecture of the proposed AGCN model, followed by details of training such as adversarial loss and a graph convolution.

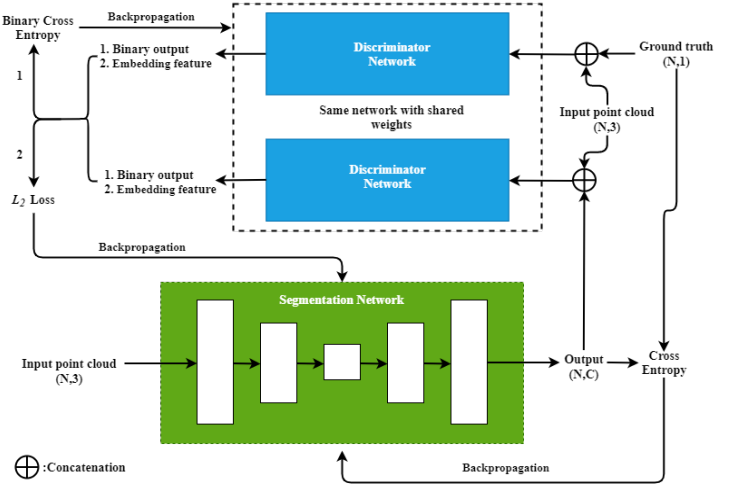

Figure 1: Overall architecture of proposed network

3.1 Architecture

As shown in Figure 1, input point clouds are first fed into the segmentation network, which outputs a predicted label map. The one-hot encoded label map is concatenated with the input point clouds as an input for the discriminator. Being trained with BCE loss, the discriminator network discriminates whether the label map is real or fake. The segmentation network is trained to minimise a hybrid loss, its own point-wise cross entropy loss and an additional adversarial loss that is defined as L_2 difference between two respective embedding feature vectors of the ground truth and the predicted label maps in the final convolution layer of the discriminator. In this way, the two networks are adversarial, and the segmentation network tries to deceive the discriminator network by predicting outputs that have the similar distribution as the ground truth. As shown in Figure 2, the segmentation network adopts the shape

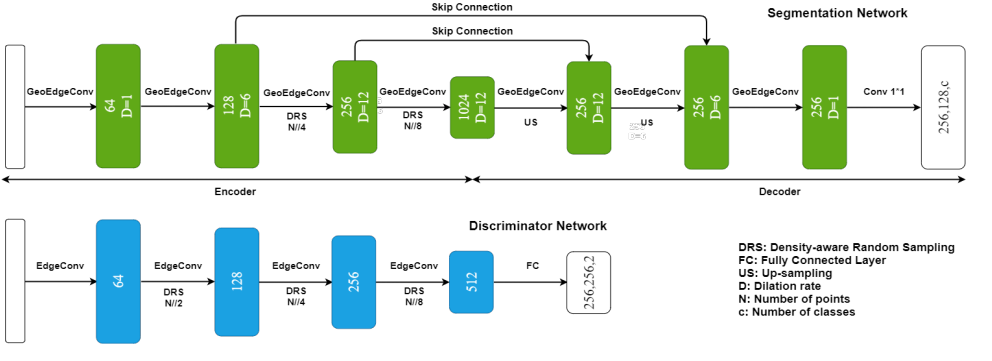

Figure 2: Proposed segmentation and discriminator networks

of U-Net [21]. In the encoder, point clouds are down-sampled using Density-aware Random Sampling (DRS), which is an extension of [5]. Random sampling (RS) can sub-sample

points quickly, but it may lose important geometric features over sparse region. In contrast, DRS estimates the density of each point by finding distances of neighbouring points and uses this distance to weigh each point during random sampling, which has a computational cost of $O(N)$. The weight of a point can be computed as follows:

$$\mathcal{W}_i = (\sum_{j=1}^{k} \|x_i - x_j\|_2)^n \tag{1}$$

where k is the number of neighbouring points in KNN, x_i and x_j are centre and neighbouring point respectively, and n decides how much to focus on the sparse regions to preserve important geometric information over sparse data. Furthermore, since we compute KNN in each convolution, there is no need to find the nearest neighbours again, reducing the computational cost significantly. The decoder consists of three components: an up-sampling by using Inverse Distance Weighted KNN Interpolation, a skip connection with the corresponding feature map from the encoder and GeoEdgeConv.

