Adaptive Pyramid Context Fusion for Point Cloud Perception

Haojia Lin, Zhipeng Luo, *Member, IEEE*, Wen Li, *Student Member, IEEE*,
Yiping Chen, *Senior Member, IEEE*, Cheng Wang, *Senior Member, IEEE*,
and Jonathan Li, *Senior Member, IEEE*

***Abstract*—Deep learning for 3-D point cloud perception has been a very active research topic in recent years. A current trend is toward the combination of the semantically strong and the fine-grained information from different scales of intermediate representations to boost network generalization power and robustness against scale variation. One prominent challenge is how to effectively conduct the allocation of multiple scales of information. In this letter, we propose a module, named adaptive pyramid context fusion (APCF), to adaptively capture scales of contextual information from a multiscale feature pyramid for the point cloud. The APCF module reweights and aggregates the features from different levels in the feature pyramid via a softmax attention strategy. The allocation of information is adaptively conducted level by level from bottom to up first and then from top to bottom. To ensure both effectiveness and efficiency, we propose a multiscale context-aware network APCF-Net through applying our proposed APCF to the PointConv architecture. Experiments demonstrate that APCF-Net surpasses its vanilla counterpart by a large margin both in effectiveness and efficiency. Especially, APCF-Net outperforms state-of-the-art approaches on 3-D object classification and semantic segmentation task, with the overall accuracy of 93.3% on ModelNet40 and mIoU of 63.1% on ScanNet V2 online test.**

***Index Terms*—Deep learning, feature pyramid, point cloud perception.**

I. Introduction

RECENTLY, point cloud perception has become an active research topic in 3-D computer vision, especially for robotics, augmented reality, and autonomous driving. Due to the revival of deep learning, a lot of neural network models have been undertaken for 3-D point cloud processing. Because of the irregular nature of point cloud, it is hard to directly apply a convolutional neural network (CNN) on such data. A straightforward way to address this issue is to convert point clouds into images [1]–[4] or voxels [5]–[7] before utilizing CNNs. However, such representation conversion will unavoidably lead to memory inefficiency and a loss of geometry information.

As a pioneering method of processing point cloud directly, PointNet [8] learns per-point representation by applying a shared MLP on each point individually. This work has inspired many methods that develop local aggregation operators to encode the context in a local region into the pointwise feature [9]–[16]. By alternating farthest point sampling (FPS) with features grouping, PointNet++ [9] extends PointNet to a hierarchical architecture that captures the local dependence layer by layer. In addition, several works borrow the idea of graph CNN [17]. DGCNN [10] constructs a KNN graph for each point and proposes an EdgeConv to exploit the local dependence in the dynamic feature space. PointASNL [11] uses the nonlocal [18] operation locally to ease the biased effect of the outliers in the subsampled points. Besides, many works develop convolution operations available for a point set. By learning an X-transformation from the input points, PointCNN [12] transforms the points into a latent and potentially canonical space and then applies standard convolution to these transformed points. PointConv [13] proposes a novel formulation for learning continuous filters to perform convolution efficiently. PatchCNN [15] proposes a PointPatch module to explicitly model geometric relationships among points. PosPool [16] proposes a deep residual network architecture and a simple local aggregation operator without learnable weights, which is able to perform similarly or slightly better than existing sophisticated operators. In these networks, local feature extraction units are repeatedly applied during the feature learning process. Due to the local nature of the aggregation operators, however, the features learned by these networks usually have limited receptive fields, which is difficult to keep consistent with the scale variation of the objects across the point clouds.

To address this problem, we propose a novel module adaptive pyramid context fusion (APCF) and an improved network APCF-Net to adaptively generate representations with multiscale contextual information. Compared with the existing methods, the representation generated by APCF has an adaptive context, which enables it to meet the challenge of scale variation across point clouds. Our contributions are summarized as follows.

