SAPCS: Semantic Attention Empowered Point Cloud Segmentation

Yiliang Zhang,[1] **Honglei Zheng,** [1] **Weijie Huang,** [1] **Jinhao Deng,** [1] **Yang Zheng** [1]

[1] School of Infomatics, Xiamen University
23020211153911,23020211153912,31520211154059,31520211154031,23020211153992

Abstract

Attention mechanism is the core of transformer, which was first applied to the field of natural language processing. Recent research has shown that the attention mechanism can also work well in the realm of computer vision. Subsequently, more and more tasks in image analysis such as image enhancement and pose estimation was explored using attention mechanism to empower, and is making impressive strides. Inspired by these success, we apply attention mechanism to segmentation task in 3D point cloud, and propose our SAPCS. In particular, we use multi-layer perceptron to aggregate information from neighbors of each point, thus applying semantic attention for a certain point. Moreover, to reduce the points we need to process for speeding up the model, we add several down-sampling and pooling block. And to recover the points that have been cut, we add several interpolation layers before producing results. Experimental results show that, on the challenging S3DIS dataset, our SAPCS get an mIoU of 68.6%, achieving competitive performance with SOTA while remaining fast running speed.

Introduction

Point cloud features have plenty of downstream applications, such as autonomous driving, AR, and robotics. With the development of LiDAR technology for collecting point clouds, point clouds have gained wide attention in recent years because of their lower cost and higher accuracy. However, point cloud itself has irregularity, disorder and sparsity, which makes feature learning and down stream task as segmentation on point cloud much more difficult than that on image.

There has been a lot of research on extracting better point cloud features for downstream tasks. Among these tasks, segmentation is one of the most challenging one. At present, most of the methods dealing with segmentation using point cloud features learning with voxel-based method or point-based method(Li, Zhang, and Xia 2016; Qi et al. 2017a; Shi et al. 2015; Su et al. 2015; Wei et al. 2016) . Attention mechanism was first proposed as the a core of transformer, which was originally used for natural language processing, but recent studies have shown that it is surprisingly effective in vision feature extraction (Chen et al. 2020; Fan, Yang, and Kankanhalli 2021; Hou et al. 2020; Zhao et al. 2017; Zheng et al. 2021). However, its use in point clouds is rarely explored.

With inherent irregularity, disorder and sparsity, we argue that attention mechanism is particularly appropriate for point cloud processing and its downstream task. Because attention is in essence a set operator: attention itself is invariant to permutation of the input elements. The application of attention to 3D point clouds is therefore quite natural, since point clouds are essentially sets embedded in 3D space. With this intuition, we further design a semantic attention layer for 3D point cloud processing. Following the attention mechanism, We investigate the form of the semantic attention operator, the application of attention to local neighborhoods around each point, and the encoding of positional information in the network. The resulting networks are based purely on self-attention and point-wise operations. Based on this network, we can get better performance.

Although using attention in point cloud feature extraction can achieve good performance, because of the huge amount of data per frame of point cloud, at the same time, it will increase the computing overhead, which makes it difficult to complete tasks with high requirements on real time. Therefore, in this work, we further improving the arithmetic speed by introducing sampling and pooling to reduce points needed to process, and using feature recovery module to recover those reduced points.

Finally, our SAPCS consist of three main parts: feature learning module, feature transfer module and feature recovery module.(1) Feature learning module integrates more features of other points in space to learn features of the current point. It mainly contains four sub-modules: Firstly, the input is sampled once and the features of other points in the space are fused as much as possible. We use FPS(farthest point sampling) to ensure the sampling effect while accelerating. Then, for each point, an MLP is used to fuse the features of more nearest points. Finally, after MLP processing, maximum pooling is adopted to retain the main features, and then attention is used to learn the degree of influence of each feature on the final feature results of this point to obtain a better feature representation. (2) The feature transfer module can better encode and transfer features through MLP and attention mechanism. In this module, the main flow of data is to transform features through MLP first, and then

use attention mechanism to give more reasonable weight to feature relations.(3)Feature recovery module We use trilinear interpolation method to recover features. After learning with the multi-layer feature transfer module, feature learning module and multi-layer feature recovery module, the point cloud features integrated with the attention mechanism will be obtained. Finally, the prediction head corresponding to segmentation task is used for predicting the class and confidence of each point.