The discriminator network resembles the encoder part of the segmentation network. However, EdgeConv[26] is used instead of GeoEdgeConv as point coordinates of the ground truth and the prediction are the same; using GeoEdgeConv could degrade the discriminating ability by diluting informative local point features with unnecessary point coordinates.

3.2 Adversarial Loss

Adversarial training helps segmentation network to improve its performance by trying to reduce the difference between two outputs of the ground truth and predicted label maps in the discriminator. Typical adversarial learning-based segmentation networks are trained with two losses, a point-wise loss $\mathcal{L}_{point}$ and an adversarial loss $\mathcal{L}_{adv}$:

$$\mathcal{L}_S = \mathcal{L}_{point}(S(x;\theta_{seg}),y) + \lambda \mathcal{L}_{adv}(S(x;\theta_{seg});x,\theta_{disc}) \tag{2}$$

where x and y are an input and the corresponding label respectively, θ_{seg} and θ_{disc} are the parameters for the segmentation and the discriminator networks respectively, $S(x;\theta_{seg}))$ indicates the output from the segmentation, and λ represents the relative importance of the adversarial loss term. The discriminator network is trained so that the ground truth and predicted label inputs can be classified as true and fake respectively.

Existing works such as [12, 30] proposed adversarial learning for point cloud segmentation, but they both used BCE loss between the two outputs of the discriminator as an adversarial loss, resulting in unstable training as well as insufficient segmentation accuracy with gradient feedback by a single binary prediction based on global average feature of input labels. To overcome this drawback, we propose to use embedding features in the last convolution layer of the discriminator, which contain richer structural features than the final output of the input label map. Accordingly, our adversarial loss for the segmentation network is expressed as follows:

$$\mathcal{L}_{adv}(\hat{y},y;x,\theta_{disc}) = \frac{1}{B\times N\times P}\sum_{i=1}^{B}\sum_{j=1}^{N}\sum_{k=1}^{P}(D_{emb}^{ijk}(y;x,\theta_{disc}) - D_{emb}^{ijk}(\hat{y};x,\theta_{disc}))^2 \tag{3}$$

where B is batch size, N is the number of channels, P is the number of points, and $D_{emb}(\hat{y};x,\theta_{disc})$ illustrates an embedding from the discriminator network, given $\hat{y}$ and x as inputs. We extract

the embedding features D_{emb} from the final convolution layer of the discriminator and the discriminator is trained through minimizing the loss $\mathcal{L}_D$, which is expressed as follows:

$$\mathcal{L}_D = \mathcal{L}_{bce}(D(S(x;\theta_{seg});\theta_{disc}),0) + \mathcal{L}_{bce}(D(y;\theta_{disc}),1) \tag{4}$$

where the first and second terms denote BCE for the predicted label map from the segmentation network and the ground truth label map respectively.

As the segmentation network is trained so that the high level features of its predicted labels resemble those of the ground truth labels, it will deceive the discriminator. By training the two networks adversarially, the discriminator will also discriminate the plausible prediction from the segmentation network from the ground truth. Compared with [12, 30], which learn global average feature of the ground truth labels with BCE, our model improves the training stability of the networks and the accuracy of label prediction by learning high level features of the ground truth labels that are smooth and consistent.

Following the method from [7], the optimal value for λ can be estimated automatically by calculating homoscedastic uncertainty during back-propagation. Hence, the final loss for our segmentation network becomes:

$$\mathcal{L}_S = \frac{1}{\sigma_1^2}\mathcal{L}_{point} + \frac{1}{2\sigma_2^2}\mathcal{L}_{adv} + R(\sigma) \tag{5}$$

where each σ indicates the uncertainty of each task and $R(\sigma) = \log \sigma_1\sigma_2$ is a regularisation term to prevent the uncertainty parameters from becoming too large.

