Improving 3D Point Cloud Classification with Self-Attentive Neural Networks

1st Anderson Antonio do Nascimento da Silva
HP Inc.
Av. Ipiranga, 6681, Porto Alegre-RS, Brazil
anderson.silva@hp.com

2nd Jônatas Wehrmann
Pontifícia Universidade Católica do RS
School of Technology, PUCRS
Av. Ipiranga, 6681, Porto Alegre-RS, Brazil
jonatas.wehrmann@acad.pucrs.br

***Abstract*—Point cloud is an important geometric representation that can be used to represent 3D data. It is very close to the raw output of laser sensors, often used to scan volumetric data into digital form. Unlike images which are represented in regular grids of pixels, point clouds are irregular and do not present explicit order making it difficult to use convolutional layers, given that such layers excel in learning from local patterns. Inspired by recent results of applying Self-attention mechanism on deep neural networks, this paper proposes the use of such technique to improve representations learned from 3D geometrics data. We provide experiments by adding Self-attention layers in state-of-the-art models, such as PointNet and DGCNN trained for the task of classification in the widely used ModelNet40 dataset. Our experiments show that self-attention improves the accuracy, while making the algorithm more stable in terms of loss and accuracy values during training.**

***Index Terms*—Deep Learning, 3D geometric, Point Clouds, Self-Attention, classification**

I. Introduction

Point clouds are a collection of points that represents a 3D shape, where each point (x, y, z) is often represented by the x, y and z coordinates. Although, note that such a representation strategy allows also adding more attributes, such as RGB color information. Point cloud is *per se* sparse, irregular and unordered, thus making it difficult to apply traditional convolution neural network (CNN) in it. Early attempts in learning point clouds typically transformed the data into regular 3D *voxel* grids, or into a collection of images before feeding them to a deep net architecture. Recent advances in point cloud-based neural networks [1]–[5] have allowed neural networks to process 3D data directly without requiring any preprocessing involving such expensive data transformations.

There has been a recent trend that regards the use of Attention-based architectures for both Computer Vision and Natural Language Processing models. In fact, it was the core component to provide large performance improvements for language modeling and Neural Machine Translation for instance. Different from the convolution operator that processes local information, attention layers always process global-level information. Thus, we believe that attention-like modules could improve results for classification of point cloud data.

This paper takes advantage of such advances to further improve the results using attention mechanism in the task of classifying 3D objects. More specifically, we add self-attention layers in well known state-of-the-art models, namely PointNet [1] and Dynamic Graph CNN [3]. Our experimental analysis shows that the proposed approaches perform better than non-attentive models.

II. Related work

Research regarding geometric data processing presented a similar path when compared to the processing of regular 2D images. Early attempts were based on hand-crafted features, then following the advances on deep learning the scientific community explored training deep neural networks on 3D data. Thus, the networks themselves demonstrated to able to learn proper feature representations and handle several tasks.

A. Hand-crafted features

Tasks in geometric data processing and analysis require some notion of similarity between shapes. Traditionally, this similarity is established by constructing hand-crafted feature descriptors taking advantage of statistical properties of the 3D data. Many papers in Computer Vision and graphs propose local feature descriptors handcrafted towards specific tasks [6], [7]. Feature descriptors are often categorized as local and global features which makes non-trivial the task of finding optimal feature combination.

B. Deep learning features

Early attempts to work on 3D data were developed by adapting ideas from traditional Deep Learning models designed to work 2D images. For instance, by taking multiple 2D images from 3D models and concatenate them to process and extract features [8], or applying convolution on 3D volumetric grids [9].