We apply the model to a challenging mainstream data set, S3DIS, and measure and compare the accuracy and time cost. Experimental results show that our method is at least 1.5% more accurate than a series of voxel and point based methods while remaining high running speed.In general, our contribution is mainly as follows

- We propose SAPCS, which applies attention mechanism to point cloud segmentation task. By carefully designing corresponding modules, attention mechanism using semantic information fits well with the inherent properties of point cloud and turns the disadvantages of point cloud to advantages.
- We designed the connection mode between each module to enable end-to-end training and focus on the progress improvement of segmentation on point cloud.
- Experiments verify that on segmentation task, our SAPCS achieves significant improvement over a series of voxel-based and point-based methods on the challenge S3DIS benchmark.

Related Works

Traditional convolution can achieve more impressive results on 2D image data. Compared to the fix data structure of 2D image data, the disorder and irregularity of point cloud data makes its processing more difficult.The processing of point cloud data has become very important as it has important applications in autonomous driving, autonomous robotics, etc. Besides, attention mechanism also shows strong feature extraction and sequence data processing capabilities in the field of semantic segmentation of point cloud images. Therefore, in this project, we will use this structure for semantic segmentation of single frame point cloud and can prospect better result.

Fixed-based Networks

Due to the success of traditional convolution, ways to apply traditional convolution to point cloud data have long been exploited. However, due to the irregularity and disorder of point cloud data, we need to transform point cloud data to a fixed data structure before we can learn its feature for further segmentation task. There are usually two ways to do this, one of which is to map the point cloud data, and the other, more widely used, is to voxelise the point cloud data.

To take advantage of the well-performing 2D convolution, many studies have represented point cloud data as 2D picture data. And then use the traditional CNN to process the 2D data (Su et al. 2015; Wei et al. 2016; Shi et al. 2015; Li, Zhang, and Xia 2016). However, the transform from point cloud data to 2D image data is computationally intensive and the information in 3D space is lost.

Another widely used approach is 3D voxelization. Projecting point cloud data into a raster of Euclidean space, and the regular 3D grid is suit for use the standard CNN operator (Maturana and Scherer 2015; Wu et al. 2015).Although this method preserves spatial information, the resolution of the raster is difficult to choose; too small a resolution loses a great deal of information, and too large a resolution results in an unsustainable computational and memory cost. Also because of the sparse feature of the point cloud data, there are large amounts of memory resources being wasted.

Point-based Networks

Due to the disadvantage of above approach, a lot methods directly process the point cloud data are proposed. PointNet (Qi et al. 2017a) is a pioneer in the direct processing of point cloud structures and uses symmetry functions to achieve order invariance. To address the disadvantage of PointNet (Qi et al. 2017a) which ignores local features, PointNet++ (Qi et al. 2017b) uses PointNet (Qi et al. 2017a) as the basic structure to build a multi-level network structure that can focus on both local and global features. SpiderCNN (Xu et al. 2018) propose the parameterized convolutional filters to process the point clouds, the filter weights are calculated from a family of polynomial functions. KPConv (Thomas et al. 2019) introduce the flexible and deformable convolution to deal with the unregular point clouds.The filter weights are obtrained from the local coordinate and can adapt to different point density distributions.

Attention and Transformer

In the current field of image semantic segmentation, convolutional neural network model plays a dominant role, and with the gradual deepening of neural network, this advantage becomes more and more obvious. However, due to the lack of memorization of CNN, the correlation of extracted feature graphs is not ideal for image sequence data with time series. Attention mechanism, which shows the processing ability of sequence data in NLP field, has attracted the attention of this field. Ashish Vaswani first proposed Transformer in 2017, which was applied in the field of NLP (Vaswani et al. 2017). Transformer completely abandons the convolution operation and innovatively uses self-attention structure to extract feature in statements, achieving the SOTA effect of CNN model. the iGPT (Chen et al. 2020) model applies Transformer in the visual field for the first time, and can achieve the same accuracy as CNN model in classification. SETR (Zheng et al. 2021) applies attention mechanism in semantic segmentation task of image sequence data for the first time, and proposes a new sequence to sequence model, which MIoU and PA in ADE20K and Pascal Context dataset surpasses the networks that only use CNN . And the P4Transformer (Fan, Yang, and Kankanhalli 2021) also has achieved good results in instance segmentation of point cloud video sequence data.