3.3 GeoEdgeConv

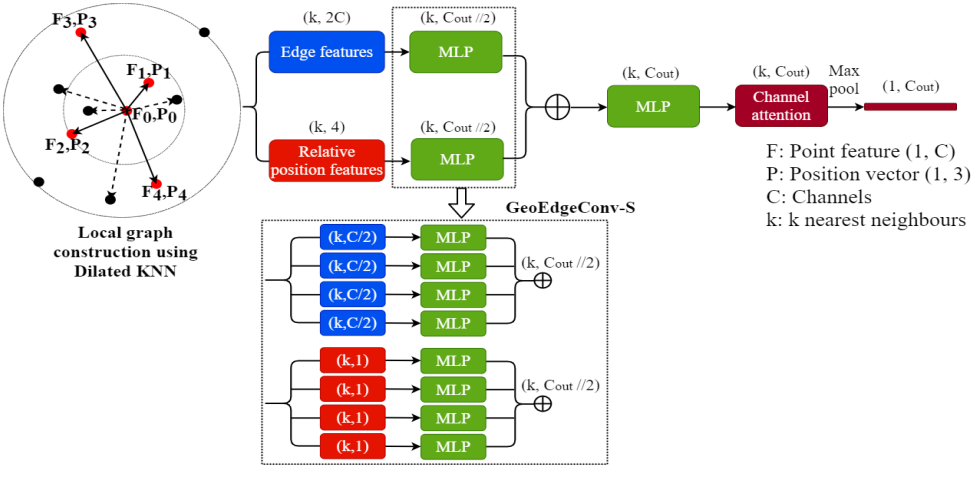

Figure 3: Proposed local feature extraction methods: GeoEdgeConv and GeoEdgeConv-S

To perform convolution onto raw point clouds, it is essential to satisfy permutation invariance and to extract local features. Previous works[1, 14, 19, 26] introduced various approaches to deal with these issues, but they still carry limitations in capturing rich local features.

Consider raw point clouds X with n points as $\{x_1...x_k...x_n\}$ where x_k has x,y,z coordinates. DGCNN[26] proposed EdgeConv as a method of local feature aggregation, which can be expressed as follows:

$$x_{new} = \max_{j:(i,j)\subseteq E} h_\theta(f_i, f_j - f_i) \tag{6}$$

where E denotes a local neighbourhood point set for convolution, h_θ is a nonlinear function with learnable parameters (e.g. ReLU with MLP) and $f_j - f_i$ is a relative point feature between a centre point i and a neighbouring point j.

However, EdgeConv is likely to lose local geometric information if points are down-sampled in intermediate layers, specially when a large receptive field is set. Eventually the drawback can result in poor segmentation accuracy such as misclassification in boundaries. This motivated us to propose GeoEdgeConv, which uses relative point coordinates explicitly in addition to edge features as point features. The GeoEdgeConv can be expressed as follows:

$$x_{new} = \max_{j:(i,j)\subseteq E} CA(\, h_{\theta 3}(h_{\theta 2}(f_i, f_j - f_i), h_{\theta 1}(x_j - x_i, \|x_j - x_i\|_2))) \tag{7}$$

where CA denotes channel-wise attention, $x_j - x_i$ and $\|x_j - x_i\|_2$ are relative positions and the Euclidean distance respectively, and $h_{\theta 1}$, $h_{\theta 2}$ and $h_{\theta 3}$ are mish[18] with shared MLP. By incorporating the two relative position features, the local geometry can be preserved over the convolution layers regardless of sub-sampling and the size of receptive field. This advantage improves segmentation accuracy significantly by enabling the network to learn fine-detailed geometry of complex structures.

GeoEdgeConv first constructs local graphs using dilated KNN as shown in Fig 3, enlarging the receptive field with no added computational cost. Then, each local point and relative position features are fed into shared MLPs, $h_{\theta 1}$ and $h_{\theta 2}$, to lift the features into the same number of channels to carry same importance. The outputs are then concatenated and fed into another MLP to fuse information. A channel-wise attention by squeeze-and-excitation[8] is added to perform feature recalibration to suppresses unnecessary features and place more weights on important features, and thus improve the segmentation accuracy.