1) We propose an APCF module that conducts a bidirection information communication and a dynamic selection mechanism across levels in the feature pyramid to

Manuscript received September 5, 2020; revised October 20, 2020; accepted November 5, 2020. Date of publication November 19, 2020; date of current version December 29, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 41871380 and Grant U1605254 and in part by the Natural Sciences and Engineering Research to Council of Canada under Grant 50503-10284. *(Corresponding authors: Yiping Chen; Jonathan Li.)*

Haojia Lin, Zhipeng Luo, Wen Li, Yiping Chen, and Cheng Wang are with the Fujian Key Lab of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: linhaojia@stu.xmu.ed-u.cn; zpluo@stu.xmu.edu.cn; liwen777@stu.xmu.edu.cn; chenyiping@xmu.edu.cn; cwang@xmu.edu.cn).

Jonathan Li is with the Fujian Key Lab of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen 361005, China, and also with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Digital Object Identifier 10.1109/LGRS.2020.3037509

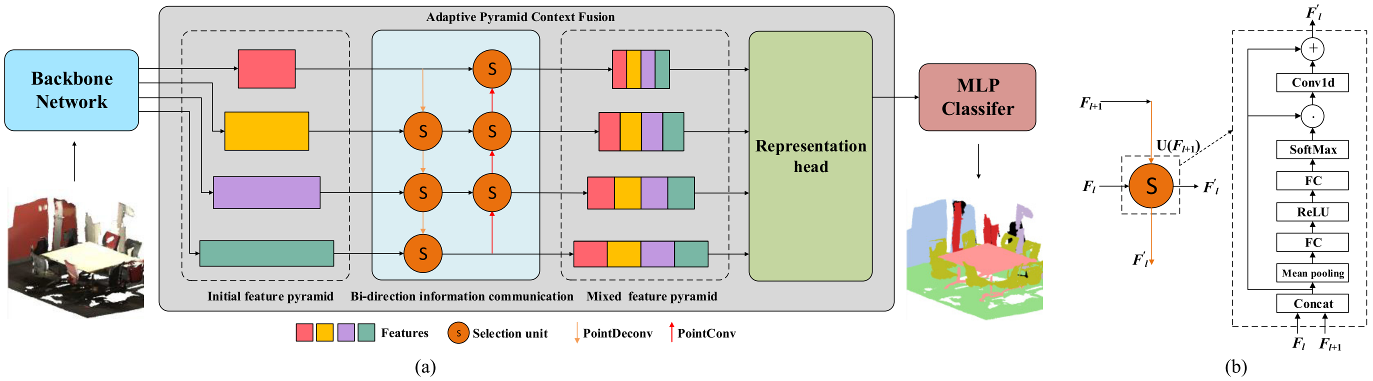

Fig. 1. (a) Framework of APCF-Net. The backbone network (PointConv) produces the initial feature pyramid that is composed of features from different network layers. After a bidirection communication, the information at each level in the pyramid is updated and mixed. Finally, the mixed feature pyramid is input to the representation head for producing the final fusion that is followed by the MLP classifier. (b) Selection unit.

combine the semantically strong and the fine-grained information effectively.

2) We propose a multiscale context-aware network named APCF-Net, which achieves large gains of effectiveness and efficiency over its vanilla counterpart.
3) Experiments on ModelNet40 [6] and ScanNet v2 [19] show that APCF-Net outperforms state-of-the-art methods for classification and semantic segmentation.

The rest of this letter is organized as follows. Section II details the proposed method. Section III presents the experiments. Section IV concludes this letter.

