PointListNet: Deep Learning on 3D Point Lists

SUPPLEMENTARY MATERIAL

A. Comparison with 4D Point Cloud Methods

The residue-level 3D structure of proteins can be seen as 3D point lists. If the sequence structure is neglected, the amino acids of a protein form a point cloud. Therefore, in Table 1 of the main paper, we compare our method with those 3D point cloud methods (PointNet++ [5], DGCNN [7], Point Transformer [9] and PointMLP [4]) for protein recognition.

In this section, we extend those 3D point cloud methods to 4D. Specifically, for PointNet++ and Point Transformer, we integrate the 1D order displacement into modeling, as shown in Table 1. For DGCNN and PointMLP, we follow the original paper to concatenate point order coordinates and features as the input of the network. We conduct the experiments by ourselves.

Table 1. Comparison with 3D and 4D point cloud methods, *i.e.*, PointNet++ [5], DGCNN [7], Point Transformer [9] and PointMLP [4], on protein fold classification and enzyme catalytic reaction classification (accuracy %). Experiments are conducted by ourselves.

Method	Modeling	Protein Fold Classification			Enzyme Reaction
		Fold	Superfamily	Family	Classification
3D PointNet++ [5]	$\boldsymbol{f}'_t = \underset{\|\boldsymbol{p}_{t'}-\boldsymbol{p}_t\|\le r}{\mathrm{MAX}} \mathrm{MLP}\big([\boldsymbol{f}_{t'}, \boldsymbol{p}_{t'}-\boldsymbol{p}_t]\big)$	26.0	37.7	93.8	78.4
3D DGCNN [7]	$\boldsymbol{f}'_t = \underset{\boldsymbol{f}_{t'}\in \mathrm{TopK}(\boldsymbol{f}_t)}{\mathrm{MAX}} \mathrm{MLP}\big([\boldsymbol{f}_{t'}, \boldsymbol{f}_{t'}-\boldsymbol{f}_t]\big)$	25.6	39.2	94.4	80.1
3D Point Transformer [9]	$\boldsymbol{f}'_t = \sum_{\|\boldsymbol{p}_{t'}-\boldsymbol{p}_t\|\le r} \alpha_{tt'} \times (\boldsymbol{W}_3 \cdot \boldsymbol{f}_{t'} + \delta_{tt'})$, $\alpha_{tt'} = \mathrm{softmax}\big(\mathrm{MLP}(\boldsymbol{W}_1 \cdot \boldsymbol{f}_t - \boldsymbol{W}_2 \cdot \boldsymbol{f}_{t'} + \delta_{tt'})\big)$, $\delta_{tt'} = \mathrm{MLP}(\boldsymbol{p}_t - \boldsymbol{p}_{t'})$,	26.4	40.1	92.0	81.3
3D PointMLP [4]	$\boldsymbol{f}'_t = \mathrm{MLP}\big(\underset{\boldsymbol{f}_{t'}\in \mathrm{TopK}(\boldsymbol{f}_t)}{\mathrm{MAX}} \mathrm{MLP}(\boldsymbol{f}_{tt'})\big)$ $\boldsymbol{f}_{tt'} = \alpha \odot \frac{\boldsymbol{f}_{t'}-\boldsymbol{f}_t}{\delta+\epsilon} + \beta$ $\delta = \sqrt{\frac{1}{K\times N\times C}\sum_{i=1}^{N}\sum_{j=1}^{K}(\boldsymbol{f}_{t'}-\boldsymbol{f}_t)^2}$	26.8	38.8	94.2	79.7
PointListNet (ours)	3D Coordinate	36.8	55.3	97.4	84.5
4D PointNet++ [5]	$\boldsymbol{f}'_t = \underset{\|\boldsymbol{p}_{t'}-\boldsymbol{p}_t\|\le r}{\mathrm{MAX}} \mathrm{MLP}\big([\boldsymbol{f}_{t'}, \boldsymbol{p}_{t'}-\boldsymbol{p}_t, t'-t]\big)$	27.3	38.8	94.0	80.0
4D DGCNN [7]	$\boldsymbol{f}'_t = \underset{\boldsymbol{f}_{t'}\in \mathrm{TopK}(\boldsymbol{f}_t)}{\mathrm{MAX}} \mathrm{MLP}\big([\boldsymbol{f}_{t'}, \boldsymbol{f}_{t'}-\boldsymbol{f}_t]\big)$	25.9	39.7	94.8	80.4
4D Point Transformer [9]	$\boldsymbol{f}'_t = \sum_{\|\boldsymbol{p}_{t'}-\boldsymbol{p}_t\|\le r} \alpha_{tt'} \times (\boldsymbol{W}_3 \cdot \boldsymbol{f}_{t'} + \delta_{tt'})$, $\alpha_{tt'} = \mathrm{softmax}\big(\mathrm{MLP}(\boldsymbol{W}_1 \cdot \boldsymbol{f}_t - \boldsymbol{W}_2 \cdot \boldsymbol{f}_{t'} + \delta_{tt'})\big)$, $\delta_{tt'} = \mathrm{MLP}(\boldsymbol{p}_t - \boldsymbol{p}_{t'}, t-t')$.	27.7	40.3	93.2	81.9
4D PointMLP [4]	$\boldsymbol{f}'_t = \mathrm{MLP}\big(\underset{\boldsymbol{f}_{t'}\in \mathrm{TopK}(\boldsymbol{f}_t)}{\mathrm{MAX}} \mathrm{MLP}(\boldsymbol{f}_{tt'})\big)$ $\boldsymbol{f}_{tt'} = \alpha \odot \frac{\boldsymbol{f}_{t'}-\boldsymbol{f}_t}{\delta+\epsilon} + \beta$ $\delta = \sqrt{\frac{1}{K\times N\times C}\sum_{i=1}^{N}\sum_{j=1}^{K}(\boldsymbol{f}_{t'}-\boldsymbol{f}_t)^2}$	28.1	41.1	94.3	82.2
PointListNet (ours)	3D Coordinate & 1D Order	**55.2**	**76.4**	**99.5**	**88.0**

