Point Cloud Classification and Segmentation Using Channel-Aware Dynamic Convolutional Neural Network

Bifang Ma

Abstract—**The existing large scale point cloud recognition and segmentation methods based on deep learning only focus on recognition accuracy. They always ignore the memory and computational complexity limitations in practical applications (eg autonomous driving applications). A large-scale 3D model recognition and segmentation method based on lightweight channel-aware dynamic convolutional neural network is proposed in this paper. Firstly, a channel-aware module is constructed to adaptively predict the most important input feature channels of point clouds. It reduces the computational costs while ensuring the integrity of the original network structure and improves the recognition ability of the model. Secondly, by constructing a feature channel attention predictor, we design a lightweight channel-aware dynamic convolution. It can dynamically prune some unimportant feature channels of point cloud and reduce redundant calculation costs. Finally, a channel-aware dynamic dynamic convolutional neural network is constructed to efficiently recognize and segment large-scale point clouds with low complexity. Experiments are carried out on the 3D model recognition dataset ModelNet40 and the large-scale point cloud indoor segmentation dataset S3DIS and the outdoor segmentation dataset vKITTI, respectively. Experiments show that our method not only has higher recognition and segmentation accuracy than existing methods, but also has lower computational and memory complexity.**

Index Terms—**Point cloud, recognition and segmentation, channel-aware, dynamic convolution**

I. Introduction

WITH the wide application of 3D depth sensors, the number of 3D models is rapidly increasing. The object recognition and semantic segmentation of 3D models, as the premise and foundation of 3D model analysis and processing, have become important research in the field of machine vision. 3D object recognition and model semantic segmentation are completed by comparing the differences between feature descriptors. Therefore, the key issue is how to extract accurate and robust three-dimensional feature descriptors. Traditional methods use hand-designed shape descriptors to extract the features of the 3D model, such as geometric shape descriptors [1] and hot-core signature descriptors [2], etc. However, the hand-designed feature descriptors rely heavily on expert experiences, and have poor generalization ability. In recent years, deep learning [3] methods have achieved certain phased results in the field of machine vision for many tasks [4], [5], [6], [7], [8]. More and more scholars have begun to try to use deep learning methods for three-dimensional recognition and semantic segmentation. The main methods are divided into multi views-based methods, voxel-based methods and point cloud representation-based methods.

Manuscript received August 01, 2021; revised March 24, 2022.

This work was supported by the Guiding Project Fund of Science and Technology Department of Fujian Province (No. 2020H0027).

Bifang Ma, the corresponding author, is an associate professor of School of Electronic and Mechanical Engineering, Fujian Polytechnic Normal University, Fuqing, 350300, P.R.China. (e-mail: mabf555@126.com).

Multi views-based methods. Due to the irregularity of the 3D point cloud, it is difficult to extract features from the 3D point cloud directly. Literature [9] first performs multi-directional rendering on the 3D model to obtain 2D projection views, and then inputs the 2D multi-view as training data into the classic VGG [10] to train and extract features, and finally through view pooling to aggregate the view features to obtain a one-dimensional global feature descriptor. Although this method improves the accuracy of 3D model recognition, it has the problems of view feature redundancy and loss of 3D model geometric information.

Voxel-based methods. Literature [11] proposes to regularize irregular point cloud into 3D voxel grids, and then uses a three-dimensional convolutional neural network to directly act on the 3D voxel data to extract feature descriptors. Literature [12] converts the point cloud data into a binary 3D voxel matrix, and extracts the features of the voxel matrix through the stochastic gradient descent algorithm with regularization items to predict the category of models. Although the above algorithms effectively retain the geometric structure of the model, the memory consumption of the voxelization operation is serious, which makes it difficult to capture high-resolution information and fine-grained features. Because the recognition accuracy of low-resolution models is not high, the literature [13] proposes a space division method, but it cannot still capture local geometric features.