II. Method

Fig. 1(a) shows the architecture of our proposed APCF-Net, which consists of a backbone network, APCF, and the MLP classifier. Let $P \in R^{N\times(3+c)}$ be the input set of unordered points, where N is the number of points and $3 + c$ denotes the dimensions of coordinates and additional input signals (e.g., color or normal information). Through the backbone neural network PointConv [13], the features generated at different layers in the network are collected to construct a feature pyramid $\left\{F_l \in R^{N_l\times C_l} \mid 1 \le l \le L\right\}$, where l denotes the level index in the pyramid (corresponds to the layer index in the network), N_l denotes the number of points in L level, C_l denotes the channel number of F_l, and L denotes the number of levels in the pyramid. In APCF, to generate a more effective multiscale feature, we conduct a bidirection information communication among layers of the pyramid. After this communication, a representation head subnetwork combines these representations in different levels into the final fusion, which is followed by the classifier. In this section, we detail the structure of our APCF that is composed of bidirection information communication, selection unit, and representation head.

A. Bidirection Information Communication

As shown in Fig. 1, the input of the APCF module is the feature pyramid from the backbone neural networks. Considering the correlation of information between adjacent levels in the pyramid, we adopt a bidirection information communication scheme, according to which the communication can be divided into two stages: top-to-down stage and bottom-to-up stage. Inside the APCF, the upsampling $U(\circ)$ and downsampling $D(\circ)$ operations stride across every two adjacent levels, followed by the selection unit $S(\circ)$. At the top-to-down stage, for level l in the feature pyramid $\left\{F_l \in R^{N_l\times C_l} \mid 1 \le l \le L\right\}$, feature F_l is fused with feature in level $l + 1$, i.e., $F'_l = S(U(F_{l+1}), F_l)$. Before communication, a feature at each level in the pyramid contains only the information belonging to itself. Through each upsampling and selection unit, feature F_l at level l contains information not only from itself but also from adjacent higher level $l + 1$. By repeating upsampling and selecting, we obtain the bottom fusion $F'_1 = S\left(U\left(F'_2\right), F_1\right)$. Then, we conduct a similar process from bottom to up but replace $U(\circ)$ with the downsampling operation $D(\circ)$. After this bidirection information communication, we obtain a feature pyramid, in which multiscale contextual information is adequately mixed together. In this letter, we use the PointConv and PointDeconv operators [13] as our $D(\circ)$ and $U(\circ)$, respectively.

B. Selection Unit

Inspired by the SKNet [20] and SENet [21], we develop a dynamic mechanism that can adaptively reweight the information from different levels based on the multiscale input. Fig. 1 (b) shows the selection unit. Let F_l and F_{l+1} denote the low-level feature and the upsampled high-level feature, respectively. First, we concatenate them

$$F_{\text{cat}} = \text{concat}(F_l, F_{l+1}). \tag{1}$$

Second, we squeeze P_{cat} into a joint global descriptor by average pooling

$$z_c = \text{squeeze}(F_{\text{cat}}) = \frac{1}{N}\sum_{i=1}^{N} F_{\text{cat}}(i) \tag{2}$$

where N is the number of points. Third, we use a two-layer bottleneck MLPs to transform z_c into an attention mask

$$s = \text{MLP}(z_c) = \sigma(\mathrm{W}_2\delta(\mathrm{W}_1 z_c)) \tag{3}$$

where σ refers to the Sigmoid function, δ refers to the ReLu function, $\mathrm{W}_1 \in R^{C\times(C/r)}$, and $W_2 \in R^{(C/r)\times C}$. r is the reduction ratio, which is used to adjust the model complexity of the MLPs. Finally, the generated attentions are leveraged to conduct soft selection on the input representations and are mixed by a 1×1 convolution layer. Following SENet, we adopt the residual strategy to prevent from conducting the feature selection overly:

$$F_{\text{final}} = \delta(W_3(s \odot F_{\text{cat}})) + F_{\text{cat}} \tag{4}$$

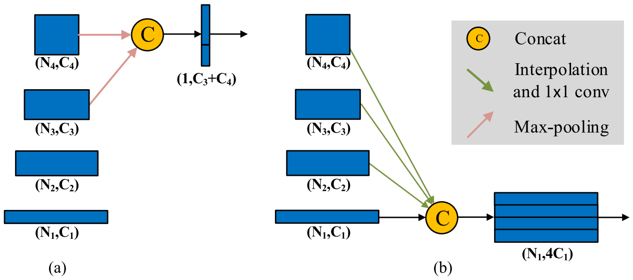

Fig. 2. Structure of (a) classification head and (b) segmentation head.

where $\odot$ refers to a channelwise multiplication. Different from the SKNet focused on selecting the scales and ignored the soft selection across channels, our selection unit can generate attention against both the scales and channels, which exploits more comprehensive dependence for the input representations.