To design a lighter model, we propose GeoEdgeConv-S, where it uses group convolution for the shared MLPs, $h_{\theta 1}$ and $h_{\theta 2}$, with group size$=4$ to reduce the model size and complexity as shown in figure 3. The size of a kernel is also halved to $k = 8$ from 16 to reduce the computational cost, while the dilation rate is increased to maintain the same receptive field. The full procedure of GeoEdgeConv is summarised in Algorithm 1.

Algorithm 1 GeoEdgeConv

Input: x_{pos}, x, k, r ▷ position vectors, point features, size of neighbours, dilation rate
Output: F_{out}
F_c, F_n, P_c, P_n ▷ Centre and neighbouring point features and point coordinates

1: $idx \leftarrow DilatedKNN(x_{pos}, k, r)$
2: $F, P \leftarrow x[idx], x_{pos}[idx]$ ▷ Construct local neighbourhood graphs
3: $F_{edge}, P_{geo} \leftarrow [F_n - F_c, F_c], [P_n - P_c, \|P_n - P_c\|_2]$ ▷ Find edge and relative position features
4: $F_{edge}, F_{geo} \leftarrow MLP(F_{edge}), MLP(P_{geo})$ ▷ Lift each feature into same dimension
5: $F_{out} \leftarrow CA(MLP([F_{edge}, F_{geo}]))$ ▷ Fuse the features and feed to the channel attention
6: **return** F_{out}

We summarise the uniqueness of our method in comparison with existing methods in table 1. Among our proposed modules, adversarial loss, features including channel attention, Dilated KNN and DRS are the modules incorporated to improve the accuracy. Also,

Group MLP and Dilated KNN reduce space and time complexities and DRS decreases time complexity.

Model	Adversarial	Feature	Learnable parameter	Graph	Sampling
PointNet[1]	x	f_i	MLP, ReLU	x	x
PPSAN[13]	BCE	$f_j, x_i - x_j$	MLP, ReLU	Ball query	FPS
DGCNN[28]	x	$f_i, f_j - f_i$	MLP, LReLU	KNN	x
RandLA-Net[9]	x	$f_i, x_i, x_j, x_i - x_j, \|x_i - x_j\|_2$	MLP, LReLU, SA	KNN	RS
AGCN	Embedding L_2	$f_i, f_j - f_i, x_j - x_i, \|x_j - x_i\|_2$	Group MLP, Mish, CA	DKNN	DRS

Table 1: Comparison with existing methods where LReLU denotes Leaky ReLU, SA is spatial attention and DKNN is dilated KNN

4 Experiments and Results

Our model was trained on NVIDIA Tesla T4 by Adam optimizer with an initial learning rate of 0.005 and a cosine annealing schedule[16] for 300 epochs. We also used group normalisation [27] with batch size$= 2$.

4.1 Dataset

We evaluated AGCN using ShapeNet Part[29], which has 16881 shapes of 16 categories, where each category has a different number of parts ranging from 2 to 6. The dataset has raw point clouds in 3-dimensional vectors with labels for each point. For a fair comparison, we used the same training/test splits as [14, 26], and for each epoch, we augmented the data by sampling 2048 points randomly using DRS.

4.2 Evaluation on ShapeNet Part

The proposed model was evaluated against the models introduced in Chapter 2 including the current SOTA [25]. Instance average IoU and class average IoU are used as metrics, where the former is calculated by the average IoU of every samples and the latter is calculated by averaging the mean IoU of each object. They are measured in mIoU, which is defined as mean IoU of each sample.