PointNet [1] is a pioneer to apply deep learning on raw point clouds without the need to transform into regular formats. Most researchers typically transform point clouds to regular formats like 3D Voxels grids before feeding them to deep nets. PointNet takes n points as input, applies input and feature transformations, then aggregates point features by max pooling, the feature vector output can then be used for classification or segmentation task. The drawback of PointNet is that it does not capture local structures induced by the metric

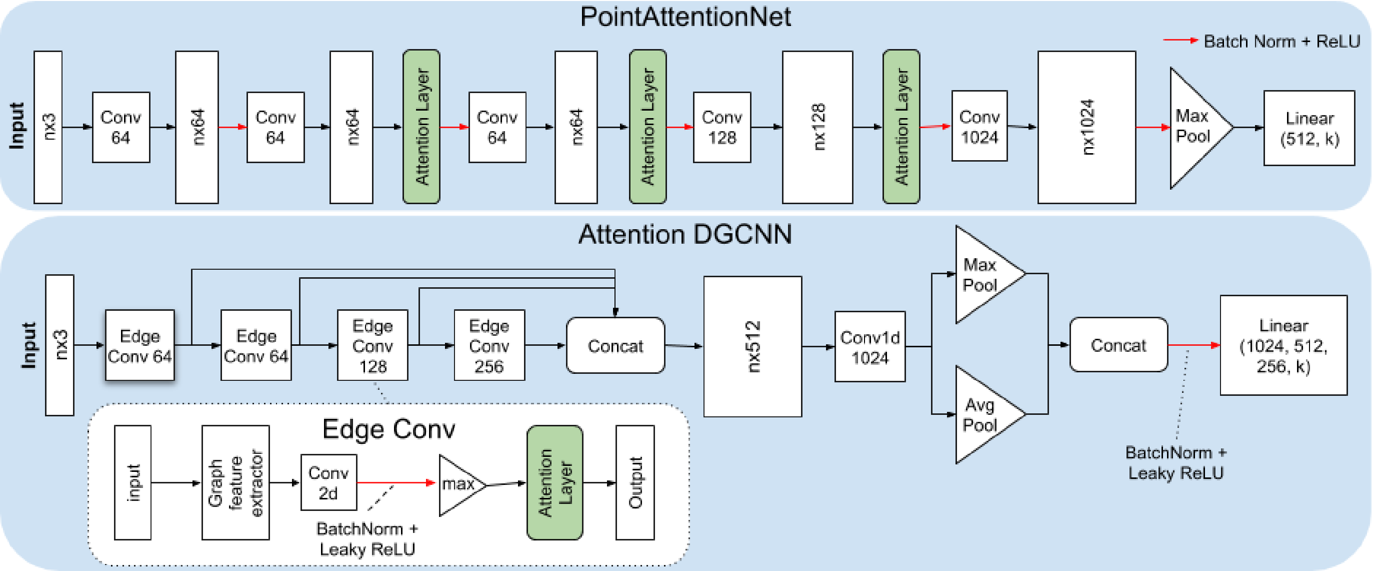

Fig. 1. Model architecture

space in which points live in, limiting its ability to recognize fine-grained patterns and generalization for complex scenes.

PointNet++ [2] introduces a hierarchical neural network that is able to learn local features. The neural network is composed by three layers, the *sampling layer* selects a set of points from the input points, the *grouping layer* constructs local region sets by finding neighboring points around the *centroids* and then *PointNet layer* that uses PointNet to encode local region patterns into feature vectors. The feature vectors can also be used as input for classification or segmentation tasks.

Dynamic Graph CNN [3] proposes an approach inspired by PointNet and graph convolution operations. DGCNN exploits local geometric structures by constructing a local neighborhood graph and applying convolution-like operations named *edge-convolution* in each layer of the network. An important aspect that differs DGCNN from graph CNNs, is that the graph in DGCNN is dynamically updated after each layer.