C. Representation Head

After the bidirection information communication, we acquire a mixed feature pyramid, in which feature at each level contains multiscale contextual information. In this section, we mainly investigate how to transform this feature pyramid into an informative representation according to the downstream task. The key to 3-D object classification is to learn a global descriptor, which should focus more attention on the high-level representations. Unlike classification, semantic segmentation for point clouds aims at assigning a class label to each point, which has urgent demand in high-/full-resolution representations, corresponding to the low levels in the feature pyramid. Therefore, as shown in Fig. 3, we propose two kinds of representation fusion head for classification and segmentation, respectively.

1) Classification Representation Head: The output is the representation from the two high levels. Other low-level representations are ignored. This is illustrated in Fig. 2(a). As mentioned earlier, N_l and C_l denote the number of points in level L and the channel number of features, respectively.

2) Segmentation Representation Head: We rescale the spatial size and the number of channels of the high-level representations to the lowest level and then concatenate all the representations. The rescaling on the dimension is to prevent the upsampled high-level representations from dominating in the final fused representations, which disobeys our intention of preserving the detailed information. This is illustrated in Fig. 2(b).

III. Results and Discussion

We conducted several experiments to evaluate our proposed method. Sections III-A and III-B, respectively, evaluate the effectiveness of our methods for classification and segmentation task on ModelNet40 [6] and ScanNet v2 [19]. Section III-C analyzes the efficiency of APCF. Section III-E performs the ablation studies. In all experiments, we implement the models with Tensorflow on one Nvida Tesla V100 GPU.

A. Classification on ModelNet40

We evaluate our method on ModelNet40 [6] for object classification. ModelNet40 consists of 12311 CAD models in 40 classes. Following the official split, we use 9843 objects for training and 2468 objects for testing. We sample 1024 points randomly and compute the normal vectors from the mesh surface. We adopt the augmentation strategy as follows: random anisotropic scaling in the range [−0.8, 1.25], random translation in the range [−0.1, 0.1], and random dropout 20%. As shown in Table I, APCF-Net outperforms almost all state-of-the-art methods, including the previous state-of-the-art PoinASNL [11]. In particular, our result is 0.8% higher than PointConv [13], with which we share the same local feature extractor.

TABLE I
Overall Accuracy on ModelNet40 Data Set

Method	Accuracy(%)
PointNet [8]	89.2
LPCCNet [22]	90.2
PointGrid [23]	92.0
PointCNN [12]	92.2
DGCNN [10]	92.2
PatchCNN [15]	92.4
PosPool [16]	93.2
PointASNL [11]	93.2
PointConv [13]	92.5
ours	**93.3**

B. Sematic Segmentation on ScanNet V2

We use ScanNet v2 [19] to evaluate the effectiveness of our APCF-Net. ScanNet data set contains 1513 scanned indoor point clouds for training and 100 test scans. The labels of the test scans are publicly unavailable. Each point has been labeled with one of 21 categories. For comparison with other approaches, we submitted our results to the official evaluation server. For training, we randomly sample 1.5 m × 1.5 m × 3 m cube with 8192 points from the indoor rooms to generate training data. For evaluation, we use a sliding window with a 0.5-m stride over the entire rooms for five voting tests. The input of our model is pure 3-D coordinates without RGB values. We will show that the additional RGB information may not favor the prediction results on this data set in Section IV. The intersection over union (IoU) is used as our main measure.