Point cloud representation-based methods. These methods can directly use matrix operations to perform the affine transformation on the point cloud, avoiding the complicated operation of transforming the point cloud into other regular data. It has been widely used in the fields of computer graphics and machine vision, such as indoor navigation [14], autonomous driving [15] and robots [16], etc. For 3D point cloud recognition and semantic segmentation, the PointNet [17] has become a pioneer in applying the deep learning framework directly to the original point cloud data, but PointNet only focuses on the features of independent points and does not consider local neighborhood information. Literature [18] proposes the PointNet++ network, which extracts fine-grained features by hierarchically dividing local point clouds, and shows good performance for 3D point cloud model recognition and semantic segmentation. Although the network effectively captures the local neighborhood information of the point cloud, it does not consider the distance measurement between points in the local neighborhood, and lacks the ability to capture the fine-grained local geometric information, resulting in poor recognition results. The DGCNN [19] constructs a local graph by searching *k* neighboring points of each sampling point, and captures the relative relationship between point pairs in the local point cloud to fully mine local geometric features. Although the recognition accuracy of the model is improved, it causes the problem of high computational and memory complexity. In summary, the existing methods only focus on how to improve the accuracy of model recognition, and have less consideration of memory and computational complexity in practical applications.

Therefore, we propose a 3D point cloud recognition and large-scale scene segmentation method based on a channel-aware dynamic convolutional neural network, which can obtain strong feature representation capabilities with lower computing costs and fewer parameters, and improve the accuracy of recognition and model semantic segmentation of point cloud. The main innovations and contributions are: (1) Aiming at the problem that the existing convolution complexity is high, a channel-aware strategy is proposed to predict the importance of feature channels automatically. (2) Based on the channel-aware strategy, a channel-aware dynamic convolution is proposed, which can adaptively prune unimportant feature channels of point cloud. (3) We build a lightweight channel-aware dynamic convolutional neural network for point cloud processing with lower complexity and higher recognition accuracy.

II. Channel-Aware Dynamic Convolution

To reduce the complexity of the deep neural networks, we intend to design a channel-aware dynamic convolution that can learn to select and adaptively obtain a strong feature representation ability with lower computational costs and fewer parameters according to different input data.

A. Channel-Aware Module

As shown in Figure 1, to reduce the computational costs, we propose to design a dynamic pruning strategy that can be used to select import channels. After the point cloud is preprocessed to obtain input features, the Channel-Aware Dynamic Convolution (CADConv) first selects the input feature channels and parameters of convolutional kernels, and equips each block with an input channel selector. According to the importance score of the input channels generated by the channel-aware coefficients, the channel-aware coefficients is used to dynamically determine the most important subset of the input channels, and then the selected important input channel subset is subjected to a convolution operation, and feature maps are output in each block. The dynamic pruning mechanism of CADConv is to adaptively sparse the connection of the input channels and the output channels, so that each block can automatically select a small part of the most relevant input channels to participate in the convolution calculation.

$$G(x_k) = F(x_k, \Theta) \otimes A(x_k, \psi), \tag{1}$$

where G is the channel-aware convolution, x is the input features, $x_k \in x$ represents the feature subset of the input channels of the i-th block, F is the convolution kernel containing parameters Θ, A is the pruning matrix containing parameters ψ, and calculates the significance score of the input channel selector, $\otimes$ represents the matrix dot multiplication operation. The channel-aware coefficients assign a score representing its importance to each input channel to select different input channels adaptively dynamically. The channel-aware coefficients of each block enables different blocks to select the most important subset of the input channels, diversifying the feature representation. For the i-th block, the calculation formula of the significance score S_i is as follows:

$$S_i = A(x_k, \psi) = FC(MaxPool(RuLU(BN(x_k)))) \tag{2}$$

where $S_i \in R^{C\times 1}$ is the channel-aware score of the i-th block input channels, *BN* is the Batch Normalization, and *FC* are the trainable weights and biases, respectively. After the saliency score is obtained, the most important input channels are selected according to the saliency score and added to the convolution kernel of the current block for convolution calculation. We set a pruning threshold to select a specific number of channels for each block in the CADConv layer. Therefore, all channels with a saliency score less than the threshold in the input channels are discarded, to effectively retain important input channels. The advantage of CADConv is that it can significantly reduce the computational and memory complexity of the network while ensuring recognition capability.

B. Channel-Aware Dynamic Convolution

After channel-aware, the use of traditional convolution in each block to process point cloud data will result in redundant parameters in the convolution kernel, which will cause a lot of waste of computing resources. This subject intends to select the input channels of the 3D point cloud into blocks, and then further design the channel-aware dynamic convolution for feature extraction at lower computational costs. Channel-aware dynamic convolution aims to adaptively select the corresponding important feature channels according to the different 3D point clouds, dynamically allocates a corresponding amount of computing resources for different inputs, and further reduce the redundant computing costs in the traditional convolution kernel. First, the channel-aware dynamic convolution uses a channel attention predictor to calculate the channel attention coefficients based on the input features to predict the importance of the output feature channels. The channel attention predictor consists of a max pooling layer, a fully connected layer and an activation function layer. The max pooling layer aggregates the feature maps containing all the feature channels into a one-dimensional feature vector, and the fully connected layer maps the one-dimensional feature vector to the channel attention coefficients, and uses the sigmoid activation function to non-linearly map the channel attention coefficients. The calculation formula for channel-aware dynamic convolutions is as follows:

$$Pruning(W) = S(\sum_{k=1}^{g} \partial_k \odot \mathrm{W}_k), \tag{3}$$

where $Pruning$ is the channel selection operation, S is the Sigmoid activation function, g is the number of blocks,