Extracting features from point cloud has became an active research field with many methods being proposed, such as Linked Dynamic Graph CNN [4], PointConv [5], PointCNN [10], SO-Net [11], and SeqViews2SeqLabels [12]

III. Model architecture

Attention mechanism has proven to improve the performance of tasks such as language modeling and machine translation. We propose and evaluate the use of attention mechanisms in 3D Deep Learning by applying Attention in two state-of-the-art models PointNet and DGCNN trained specifically for classification. We select two models as they present distinct characteristics, so as to improve the robustness of our experiments. For instance, DGCNN has somewhat sophisticated layers which output new graphs to represent the features. Hence, the order of the graph changes, which could affect how attention layers process the inputs. On the other hand, PointNet is a simpler network with few layers, making the input order unchanged throughout the entire forward pass, which could provide an easier scenario for the attention layer to focus on specific input parts.

The intuition behind attention is to enable a neural network to focus on the more relevant part of the input in order to improve its performance on the task at hand. Attention layers employ three input matrices, namely *query*, *key* and *values*. They are designed to use *key* and *query* matrices to compute a compatibility function, whose result is used to filter input *values*. There are many variants of Attention modules, and among them our work is based on the mechanisms *Self-attention* and *Multi-head attention* presented by Vaswani [13]. Self-attention is a module that computes query, key and value matrices based on the same input. On the other side, *Multi-head attention* consists of h parallel Self-attention layers, which one called as *head*. Each head receives the same query, key and value inputs and the outputs of h attention heads are concatenated.

In Figure-1 we represent the proposed models, namely PointAttentionNet and AttentionDGCNN. The first one, is a neural network based on the PointNet, in which we introduced three attention layers. More specifically, each introduced layer is a multi-head self-attention with residual connection. As in original PointNet all layers (convolution and convolution/attention) use batch normalization and ReLU as primary non-linearity. At the end of the network we employ an Adaptive Max-pooling layer followed by a linear layer with batch normalization, ReLU and dropout. The final classification layer takes the generated features, and through a linear projection provides class scores. In addition to the attention mechanism, the proposed architecture removes the transformation layer proposed in the original model to better analyze the impact of attention in this type of data.

The second neural network is based on DGCNN, where we introduced the attention layer as part of the EdgeConv[1] layers. The attention mechanism is a multi-head self-attention with residual connection which is applied at the end of the EdgeConv proposed by DGCNN. The model takes as input *n* points, applies 4 layers of EdgeConv of different sizes and aggregates the features of all EdgeConv layers. The features are then aggregated globally to a form of 1D global descriptor then applying global max pooling and average pooling which are aggregated before linear layers to compute the score of the classes.

IV. Experiments

This section describes the experimental setup, that comprises a brief review of the dataset used, along with the main hyperparameters used for training our models.

A. Training Details

We trained the models using the ModelNet40 [14] dataset. ModelNet40 contains 12311 meshed CAD 3D models in 40 different classes. The dataset is available with predefined sets of training and test splits. Following important Machine Learning good practices, we decided to update splits so as to use a validation set as well. Hence, it was divided in training, validation, and test sets. The training set corresponds to 70% of the dataset while the remaining is equally divided in validation and test sets. We sample 1024 points from each of the meshed surfaces, and after loading we randomly sample the order of the points as a data augmentation strategy.

There is a large variety of *hyperparameters* to train such models. We use results from the validation set to empirically define those settings. More specifically, we use dropout of 0.5 in the linear layers and we evaluate training with both Adam and SGD optimizers. The learning rates for Adam are 10^{-5}, 10^{-4}, 10^{-3}, and 10^{-2}, while for SGD the learning rates are 2×10^{-1}, 10^{-1}, 5×10^{-2}, 25×10^{-3}, and 125×10^{-4}.

In addition, models trained with SGD use 0.9 for momentum. Self-attention-based networks employ either 4 or 8 attention heads. PointNet-based models were trained for 300 epochs, while DGCNN-based ones were trained during 150 epochs. The loss function used in all models is cross-entropy with label smoothing.

V. Results

We observed that results from the proposed models present better overall results considering their non-attentive counterparts. In overall, results in terms of accuracy improve 1% when using self-attention models allied to those models. We do believe that one could achieve larger result difference when trained in larger datasets, which seem to play an important role for deep self-attentive models, such as Transformers. Moreover, ModelNet40 does not comprise overly complex 3D models, and even simple models are able to perform well. That being said, it is very likely that the proposed models would present larger performance difference when trained on more complex data. Though, given the scope of this work, and hardware limitations, we let such experiments for future work.