We compare our APCF-Net with other state-of-the-art methods under the same training and testing strategy (randomly chopping cubes with a fixed number of points), e.g., PointNet++ [9], PointCNN [12], PointConv [13], HPEIN [24], and PointASNL [11].

As shown in Table II, APCF-Net outperforms all methods, including the previous state-of-the-art PoinASNL [11]. In particular, our result is 8% higher than PointConv [13], with which we share the same local feature extractor and upsampling operation. This huge performance gap demonstrates the superiority of our multiscale context fusion scheme.

Some example semantic segmentation results are visualized in Fig. 3. Due to the well-designed multiscale context fusion scheme, our APCF-Net performs better recognition of fine-grained details: for instance, the door in the first column, the picture in the wall in the second column, and the chair at the top left corner in the third column.

C. Efficiency

In this section, we evaluate the efficiency of our APCF-Net on ScanNet V2 from three aspects: 1) model complexity; 2) inference time; and 3) training time. For the model complexity, we use the parameter number of the network as the evaluation metric. For the inference time, we measure the total

TABLE II

SEGMENTATION RESULTS ON SCANNET V2 DATA SET IN MEAN PER-CLASS IOU(MIOU, %)

Method	mIoU(%)
PointNet++ [9]	33.9
PointCNN [12]	45.8
MS-PCNN [25]	56.8
HPEIN [24]	61.8
FusionAwareConv [26]	63.0
PointASNL [11]	63.0
PointConv [13]	55.6
ours	**63.1**

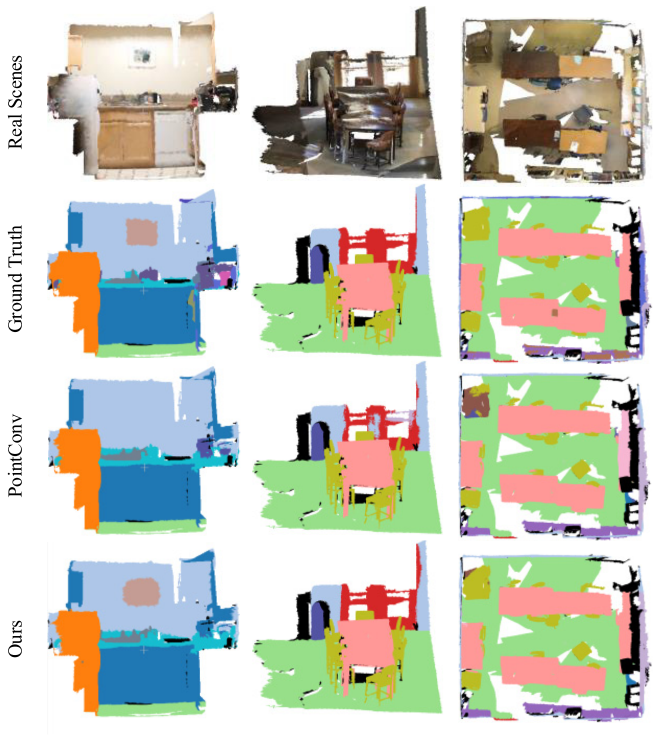

Fig. 3. Examples semantic segmentation results on ScanNet v2. We visualize and compare our segmentation results with PointConv [13]. Different colors denote different categories of the object in the real scene.

TABLE III

EFFICIENCY COMPARISON BETWEEN POINTCONV AND APCF-NET

Method	Test mIoU (%)	Parameters (millions)	Test time (seconds)	Training time (seconds)
PointConv [13]	55.6	21.7	76.2	156.6
ours	**63.1**	**13.6**	**61.7**	**146.3**

time consumption of the prediction process on Scene 568, which has three scans of point clouds in total. For the training time, we measure the total training time consumption for one epoch. Note that our APCF-Net shares the same local feature extractor and upsampling operation with PointConv [13]. For a fair comparison, we also evaluate the efficiency of PointConv.