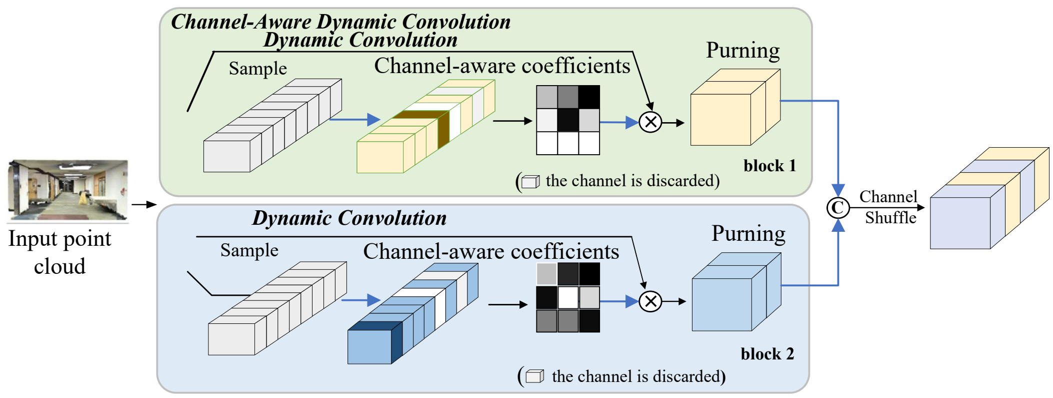

Fig. 1. The Structure of Channel-Aware Dynamic Convolution.

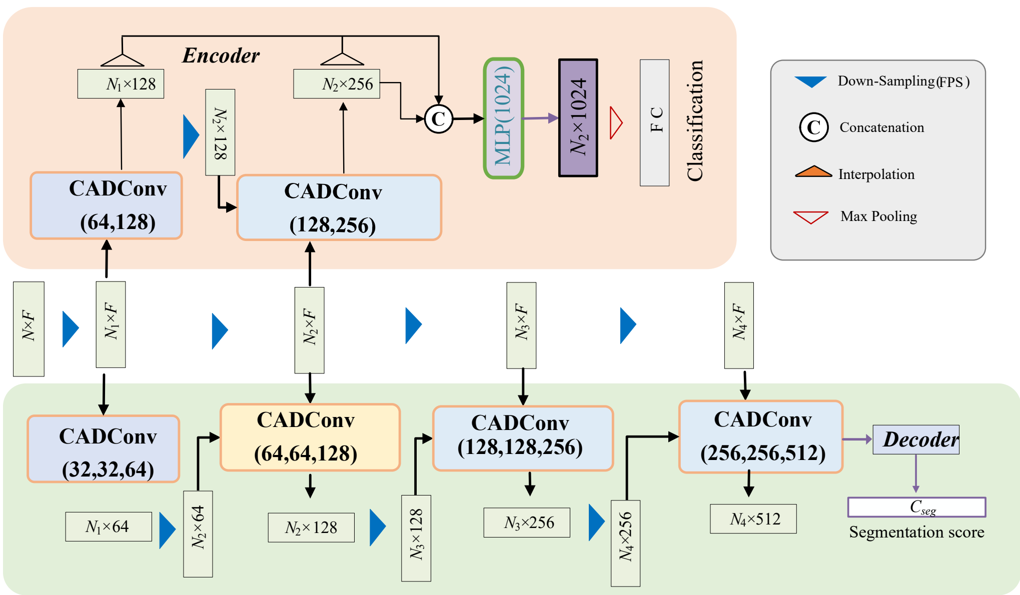

Fig. 2. The Structure of Channel-Aware Dynamic Convolutional Neural Network.