Table 2 shows that both AGCN and AGCN-S outperform the current SOTA [6, 25] in both metrics of instance average IoU and class average IoU by large margins with much lower space complexity. Especially, AGCN performed worse on only 3 out of 16 objects, *Cap*, *Lamp* and *Skateboard* and achieved exceptional performances in *Aeroplane*, *Bag*, *Car*, *Earphones* and *Mug*, outperforming the SOTA by 1.3 - 2.6 IoU. However, our models performed relatively poorly in *Skateboard*, ranking 2*nd* and 4*th* in the table. Since the number of points for wheels is very small compared to board, it gives a greater penalty for IoU, if the predictions made were wrong for wheels, which may have caused the our models to perform badly.

Although AGCN achieves exceptional performances with very small space complexity, it has relatively high time complexity. This may limit AGCN in a situation where fast processing time is required in low powered devices. To alleviate this, we introduced AGCN-S employing GeoEdgeConv-S in Chapter 3. Table 2 shows that our AGCN-S model has the smallest MAC and the second smallest model size, where the model size is only marginally

Shape/Model	POINTNET [8]	DGCNN [22]	PointCNN [12]	KPConv [20]	PPSAN [11]	PCT [6]	**AGCN-S**	**AGCN**
No. of Parameters	7.9M	**1.5M**	8.5M	15.0M	1.8M	2.9M	1.7M	2.3M
Model size	35.8MB	**6.2MB**	32.7MB	56.5MB	22MB	-	6.7MB	8.9MB
MAC	4.8G	4.9G	4.5G	-	4.9G	-	**4.4G**	20.0G
Instance avg. IoU	83.7	85.2	86.1	86.4	85.2	86.4	**87.0**	**87.9**
Class avg. IoU	80.4	82.3	84.5	85.1	82.6	83.1	**85.6**	**86.7**
AERO	83.4	84.0	84.1	84.6	82.9	85.0	**86.4**	**87.6**
BAG	78.7	83.4	84.4	86.3	82.7	82.4	**90.5**	**92.3**
CAP	82.5	86.7	86.0	87.2	86.4	**89.0**	86.6	87.3
CAR	74.9	77.8	80.8	81.1	78.9	81.2	81.1	**82.4**
CHAIR	89.6	90.6	90.6	91.1	90.6	91.9	91.6	**92.8**
EARPHONES	73.0	74.7	79.7	76.5	76.5	71.5	**82.3**	**82.3**
GUITAR	91.5	91.2	92.3	92.6	91.0	91.3	**93.1**	**93.4**
KNIFE	85.9	87.5	88.4	88.4	85.7	88.1	**88.5**	**89.1**
LAMP	80.8	82.8	85.3	82.7	84.3	**86.3**	85.4	86.2
LAPTOP	95.3	95.7	96.1	96.2	96.1	95.8	**96.5**	**96.5**
MOTOR	65.2	66.3	77.2	78.1	74.4	64.6	76.4	**80.4**
MUG	93.0	94.9	95.3	95.8	95.1	95.8	**97.3**	**97.9**
PISTOL	81.2	81.1	84.2	85.4	81.8	83.6	**86.8**	**86.2**
ROCKET	57.9	63.5	64.2	**69.0**	58.2	62.2	67.6	**69.0**
SKATE BOARD	72.8	74.5	80.0	**82.0**	75.5	77.6	76.8	80.0
TABLE	80.6	82.6	83.0	83.6	82.2	83.7	83.5	**84.4**

Table 2: Comparison results on ShapeNet Part against the SOTA

larger than that of the smallest model. These results indicate that our models are ideal for applications that require high segmentation accuracy in low-powered devices.

4.3 Model Analysis

Table 3 evaluates the effects of modules proposed in this paper. The evaluation shows that

Version	Adversarial	Convolution	Instance Avg. IoU	Class Avg. IoU
Baseline	x	EdgeConv	85.2	83.0
AGCN (No adv.)	x	GeoEdgeConv	86.3	84.4
AGCN (Full)	Embedding L_2	GeoEdgeConv	**87.9**	**86.7**

Table 3: Effects of the modules on performance

applying GeoEdgeConv to the baseline model with EdgeConv resulted in 1.1 mIoU and 1.4 mIoU gains in the instance average IoU and the class average IoU respectively. The results exhibit that the relative position features incorporated into GeoEdgeConv played an important role in improving the accuracy. The additional application of the adversarial learning produced 1.6 mIoU and 2.3 mIoU gains from the model with GeoEdgeConv in the instance average IoU and the class average IoU respectively. They also prove that the proposed adversarial learning affected the accuracy significantly. We can also see that total IoU gain in the class average are bigger by 0.7 mIoU than that in the instance average. This suggests that our model is especially effective when there is only small training data available.