As shown in Table III, our APCF-Net is able to surpass PointConv with almost 8% margin and enjoys a much higher computation and parameter efficiency. This is due to the effective multilevel representations' fusion scheme that we adopt, which enables us to use a simple representation head instead of an enormous decoder to rescale the multiple features to one resolution.

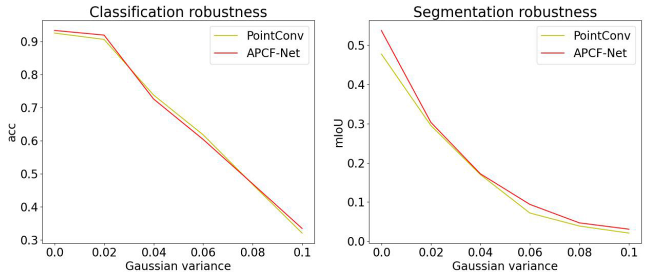

Fig. 4. Robustness evaluation on (Left) classification and (Right) segmentation.

D. Robustness

We also evaluate the robustness of our model compared with PointConv [13]. We add pointwise Gaussian perturbation to the input points. For classification, we evaluate the robustness of the whole test set. For segmentation, we add this Gaussian noise to a subset of the ScanNet validation set (19 out of 312 scans). We fix the mean of the Gaussian noise and tune its variance to control the perturbation strength. As shown in Fig. 4, APCF-Net has little difference with its baseline [13] against the Gaussian noise for both classification and segmentation.

E. Ablation Study

In this section, we conduct the following ablation studies for APCF. All ablated networks are trained by using the standard training/validation split provided by ScanNet [19].

(1∼2) Remove Fusion With the Highest/Lowest Level Feature in APCF: We, respectively, remove the fusion with the highest level representations and the lowest level to study their effect on the effectiveness of APCF.

(3∼6) Remove Bidirection Communication or Each Component of it: By removing top-to-down communication or bottom-to-up communication, the information can be transmitted only from top to bottom or bottom to up. After removing the selection unit, the remained upsampling and downsampling tend to hard combine the representations from different levels.

(7) Add RGB Input: Following PointConv [13], we compare the result of the model with and without RGB input.

(8) Stack More Bidirection Communication Modules: Inside APCF, we try to conduct bidirection communication one, two, and three times.

Tables IV and V show the compared mIoU scores of all ablated networks. We can observe the following.

1) The removal of the highest/lowest level representations has a significantly negative impact on the performance of the model. This suggests the rationality of our idea that exploiting multiscale context from different layers in the network can benefit the downstream task a lot.
2) The bidirection information communication module is necessary to effectively exchange information across different levels in APCF.
3) The respective removal of the top-to-down and bottom-to-up communications shows that the high- and low-level features are complementary with each other.
4) The selection unit favors the fusion of multiple representations.
5) Similar to PointConv [13], RGB information does not favor to the segmentation results.

TABLE IV

MEAN PER-CLASS IOU OF ALL ABLATED NETWORKS BASED ON OUR FULL APCF-NET

	Val mIoU(%)
(1) Remove the lowest-level fusion	63.9
(2) Remove the highest-level fusion	63.1
(3) Remove bi-direction communication	60.7
(4) Remove top-to-down communication	62.9
(5) Remove bottom-to-up communication	63.7
(6) Remove selection unit	62.5
(7) Add RGB input	65.0
(8) APCF	**65.1**

TABLE V

STACK MORE BIDIRECTION COMMUNICATION MODULES

Stack number	Val mIoU (%)	Parameters (millions)	Test time (seconds)	Training time (seconds)
1	65.1	**13.6**	**61.7**	**146.3**
2	65.3	20.9	104.8	199.1
3	**66.2**	28.2	173.2	265.4

6) Stacking more communication modules can bring about slightly higher generation power but at a much higher cost of efficiency. Therefore, we recommend the configuration with single communication for APCF to achieve a good balance between effectiveness and efficiency.