∂_k is the channel aware coefficients of the feature channels in the k-th block, and W_k is the weight parameter of the feature channels in the k-th block. According to the channel attention coefficients, the feature channels with different importance of the feature channels of point cloud are adaptively selected, which can significantly reduce the computational costs while improving the recognition performance. Specifically, for point clouds that are less difficult to recognize, CADConv adaptively selects fewer important feature channels to perform convolution operations to do the feature representation; For large-scale point clouds, CADConv dynamically allocates more important feature channels to Perform convolution to ensure the adequacy of computing resources. Finally, the output channels from different blocks are connected together to enhance the integrity of features, and RuLU activation layer is used for nonlinear mapping. The calculation formula of CADConv is as follows:

$$\begin{aligned} y = G(x_k) \otimes Pruning(W) &= G(x_k) \otimes S(\sum_{k=1}^{g} \partial_k \odot \mathrm{W}_k) \\ &= F(x_k, \Theta) \otimes A(x_k, \psi) \otimes S(\sum_{k=1}^{g} \partial_k \odot \mathrm{W}_k) \end{aligned} \quad (4)$$

In summary, CADConv can reduce computational costs while preserving high recognition capabilities.

C. Channel-Aware Dynamic Convolutional Neural Network

We use the lightweight CADConv as the basic feature representation operation, and further build a lightweight CADCNN for 3D model recognition and large-scale point cloud segmentation, as shown in Figure 2. For the classification task, we design a neural network with a depth of three layers. Finally, the max pooling is used to aggregation features, and three-fully connected layers are used to obtain the classification score. For the semantic segmentation task, each encoder layer is similar to the settings in the classification, but the network has a deeper structure. Since the semantic segmentation of 3D point clouds has a larger-scale point cloud, we sample four times in total to reduce the computation. Similar to PointNet++ [18], we use the inverse distance weighted average of k nearest neighbors for interpolation. In addition, we introduce skip connections for feature propagation in the middle layers of the encoder and decoder.

III. Experiments

A. Datasets

For the 3D point cloud recognition task, we choose ModelNet40 [11] standard dataset for the experiments. ModelNet40

TABLE I
RECOGNITION ON MODELNET40

Methods	mA(%)	OA(%)	params (Million)	FLOPs(Million)
VoxNet [12]	83.0	85.9	-	-
PointNet [17]	86.0	89.2	3.5	919
PointNet++ [18]	-	90.7	2.0	3136
KC-Net [20]	-	91.0	-	-
Kd-Net [13]	-	91.8	-	-
DGCNN [19]	90.2	92.2	1.8	3212
PCNN [21]	-	92.3	8.2	-
SpiderCNN [22]	90.7	92.4	-	-
PointCNN [23]	88.1	92.2	0.6	1682
PointASNL [24]	-	92.9	3.4	1700
KPConv [25]	-	92.9	1.5	-
CADCNN(Ours)	91.9	93.4	1.6	2200

TABLE II
QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON INDOOR S3DIS

Methods	OA(%)	mIoU(%)	FLOPs(M)
PointNet [17]	78.5	47.6	7222
MS+CU [26]	79.2	47.8	-
G+RCU [26]	81.1	49.7	-
PointNet++ [18]	81.0	54.5	-
DGCNN [19]	84.4	56.1	12558
3P-RNN [27]	86.9	56.3	-
RSNet [28]	-	56.5	-
SPG [29]	85.5	62.1	-
LSANet [30]	86.8	62.2	-
PointCNN [23]	88.1	65.4	8038
PointWeb [31]	87.3	66.7	-
ShellNet [32]	87.1	66.8	-
HEPIN [33]	88.2	67.8	-
KPConv [25]	-	70.6	-
SSP+SPG [34]	87.9	68.4	-
RandLA-Net [35]	88.0	70.0	-
PointASNL [24]	88.8	68.7	-
CADCNN(Ours)	90.0	71.2	5368

TABLE III
QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON OUTDOOR VKITTI WITH 6-FOLD CROSS VALIDATION

Methods	OA(%)	mIoU(%)
PointNet [17]	79.7	34.4
G+RCU [26]	80.6	36.2
3P-RNN [27]	87.8	41.6
SSP+SPG [34]	84.3	52.0
CADCNN(Ours)	88.2	56.0

has a total of 12,311 CAD models in 40 categories, of which 9,943 models are used for network training and 2468 models are used for network testing. For the 3D model semantic segmentation task, the indoor scene semantic segmentation dataset S3DIS [36] and the outdoor autonomous driving scene semantic segmentation dataset vKITTI [37] are used for experiments. S3DIS is an indoor large-scale point cloud dataset, including 6 indoor areas, a total of 272 rooms, of which all points are marked as 13 semantic categories such as