Methods

We use end-to-end processing structure to divide the semantic segmentation tasks of point clouds into multi-step tasks, and composed into the same network. The end-to-end structure ensures that semantic segmentation function is the optimal result in the whole tasks. In this chapter, after a detailed review of the structure and calculation methods of transformer and attention mechanism, we use the semantic attention structure to process the input point cloud image and extract relevant features. Then, the final task is to complete the semantic segmentation based on 2D point cloud image in an end-to-end process.

Prior Knowledge

Transformer and self attention have been applied to the field of computer vision, and the effect of large-scale image processing has exceeded the traditional convolutional neural network architecture. The self-attention mechanism uses position embedding to correlate the position relationship of the segmented image, and uses the trigonometric function to realize the relative position coding between each patch and pixels, as shown in (1) (2),

$$\upsilon\left(pos_i\right) = PE\left(pos_i, 2i\right) = \sin\left(\frac{pos_i}{10000^{\frac{2i}{d_{model}}}}\right) \tag{1}$$

$$\upsilon\left(pos_j\right) = PE\left(pos_j, 2i+1\right) = \cos\left(\frac{pos_j}{10000^{\frac{2i}{d_{model}}}}\right) \tag{2}$$

where pos represents the abscissa and ordinate of 2D point cloud pixels, i represents the relative position between images, and d_{model} represents the dimension of images.

After completing position embedding, Self-attention will perform vector mapping. In the self-attention layer, the input vector A=$\{ a_1, a_2 , a_3 \ldots \}$ is first transformed into three different vectors: the query vector q, the key vector k and the value vector v. V is the mapping vector passing through two linear layers and a MLP, K and Q are the mapping completed through two linear layers, as shown in (3) (4) (5), where ε and δ represent two linear layers.μ represents two linear layers and MLP, both are mapping functions, a_i and a_j are the horizontal and vertical coordinates of the pixel.

$$Q = \varepsilon\left(a_i\right) \tag{3}$$

$$K = \delta\left(a_j\right) \tag{4}$$

$$V = \mu\left(a_i, a_j\right) \tag{5}$$

The standard scalar dot-product attention layer can be expressed as(Vaswani et al. 2017):

$$y_i = \sum_{a \in A} \tau\left(\varepsilon\left(a_i\right)^{\top} \delta\left(a_j\right) + \rho\right) \cdot \mu\left(a_i, a_j\right) \tag{6}$$

where y_i is the output feature,τ is the normalization function such as the softmax function, ε and δ represent the mapping function, ρrepresents the position encoding.

Through the extraction of feature maps, attention can be focused on the areas that need to be segmented. At the same time, the weight function α , such as MLP, is used to distribute the weight between each patch to achieve a set of global features and local features. When calculating the point cloud vector, due to the particularity of vector attention, the original scalar attention is deformed by the residual, as shown in (7):

$$y_i = \sum_{a \in A} \tau\left(\alpha\left(\varepsilon\left(a_i\right), \delta\left(a_j\right)\right) + \rho\right) \odot \mu\left(a_i, a_j\right) \tag{7}$$

where α represents the relationship function between two position codes, such as residual connection or subtraction, connecting two position codes.