Figure 4 shows that the baseline has many incorrect labels, including misclassification on boundaries and within a class, while AGCN has less errors in general with only few boundary errors. This demonstrates the effects of using our proposed modules, GeoEdgeConv and the adversarial loss, where GeoEdgeConv helps reduce boundary errors as well as noisy labels by providing fine-detailed information of geometric structures and the adversarial training helps the model to predict more smooth and consistent labels, reducing both boundary and noise errors. Finally, we performed hypothesis testing on AGCN and the baseline; the p-value was $1.81e^{-}11$, confirming that the improvements are statistically significant.

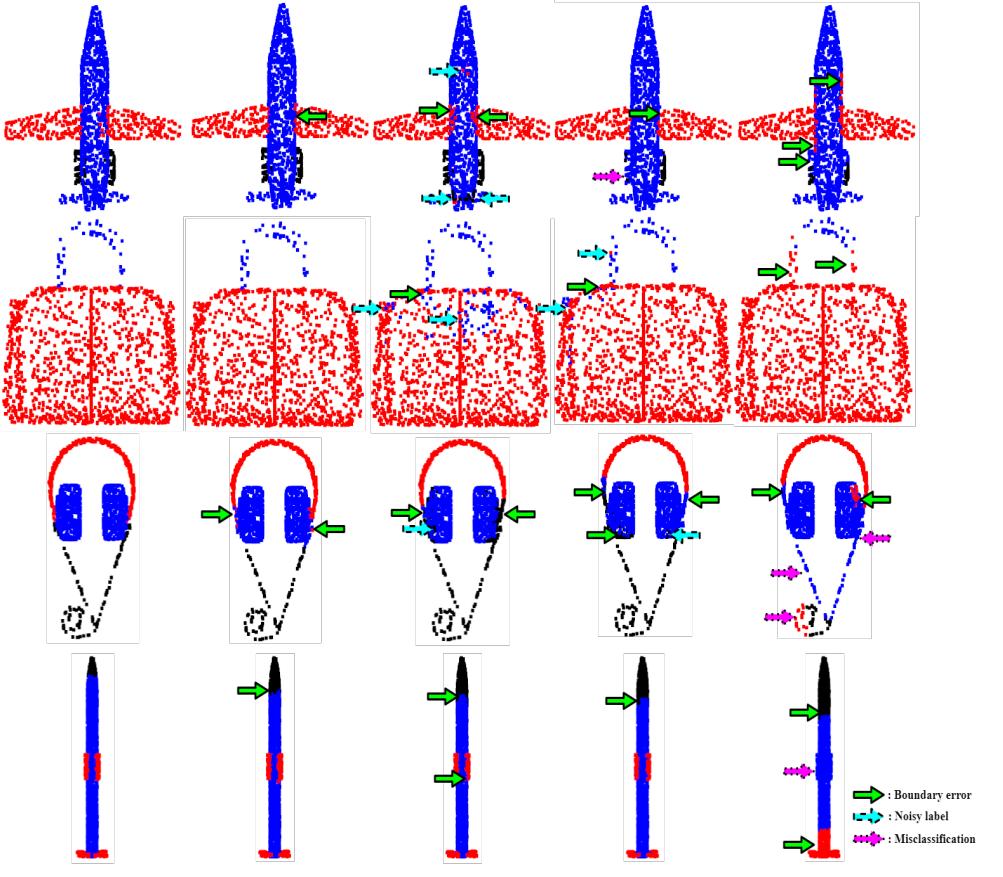

Figure 4: Visualisation of segmentation results of different models (From left: Ground truth, AGCN, AGCN (Baseline), DGCNN, PointNet. From top: Aeroplane, Bag, Earphones, Rocket)