TABLE IV
ABLATION STUDY ON MODELNET40

Model	Ablation	ModelNet40(OA)
A	CADCNN (W CA)	92.4%
B	CADCNN (W DC)	92.6%
C	CADCNN (W CADC)	91.1%
D	CADCNN(Fully Network)	93.4%

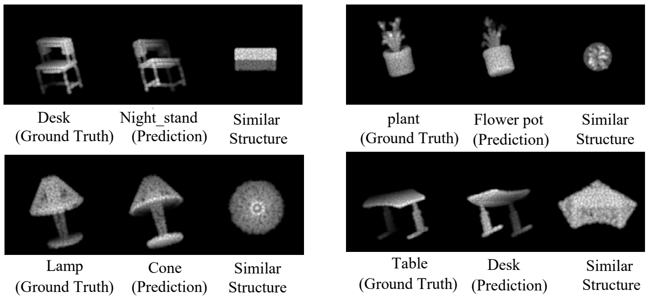

Fig. 3. Visualization of misclassification results on the ModelNet40 dataset.

board, bookcase, chair, ceiling and beam and so on. vKITTI is an outdoor large-scale point cloud dataset of actual scenes of autonomous driving, divided into 6 different urban scenes, among them, all points are marked as 13 semantic categories such as car, tree, building, road, traffic light and pedestrian in the autonomous driving scene.

B. Network Settings

The experiment uses momentum-based Stochastic Gradient Descent (SGD) optimization, the momentum factor is 0.9, the initial learning rate is 0.001, the learning rate decay index is 0.7 and the decay speed is 200,000. The Adam is used to update the step size of SGD and the network parameters are initialized with the Xavier optimizer. The initial value of the batch normalized attenuation rate is 0.5, and the final value is 0.99. The activation function uses Selu to alleviate the disappearance of the gradient and increase the nonlinear fitting ability of the network.

C. Point Cloud Classification

We select the ModelNet40 [11] dataset for the 3D point cloud recognition, and the experimental results are shown in Table I. It can be seen that the recognition accuracy of our method has reached 93.4%, which is higher than other mainstream methods, fully verifying the superiority of our method. It can also be seen from the table that our algorithm has lower parameters and FLOPs. The reason is that the channel-aware dynamic convolution we designed not only has strong feature extraction capabilities, but also has low computational and memory complexity, and can be easily embedded in other models to efficiently perform point cloud processing. Besides, to qualitatively analyze the classification results, Figure 3 shows the visualization results of several