Semantic Attention

The location embedding feature of Transformer structure and Self-attention mechanism is very conducive to semantic segmentation in 2D point cloud images. We design the semantic attention Layer based on this structure and position encoding. The structure is based on the input 2D point cloud vector, adding an additive relationship to the vector mapping and position encoding, and put weight parameters and position coding into the key vector K, T, as shown in (8)(9):

$$y_i = \sum_{a \in A} \theta \cdot \tau\left(\alpha\left(\varepsilon\left(a_i\right), \delta\left(a_j\right)\right) + \rho\right) \odot \left(\mu\left(a_i, a_j\right) + \rho\right) \tag{8}$$

$$\rho = \upsilon\left(pos_i\right) \odot \sigma\left(pos_j\right) \tag{9}$$

The vector A a1, a2, a3... is the coordinates of 8 adjacent points in the point cloud, and θ is the weight function between each pixel. Therefore, the semantic attention layer will be used to calculate the local features between points in the 2D point cloud image, and the local features and global features are fused through the weight parameter θ. The point transformer layer is illustrated in Figure 1.

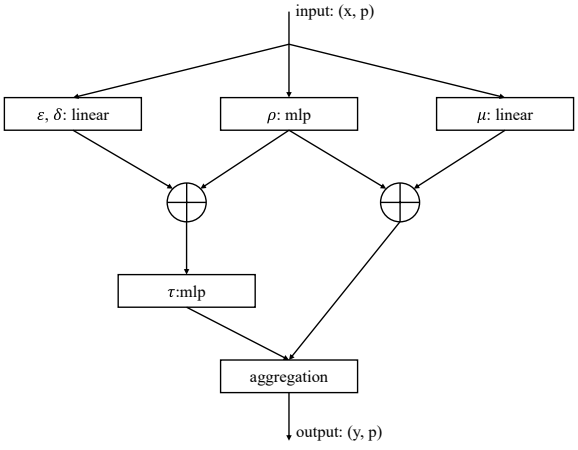


Figure 1: Point transformer layer.

Network Architecture

We construct a residual point transformer block with the point transformer layer as the core, as shown in Figure 3(a). The transformer block is integrated with self-attention layer,

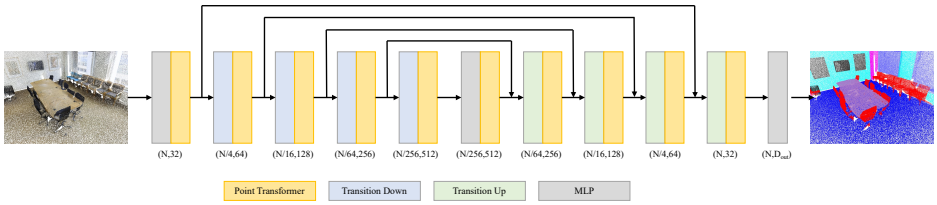
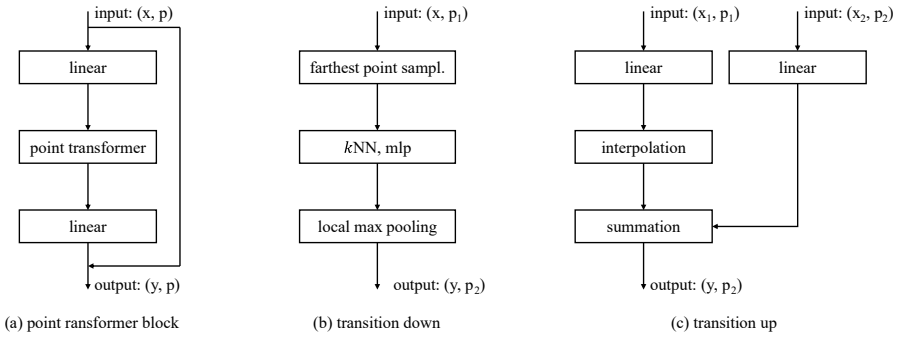


Figure 2: Point transformer networks for semantic segmentation.



(a) point ransformer block (b) transition down (c) transition up

Figure 3: Detailed structure design for each module.

linear projection which can reduce dimension and accelerate processing and residual connection. The input is a set of feature vectors x and the associated three-dimensional coordinates P. The point transformer block facilitates the information exchange between these localized feature vectors and generates new feature vectors for all the data points it outputs. Based on the point transformer block, a 3D point cloud understanding network is constructed. We do not use convolution for preprocessing branches: the network is entirely based on point transformer layers, pointwise transformations, and pooling. The network architectures are shown in Figure 2.