The modeling approaches of these point cloud methods and the experimental results are shown in Table 1. Compared to the original 3D methods, these extended 4D variants do not significantly improve the accuracy. This may be because they do

not model the 1D sequence structure in a regular manner.

B. Gene Ontology Term Prediction

Gene ontology term prediction is a multi-label classification problem. The goal of gene ontology term prediction aims at predicting the functions of a protein via multiple gene ontology terms. This task gene ontology term prediction includes three sub-tasks: biological process, molecular function and cellular component ontology term prediction. Following [1], biological process, molecular function and cellular component include 1,943 classes, 489 classes and and 320 classes, respectively. The evaluation dataset contains 29,898/3,322/3,415 proteins for training/validation/test, respectively. As shown in Table 2, the proposed methods outperforms existing methods.

Table 2. Accuracy (F_{max}) of gene ontology term prediction.

Method	Gene Ontology		
	Biological Process	Molecular Function	Cellular Component
IEConv (Protein) [2]	0.421	0.624	0.431
GearNet [8]	0.356	0.503	0.414
GearNet-IEConv [8]	0.381	0.563	0.422
GearNet-Edge [8]	0.403	0.580	0.450
GearNet-Edge-IEConv [8]	0.400	0.581	0.430
PointListNet (ours)	**0.439**	**0.643**	**0.486**

C. Comparison with More Point Cloud Methods

In this section, we investigate two more point cloud methods, *i.e.*, Stratified Transformer [3] and PointNeXt [6], for protein recognition. Stratified Transformer employs a UNet architecture, which includes a downsampling part and an upsampling part, for point cloud segmentation. To apply Stratified Transformer to protein classification, we use the downsampling part. PointNeXt is an improved version of PointNet++. For PointNeXt, we use its large model, *i.e.*, PointNeXt-L.

Table 3. Comparison with Stratified Transformer [3] and PointNeXt [6] on protein fold classification and enzyme catalytic reaction classification (accuracy %). Experiments are conducted by ourselves.

Method	Protein Fold Classification			Enzyme Reaction
	Fold	Superfamily	Family	Classification
Stratified Transformer [3]	27.1	40.8	95.3	82.8
PointNeXt-L [6]	30.5	42.7	96.4	84.0
PointListNet (ours)	**55.2**	**76.4**	**99.5**	**88.0**

Results are shown in Table 3. Our method significantly outperform Stratified Transformer and PointNeXt-L.

D. Comparison with Individual 1D Order Network and 3D Coordinate Network

To model the geometry-sequence structure in proteins, one can also employ an independent 1D order network and an independent 3D coordinate network for sequence and geometry modeling, respectively. Specifically, we can consider the 1D and 3D networks as two experts and employ the mixture technique of expert ensembles. In this section, we split our

PointListNet into two independent networks. The first one only models the sequence structure and the other one only models the geometry structure. Then, the outputs of the two networks are aggregated for protein recognition.

As shown in Table 4, the combination of 1D order network & 3D coordinate network does not significantly improve the accuracy. Because the two structures are processed separately, it may not properly understand proteins' local geometry-sequence structure.

Table 4. Combination of 1D order network and 3D coordinate network on protein fold classification and enzyme catalytic reaction classification (accuracy %).

Method	Fold Classification			Enzyme
	Fold	Superfamily	Family	Reaction
1D Order Network	13.1	18.7	86.4	70.0
3D Coordinate Network	36.8	55.3	97.4	84.5
1D Order Network & 3D Coordinate Network	41.4	62.0	98.4	85.4
PointListNet	**55.2**	**76.4**	**99.5**	**88.0**

References

[1] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):1–14, 2021. 2

[2] Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. *arXiv*, 2205.15675, 2022. 2

[3] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *CVPR*, 2022. 2

[4] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *ICLR*, 2022. 1

[5] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 1

[6] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *arXiv*, 2206.04670, 2022. 2

[7] 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 Trans. Graph.*, 38(5):146:1–146:12, 2019. 1

[8] Zuobai Zhang, Minghao Xu, Arian R. Jamasb, Vijil Chenthamarakshan, Aurélie C. Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv*, 2203.06125, 2022. 2

[9] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 1