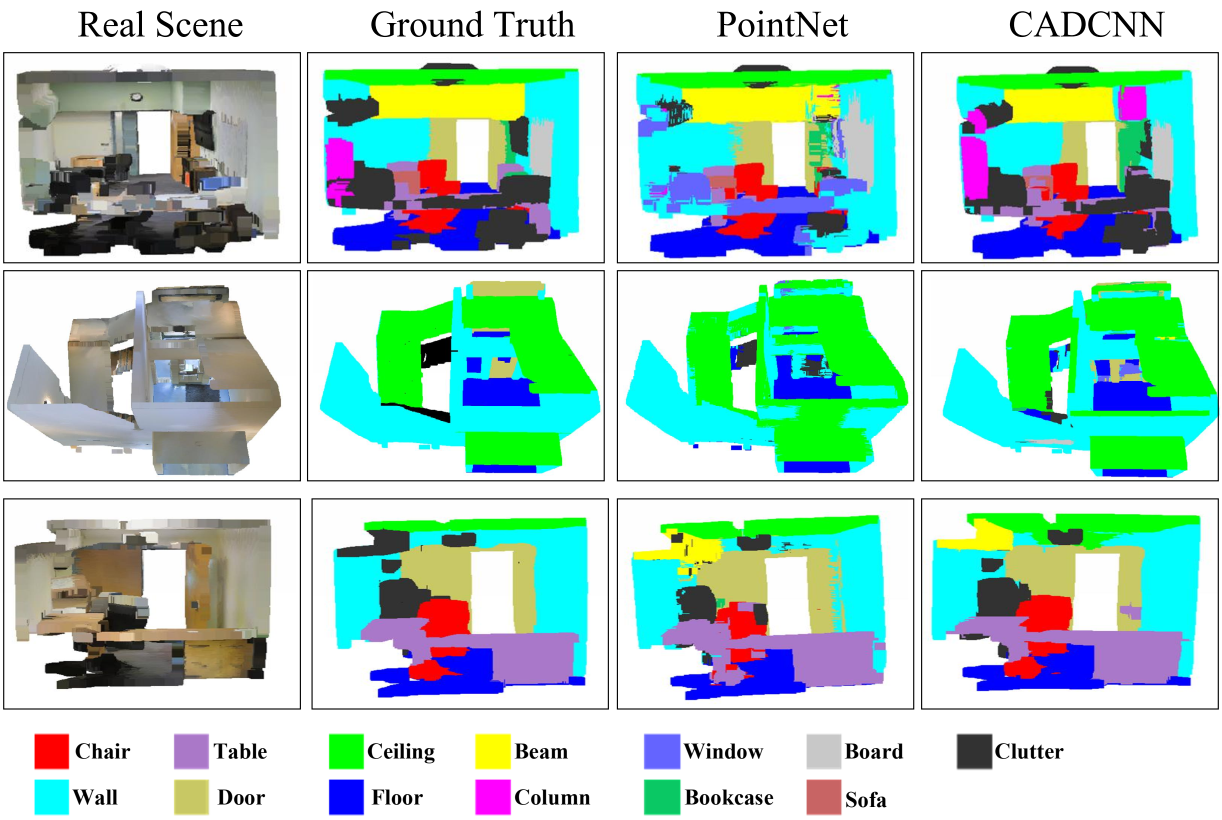

Fig. 4. Qualitative results of CADCNN on indoor S3DIS.

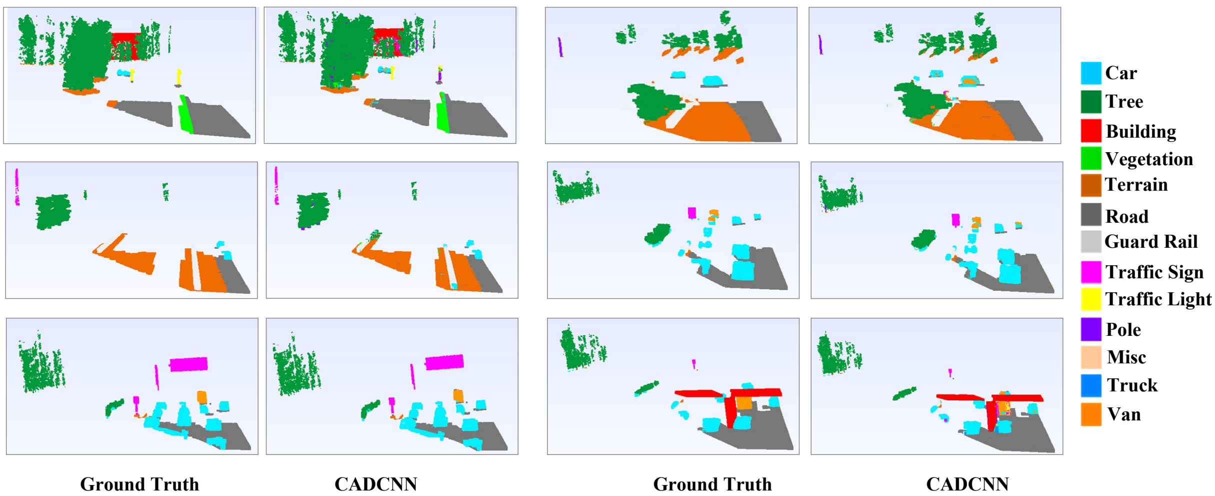

Fig. 5. Qualitative results of CADCNN on outdoor vKITTI.

typical misclassification models on the ModelNet40 dataset. It can be seen from the examples that the real categories and the predicted categories of models contain some similar local structures. These similar local structures lead to confusion in recognition of our algorithm. It shows that our method has insufficient capabilities for fine-grained local feature extraction.

D. Point Cloud Segmentation

To further verify the effectiveness of our algorithm for fine-grained shape analysis of 3D point clouds, experiments have evaluated our network performance on large-scale indoor point clouds S3DIS [36] and outdoor point clouds vKITTI [37]. The experimental results are shown in Table II and Table III. As shown, comprehensively, our algorithm has shown good recognition ability on all datasets, and the overall accuracy (OA) and mean Intersection-over-Union (mIoU) are both superior to other mainstream methods and fully prove the advanced nature of our algorithm. In addition to quantitative comparison, we also perform qualitative analysis. Figure 4 and Figure 5 show the visualization of the segmentation results of our algorithm on the S3DIS and vKITTI datasets, respectively. It can be seen that the segmentation result of our algorithm is closer to Ground Truth, can perfectly segment the entire large-scale point cloud scene, and has better robustness to missing points and occlusion. The reason is that the channel-aware dynamic convolution can adaptively block to ensure the integrity of the network structure, to improve the recognition ability of the model.