Table-I compares our best results based on *balanced accuracy*[2] comparing with state-of-the-art deep learning models.

TABLE I
ModelNet40 Classification Accuracy

Method	Accuracy (%)
PointNet++ [2]	90.2
PointCNN [10]	92.2
PointConv [5]	92.5
Linked Dynamic Graph CNN [4]	92.9
SeqViews2SeqLabels [12]	93.4
SO-Net [11]	93.4
PointNet	83.6
PointAttentionNet	**84.5**
DGCNN	94.2
AttentionDGCNN	**95.7**

Results have demonstrated that Attention models showed to be more robust than the original ones. Figures-2 and Figure-3 show plot the validation accuracy and loss over the training epochs. We noticed that the proposed attention models seem to be more stable and less dependent on the learning rate. Moreover, different learning rates generate similar results in accuracy terms, and loss values also present less fluctuation over the training compared to original implementations without Self-Attention.

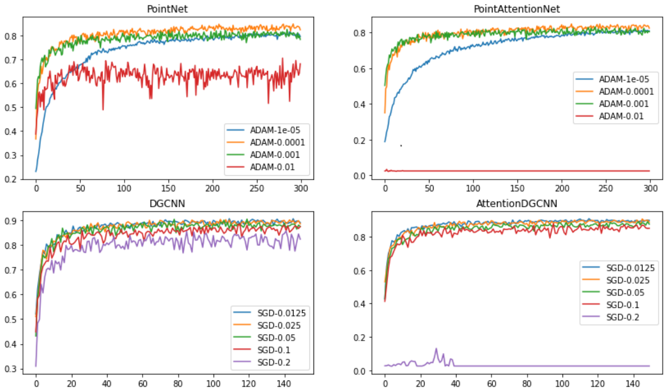

Fig. 2. Accuracy over the training in original and proposed models

The major drawback of attention-based models is the computational cost, which roughly doubles with 4 and 8 heads when compared with models without attention. Nonetheless, the number of epochs required to achieve satisfactory results did not decrease by using attention.

Figure-4 showcases qualitative results depicting classification predictions for randomly selected 3D point cloud inputs. The green plots represent correct classification and red plots represent wrong. In this example, all models were unable

[1] The EdgeConv, proposed by Wang [3], is able to extract local domain information by constructing a neighborhood graph and applying convolution-like operation on edges connecting pairs of points

[2] Balanced accuracy is the average of recall obtained on each class. This metric is used when the dataset is unbalanced, that is a dataset composed of classes with different number samples in training, test, and validation sets

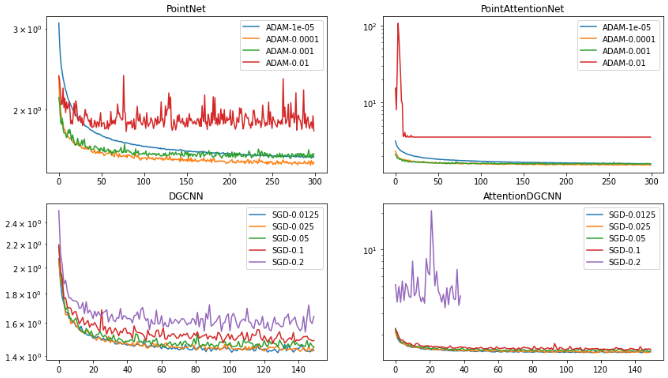

Fig. 3. Loss over the training in original and modified models

to classify the *curtain* object and all models classified it differently.