5 Conclusion

In this paper, we presented a novel neural network approach for point cloud segmentation. The method of training the GCN in the adversarial learning scheme using an embedding L_2 loss tries to reduce noisy labels by learning high level features of the ground truth labels. The proposed GeoEdgeConv preserves geometric features over convolution layers by explicitly using relative position features in addition to edge features, and also enlarges receptive field without increase in computational cost by incorporating dilated KNN. Being benefited from the proposed approaches, AGCN outperforms the SOTA significantly. With the aid of efficient modules such as a small kernel size, DRS and group convolution, AGCN-S becomes more efficient and light, achieving lower space and time complexities, and therefore can be extended to various applications, particularly for low-powered devices requiring high segmentation performance.

6 Acknowledgments

The authors gratefully acknowledge helpful input and guidance from Yukun Zhou in preparing this manuscript. The UK EPSRC grant numbers M020533, R006032, R014019, V034537, Wellcome Trust UNS113739, and the NIHR UCLH Biomedical Research Centre support DCA's work on this topic.

References

[1] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[2] Roberto Cipolla, Yarin Gal, and Alex Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[3] M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, and Michael Hofmann. El-gan: Embedding loss driven generative adversarial networks for lane detection. *ArXiv*, abs/1806.05525, 2018.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, volume 2, pages 2672–2680, 2014.

[5] Fabian Groh, Patrick Wieschollek, and Hendrik P. A. Lensch. Flex-convolution (million-scale point-cloud learning beyond grid-worlds). In *Asian Conference on Computer Vision (ACCV)*, December 2018.

[6] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Apr 2021.

[7] Jinseok Hong, Keeyoung Kim, and Hongchul Lee. Faster dynamic graph cnn: Faster deep learning on 3d point cloud data. *IEEE Access*, 8:190529–190538, 2020.

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[9] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[11] Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv*, 2020.

[12] Hongyan Li, Zhengxing Sun, Yunjie Wu, and Bo Li. Ppsan: Perceptual-aware 3d point cloud segmentation via adversarial learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4025–4029, 2019.

[13] Xuyou Li, Shitong Du, Guangchun Li, and Haoyu Li. Integrate point-cloud segmentation with 3d lidar scan-matching for mobile robot localization and mapping. *Sensors*, 20(1), 2020.

[14] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, X. Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, 2018.

[15] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299, 2007.

[16] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016.

[17] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[18] Diganta Misra. Mish: A self regularized non-monotonic activation function. In *BMVC*, 2020.

[19] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5105–5114, 2017.

[20] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015.

[22] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Michael Beetz, Intelligent Autonomous Systems, and Technische Universität München. Persistent point feature histograms for 3d point clouds. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10)*, 2008.

[23] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.

[24] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, 2015.

[25] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6410–6419, 2019.

[26] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[27] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[28] Yuan Xue, Tao Xu, Han Zhang, L. Rodney Long, and Xiaolei Huang. Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. *Neuroinformatics*, 16:383–392, 2018.

[29] 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. In *SIGGRAPH Asia*, 2016.

[30] Farhad Ghazvinian Zanjani, David Anssari Moin, Bas Verheij, Frank Claessen, Teo Cherici, Tao Tan, and Peter H. N. de With. Deep learning approach to semantic segmentation in 3d point cloud intra-oral scans of teeth. In *International Conference on Medical Imaging with Deep Learning*, 2019.

[31] Zhengli Zhai, Xin Zhang, and Luyao Yao. Multi-scale dynamic graph convolution network for point clouds classification. *IEEE Access*, 8:65591–65598, 2020.

[32] Xinming Zhang, Xiaobin Zhu, 3rd Xiao–Yu Zhang, Naiguang Zhang, Peng Li, and Lei Wang. Seggan: Semantic segmentation with generative adversarial network. In *IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pages 1–5, 2018.

[33] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021.

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.