E. Ablation Study

To explore the influence of the channel-aware and full dynamic convolution designed in this paper on the point cloud recognition performance, different networks were constructed for training and testing. The experimental results are shown in Table IV. Among them, "W CA" means without channel-aware. It can be seen that the recognition accuracy is reduced by 1.0% after using random pruning instead of channel-aware. "W DC" means without dynamic convolution. After removing dynamic convolution, the recognition accuracy is reduced by 0.8%. "W CADC" means without channel-aware dynamic convolution, that is, using Multilayer Perceptron(MLP) for feature extraction, the recognition accuracy is reduced by 2.3%. The above experiments fully verify the advanced nature of channel-aware and channel-aware dynamic convolution, and fully demonstrate that the three components constructed are effective.

IV. Conclusion

We propose a channel-aware dynamic convolutional neural network for 3D model recognition and semantic segmentation. Firstly, a channel-aware strategy is designed to block the input features and the parameters of the convolution kernel adaptively, which can ensure the integrity of the network structure and improve the feature extraction ability of the model. Secondly, channel-aware dynamic convolution is used to adaptively prune unimportant feature channels, reducing redundant feature channels' computational costs. Finally, a lightweight channel-aware dynamic convolutional neural network is constructed for efficient point cloud classification and segmentation. Experiments show that our algorithm has good advantages in recognition accuracy, complexity for large scale point cloud recognition and semantic segmentation.

References

[1] Robert Osada, Thomas Funkhouser, Bernard Chazelle and David Dobkin, "Shape Distributions," ACM Transactions on Graphics, vol. 21, no. 4, pp. 807–832, 2002.
[2] Jian Sun, Maks Ovsjanikov and Leonidas Guibas, "A Concise and Provably Informative Multi-Scale Signature based on Heat Diffusion," Computer Graphics Forum, vol. 28, no. 5, pp. 1383–1392, 2009.
[3] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," Commun.ACM, vol. 60, no. 6, pp. 84–90, 2017.
[4] Yuanbo Fang, Hongliang Fu, Huawei Tao, Xia Wang and Li Zhao, "Bidirectional LSTM with Multiple Input Multiple Fusion Strategy for Speech Emotion Recognition." IAENG International Journal of Computer Science, vol. 48, no. 3, pp. 613–618, 2021.
[5] Jingxuan Yu, Kian Ming Lim and Chin Poo Lee, "MoVE-CNNs: Model Averaging Ensemble of Convolutional Neural Networks for Facial Expression Recognition," IAENG International Journal of Computer Science, vol. 48, no. 3, pp. 519–523, 2021.
[6] Iromi R Paranavithana and Viraj R Kalansuriya, "Deep Convolutional Neural Network Model for Tea Bud(s) Classification," IAENG International Journal of Computer Science, vol. 48, no. 3, pp. 599–604, 2021.
[7] Jian Liu and Weisheng Wu, "Automatic Image Annotation Using Improved Wasserstein Generative Adversarial Networks," IAENG International Journal of Computer Science, vol. 48, no. 3, pp. 507–513, 2021.
[8] Xinyu Tian, Qinghe Zheng and Nan Jiang, "An Abnormal Behavior Detection Method Leveraging Multi-modal Data Fusion and Deep Mining," IAENG International Journal of Applied Mathematics, vol. 51, no. 1, pp. 92–99, 2021.
[9] Hang Su, Subhransu Maji, Evangelos Kalogerakis and Erik Learned-Miller, "Multi-view Convolutional Neural Networks for 3d Shape Recognition," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 945–953.
[10] Simonyan, Karen and Andrew Zisserman. , "Very Deep Convolutional Networks for Large-scale Image Recognition," in arXiv preprint arXiv:1409.1556, 2014.
[11] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang and Jianxiong Xiao, "3d Shapenets: A Deep Representation for Volumetric Shapes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, pp. 1912–1920.
[12] Maturana Daniel and Sebastian Scherer, "VoxNet: A 3d Convolutional Neural Network for Real-time Object Recognition," in IEEE/RSJ International Conference on Intelligent Robots and Systems 2015, pp. 922–928.
[13] Klokov Roman and Victor Lempitsky, "Escape from Cells: Deep kd-networks for the Recognition of 3d point Cloud Models," in Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 863–872.
[14] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei and Ali Farhadi, "Target-driven Visual Navigation in Indoor Scenes Using Deep Reinforcement Learning," in IEEE International Conference on Robotics and Automation 2017, pp. 3357–3364.
[15] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su and Leonidas J. Guibas, "Frustum Pointnets for 3d Object Detection from rgb-d Data," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 918–927.
[16] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha and MichaelBeetz, "Towards 3D Point Cloud based Object Maps for Household Environments," Robotics and Autonomous Systems, vol. 56, no. 11, pp. 927–941, 2008.
[17] Charles R. Qi, Hao Su, Kaichun Mo and Leonidas J. Guibas, "Pointnet: Deep Learning on Point Sets for 3d Classification and Segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 652–660.
[18] Charles Ruizhongtai Qi, Li Yi, Hao Su and Leonidas J. Guibas, "Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," arXiv preprint arXiv:1706.02413, 2017.
[19] 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, vol. 38, no. 5, pp. 1–12, 2019.
[20] Yiru Shen, Chen Feng, Yaoqing Yang and Dong Tian, "Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 4548–4557.
[21] Matan Atzmon, Haggai Maron and Yaron Lipman, "Point Convolutional Neural Networks by Extension Operators," arXiv preprint arXiv:1803.10091, 2018.
[22] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng and Yu Qiao, "Spider-cnn: Deep Learning on Point Sets with Parameterized Convolutional Filters," in Proceedings of the European Conference on Computer Vision 2018, pp. 87–102.
[23] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di and Baoquan Chen, "Pointcnn: Convolution on x-transformed points," Advances in Neural Information Processing Systems, vol. 31, pp. 828–838, 2018.
[24] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang and Shuguang Cui, "Pointasnl: Robust Point Clouds Processing Using Nonlocal Neural Networks with Adaptive Sampling," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 5589–5598.
[25] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette and Leonidas J. Guibas, "Kpconv: Flexible and Deformable Convolution for Point Clouds," in Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, pp. 6411–6420.
[26] Francis Engelmann, Theodora Kontogianni, Alexander Hermans and Bastian Leibe, "Exploring Spatial Context for 3d Semantic Segmentation of Point Clouds," in Proceedings of the IEEE International Conference on Computer Vision Workshops 2017, pp. 716–724.
[27] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du and Xiaolin Zhang, "3d Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation," in Proceedings of the European Conference on Computer Vision 2018, pp. 403–417.