Backbone Structure

The feature encoder of the point transformer network for semantic segmentation has five stages, which act on progressively down-sampled point sets. Since the down-sampling rate of each stage is [1,4,4,4,4], the cardinality of the point set generated in each stage is [N, N/4, N/16, N/64, N/256], where N is the number of input points. Consecutive stages are connected by transition modules: transition down for feature coding and transition up for feature decoding.

Transition Down

A key function of the transition down module is to reduce the cardinality of the point set as needed. We perform farthest point sampling in $\mathcal{P}_\infty$ (Qi et al. 2017b) to identify a well-distributed subset $\mathcal{P}_\in \subset \mathcal{P}_\infty$ with the necessary cardinality. Each input feature is linearly transformed, followed by batch normalization and ReLU, followed by max pooling from k neighbors in $\mathcal{P}_\infty$ to each point in $\mathcal{P}_\in$. The transition down module is shown in Figure 3(b).

Transition Up

For intensive prediction tasks such as semantic segmentation, U-net design is adopted, in which the encoder mentioned above is combined with a symmetric decoder (Qi et al. 2017b; Choy, Gwak, and Savarese 2019). The consecutive stages in the decoder are connected by the transition up modules. Their main function is to map the features of the downsampled input point set $\mathcal{P}_\in$ to its superset $\mathcal{P}_\in \subset \mathcal{P}_\infty$. To this end, each input point feature is linear layer processed, followed by batch normalization and ReLU processing, and then the feature is mapped to the higher-resolution point set $\mathcal{P}_\infty$ by trilinear interpolation. These interpolation features from the previous decoder stage are summarized with those from the corresponding encoder stage, which are provided by a skip connection. The upward transition module is shown in Figure 3(c).

Experiments

In this section, we validate the model on the point cloud semantic segmentation task. We compare the results with several existing network structures

DataSet

Stanford Large-Scale 3D Indoor Spaces Dataset (Armeni et al. 2016) is widely used in semantic scene segmentation task. This dataset includes 3D scan point clouds for 6 indoor areas including 272 rooms in total. This dataset includes 11 scenarios e.g. office, conference room, hallway, auditorium, open space, lobby, lounge, pantry, copy room, storage and WC.In total, there are 13 semantic labels e.g. ceiling, floor, wall, beam, column, window, door, chair, table, bookcase, sofa, board and clutter. Each point is labelled as one of these categories. Area 5 is withheld during training and is used for

Method	OA	mACC	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet (Qi et al. 2017a)		49.0	41.1	88.8	97.3	69.8	**0.1**	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
SegCloud (Tchapmi et al. 2017)		57.4	48.9	90.1	96.1	69.9	0.0	18.4	38.4	23.1	70.4	75.9	40.9	58.4	13.0	41.6
PointWeb (Zhao et al. 2019)	87.0	66.6	60.3	92.0	**98.5**	79.4	0.0	21.1	59.7	34.8	76.3	88.3	46.9	69.3	64.9	52.5
PointCNN (Li et al. 2018)	85.9	63.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
KPConv (Thomas et al. 2019)		72.8	67.1	92.8	97.3	82.4	0.0	23.9	58.0	69.0	**81.5**	**91.0**	**75.4**	**75.3**	66.7	**58.9**
SAPCS	**90.3**	**74.6**	**68.6**	**94.2**	**98.5**	**85.6**	0.0	**36.0**	**62.9**	**78.0**	81.3	89.6	63.1	71.0	**73.9**	56.6

Table 1: Semantic segmentation results on the S3DIS dataset, evaluated on Area 5.

Method	OA	mACC	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
SAPCS	90.3	74.6	68.6	94.2	98.5	85.6	0.0	36.0	62.9	78.0	81.3	89.6	63.1	71.0	73.9	56.6
Channel	52.1	30.0	20.2	65.9	0.0	55.8	0.1	2.2	0.4	13.8	28.6	16.7	5.3	45.4	14.8	13.8

Table 2: Ablation study between Point Transformer layer and Channel Attention layer

testing, the remaining five as the training set. To evaluate the semantic segmentation of our model, we use the standard 6-fold cross-validation in our experiments. During training, the input points are voxelised in order to reduce the density of points and enhance the training efficiency, as well as for the convenience of KNN operation.