IV. CONCLUSION

We presented a novel module APCF that can adaptively capture scales of contextual information from features at different layers in the neural network. Experimental results on two challenge benchmarks demonstrate the effectiveness and efficiency of our methods. However, our APCF does not have a significant improvement in noise robustness, which is an orientation to be explored in our future work. Furthermore, our APCF module can be easily embedded into various neural networks for point cloud feature learning and, thus, has the potential to be applied to other downstream tasks, such as instance segmentation [27] and point cloud reconstruction [28]–[30], which heavily relies on fine-grained representation. We will explore such applications in the future.

REFERENCES

[1] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[2] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6630–6639.

[3] A. Boulch, J. Guerry, B. Le Saux, and N. Audebert, "SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks," *Comput. Graph.*, vol. 71, pp. 189–198, Apr. 2018.

[4] J. Balado, R. Sousa, L. Díaz-Vilariño, and P. Arias, "Transfer learning in urban object classification: Online images to recognize point clouds," *Autom. Construct.*, vol. 111, Mar. 2020, Art. no. 103058.

[5] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[6] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[7] C. R. Qi, H. Su, M. NieBner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656.

[8] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 984–993.

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NeurIPS*, 2017, pp. 5099–5108.

[10] 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, pp. 1–12, Nov. 2019.

[11] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5588–5597.

[12] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *NeurIPS*, 2018, pp. 820–830.

[13] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9613–9622.

[14] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6410–6419.

[15] F. Wang, X. Zhang, Y. Jiang, L. Kong, and X. Wei, "Patch-CNN: An explicit convolution operator for point clouds perception," *IEEE Geosci. Remote Sens. Lett.*, early access, Apr. 1, 2020, doi: 10.1109/LGRS.2020.2981507.

[16] Z. Liu, H. Hu, Y. Cao, Z. Zhang, and X. Tong, "A closer look at local aggregation operators in point cloud analysis," in *Proc. ECCV*, 2020, pp. 1–26.

[17] G. Justin, S. S. Samuel, F. R. Patrick, V. Oriol, and E. D. George, "Neural message passing for quantum chemistry," in *Proc. ICML*, 2017, pp. 1263–1272.

[18] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. CVPR*, 2018, pp. 7794–7803.

[19] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2432–2443.

[20] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[21] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[22] M. Li, Y. Hu, N. Zhao, and L. Guo, "LPCCNet: A lightweight network for point cloud classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 962–966, Jun. 2019.

[23] T. Le and Y. Duan, "PointGrid: A deep network for 3D shape understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9204–9214.

[24] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia, "Hierarchical point-edge interaction network for point cloud semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10432–10440.

[25] L. Ma, Y. Li, J. Li, W. Tan, Y. Yu, and M. A. Chapman, "Multi-scale point-wise convolutional neural networks for 3D object segmentation from LiDAR point clouds in large-scale environments," *IEEE Trans. Intell. Transp. Syst.*, early access, Dec. 27, 2019, doi: 10.1109/TITS.2019.2961060.

[26] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "Fusion-aware point convolution for online semantic 3D scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4534–4543.

[27] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. NieBner, "3D-MPA: Multi-proposal aggregation for 3D semantic instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9028–9037.

[28] Y. Tang *et al.*, "Vision-based three-dimensional reconstruction and monitoring of large-scale steel tubular structures," *Adv. Civil Eng.*, vol. 2020, pp. 1–17, Sep. 2020.

[29] M. Chen *et al.*, "Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology," *Comput. Electron. Agricult.*, vol. 174, Jul. 2020, Art. no. 105508.

[30] M. Chen, Y. Tang, X. Zou, K. Huang, L. Li, and Y. He, "High-accuracy multi-camera reconstruction enhanced by adaptive point cloud correction algorithm," *Opt. Lasers Eng.*, vol. 122, pp. 170–183, Nov. 2019.