[28] Qiangui Huang, Weiyue Wang and Ulrich Neumann, “Recurrent Slice Networks for 3d Segmentation of Point Clouds,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 2626–2635.
[29] Loic Landrieu and Martin Simonovsky, “Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 4558–4567.
[30] Linzhuo Chen, Xuanyi Li, Dengping Fan, Kai Wang, Shaoping Lu and Mingming Cheng, “LSANet: Feature Learning on Point Sets by Local Spatial Aware Layer,” arXiv preprint arXiv:1905.05442, 2019.
[31] Hengshuang Zhao, Li Jiang, Chi-Wing Fu and Jiaya Jia, “Pointweb: Enhancing Local Neighborhood Features for Point Cloud Processing,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, pp. 5565–5573.
[32] Zhiyuan Zhang, Binh-Son Hua and Sai-Kit Yeung, “Shellnet: Efficient Point Cloud Convolutional Neural Networks Using Concentric Shells Statistics,” in Proceedings of the IEEE International Conference on Computer Vision 2019, pp. 1607–1616.
[33] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu and Jiaya Jia, “Hierarchical Point-edge Interaction Network for Point Cloud Semantic Segmentation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, pp. 10 433–10 441.
[34] Loic Landrieu and Mohamed Boussaha, “Point Cloud Oversegmentation with Graph-structured Deep Metric Learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, pp. 7440–7449.
[35] 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/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 11 108–11 117.
[36] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer and Silvio Savarese, “3d Semantic Parsing of Large-scale Indoor Spaces,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 1534–1543.
[37] Francis Engelmann, Theodora Kontogianni, Alexander Hermans and Bastian Leibe, “Exploring Spatial Context for 3d Semantic Segmentation of Point Clouds,” in Proceedings of the IEEE International Conference on Computer Vision Workshops 2017, pp. 716–724.