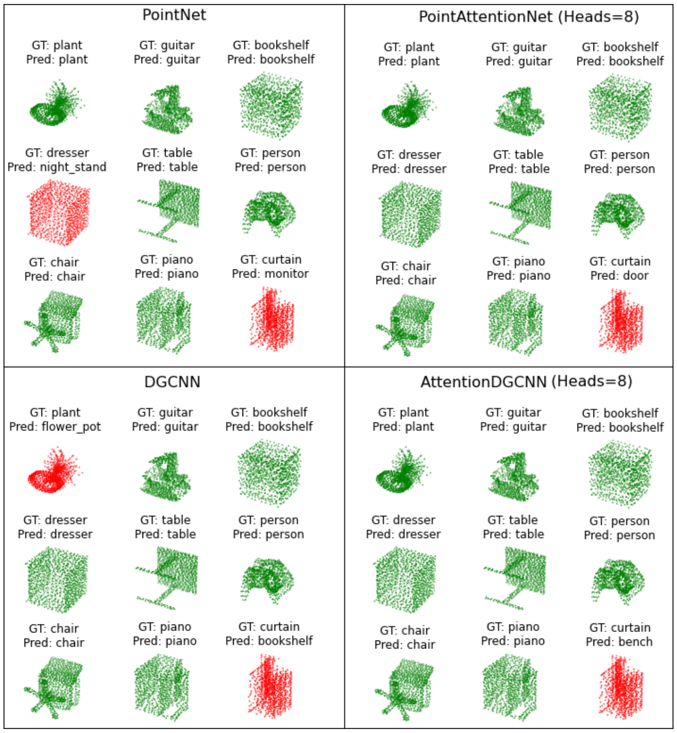

Fig. 4. Classification results

VI. Conclusion and Future Work

In this paper, we explore the use of Attention mechanisms in state-of-the-art models for 3D point cloud classification, namely PointNet and DGCNN. We compare results of the two algorithms with and without the attention layers and also compared the results with additional state-of-the-art approaches. We observed that both proposed models equipped with attention layers achieved better results than their non-attentive counterparts. In addition, they proved to be more robust in terms of accuracy and presented more stable loss values during training. The main drawback of the presented approaches is that they are heavier than the original models, requiring additional time for training.

For future work, we intend to evaluate the proposed models in larger and more complex datasets, as well as in real-world objects dataset [15]. It is also interesting to provide visualizations towards the understanding of the networks and evaluate its behavior for the segmentation task.

References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2016, cite arxiv:1612.00593. [Online]. Available: http://arxiv.org/abs/1612.00593

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5099–5108. [Online]. Available: http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, Oct. 2019. [Online]. Available: https://doi.org/10.1145/3326362

[4] 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," *CoRR*, vol. abs/1904.10014, 2019. [Online]. Available: http://arxiv.org/abs/1904.10014

[5] W. Wu, Z. Qi, and F. Li, "Pointconv: Deep convolutional networks on 3d point clouds," *CoRR*, vol. abs/1811.07246, 2018. [Online]. Available: http://arxiv.org/abs/1811.07246

[6] S. Biasotti, A. Cerri, A. Bronstein, and M. Bronstein, "Recent trends, applications, and perspectives in 3d shape similarity assessment," *Computer Graphics Forum*, vol. 35, no. 6, pp. 87–119, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12734

[7] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1681–1707, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.01884.x

[8] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proc. ICCV*, 2015.

[9] 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)*, 2015, pp. 922–928.

[10] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 820–830. [Online]. Available: http://papers.nips.cc/paper/7362-pointcnn-convolution-on-x-transformed-points.pdf

[11] J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis," *arXiv preprint arXiv:1803.04249*, 2018.

[12] H. Zhizhong, S. Mingyang, L. Zhenbao, V. Chi-Man, L. Yu-Shen, H. Junwei, M. Zwicker, and C. P. Chen, "Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention," *IEEE Transactions on Image Processing*, 2019, 28(2): 658-672.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[14] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," 2014.

[15] M. A. Uy, Q.-H. Pham, B.-S. Hua, D. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1588–1597, 2019.