Metric

For fair and diverse comparisons, our metrics for the results use the mean IoU(mIoU), mean class Accuracy(mAcc) and Overall Accuracy(OA) of the total 13 classes.

Implementation details

We implement the model in PyTorch (Paszke et al. 2019). We use the SGD optimizer with momentum and weight decay set to 0.9 and 0.0001, respectively. Voxel size is set to 0.04 and the maximum number voxels is set to 80000. We train for 100 epochs with initial learning rate 0.5, dropped by $10\times$ at the epochs 60 and 80. A distributed training scheme is further implemented on four NVIDIA GEFORCE RTX 3090 GPUs to maintain the training batch size which is set to 16.

Result

As shown in Table 1, compared to some work in recent years sycg as PointNet (Qi et al. 2017a), SegCloud (Tchapmi et al. 2017), PointWeb (Zhao et al. 2019), PointCNN (Li et al. 2018) and KPConv (Thomas et al. 2019). SAPCS has achieved the highest accuracy in OA, mACC and mIoU. In the mIoU comparison, SAPCS is 1.5% more accurate than the previous highest accuracy KPConv and 1.8% more than the previous best accuracy KPConv in mACC. SAPCS achieves the highest accuracy in mIoU in six of the thirteen classes.

Ablation Study

We replace the point transformer layer implemented in MLP with the Channel Attention Module in MPRM (Wei et al. 2020) for the ablation experiment. The result shown in Table 2, the experimental results fully demonstrate the effectiveness of the transformer module in our model.

Conlusion

Transformers have revolutionized natural language processing and are making impressive gains in 2D image analysis.Inspired by this progress, we have developed a transformer architecture for 3D point clouds on segmentation tasks. Through analyzing, transformers are perhaps an even more natural fit for point cloud processing than they are for language or image processing, because point clouds are essentially sets embedded in a metric space, and the self-attention operator at the core of transformer networks is fundamentally a set operator. By combining the attention mechanism with the point cloud segmentation task, we achieved an significant improvement over the existing methods while remaining fast running speed. Empirical experiments on several segmentation tasks in different datasets show the effectiveness of our method. In the future, based on the existing work, we will extract more excellent features through gaussian mixture model and take a series of measures to reduce the time complexity of the model from both efficiency and accuracy. Apart from these,applying Transformer to other tasks in the point cloud is also one of the areas that will be studied. We sincerely hope that our work can provide some reference and help for further investigation of the properties of point transformers, the development of new operators and model architecture designs, and the practical application of transformers to other tasks, such as 3D object detection, shape classification and so on.

References

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*.

Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; and Sutskever, I. 2020. Generative pretraining from pixels. In *International Conference on Machine Learning*, 1691–1703. PMLR.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3075–3084.

Fan, H.; Yang, Y.; and Kankanhalli, M. 2021. Point 4D Transformer Networks for Spatio-Temporal Modeling in Point Cloud Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14204–14213.

Hou, Q.; Zhang, L.; Cheng, M.-M.; and Feng, J. 2020. Strip pooling: Rethinking spatial pooling for scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4003–4012.

Li, B.; Zhang, T.; and Xia, T. 2016. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31: 820–830.

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

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*.

Shi, B.; Bai, S.; Zhou, Z.; and Bai, X. 2015. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12): 2339–2343.

Su, H.; Maji, S.; Kalogerakis, E.; and Learned-Miller, E. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, 945–953.

Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; and Savarese, S. 2017. Segcloud: Semantic segmentation of 3d point clouds. In *2017 international conference on 3D vision (3DV)*, 537–547. IEEE.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6411–6420.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wei, J.; Lin, G.; Yap, K.-H.; Hung, T.-Y.; and Xie, L. 2020. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4384–4393.

Wei, L.; Huang, Q.; Ceylan, D.; Vouga, E.; and Li, H. 2016. Dense human body correspondences using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1544–1553.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; and Qiao, Y. 2018. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 87–102.

Zhao, H.; Jiang, L.; Fu, C.-W.; and Jia, J. 2019. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5565–5573.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6881–6890.