Supplementary Material for "Long Range Pooling for 3D Large-Scale Scene Understanding"

This document presents supplementary investigations into the generality and applicability of the LRP module, as well as the impacts of non-linear functions on feature extraction as an alternative to feature aggregation via MaxPool.

1. Generality and Applicability of the LRP Module

To verify the generality and applicability of the LRP module, we conducted experiments on different networks and tasks.

Firstly, we explored the generality and applicability of the LRP module by conducting experiments on another 3D understanding architecture, MinkowskiNet [1]. We added the LRP module after each stage of MinkowskiNet, without adjusting hyperparameters, and still achieved a 0.5 mIoU improvement as shown in Table 1. Additionally, we observed that the incorporation of the LRP module resulted in a low increase in network parameters and time consumption.

Secondly, we investigated the effect of the LRP module for point cloud classification. We evaluated both voxel-based and point-based models. For voxel-based methods, we replaced the decoders of Baseline and LRPNet with fully connected layers for point classification. For point-based methods, we used the popular point-based method, PointNet++ [3], and added the LRP module after each set abstraction layer. Table 2 demonstrates that the LRP module consistently improved point classification performance for both voxel-based and point-based methods.

Thirdly, we investigated the impact of the LRP module on image classification by performing experiments on Cifar10/100. We added the LRP module after each stage of ResNet [2] without adjusting hyperparameters. The results in Table 3 demonstrate that the LRP module can still improve image classification performance without tuning hyperparameters, while large kernel convolution only resulted in a slight improvement on Cifar10 and decreased performance on Cifar100 under the same conditions. Although recent studies on image classification have shown the effectiveness of large kernel convolution, it requires careful tuning of hyperparameters and locations to work well. In contrast, our LRP module is more adaptable and generally applicable.

Method	Params(M)	Runtime(ms)	mIoU
MinkowskiNet	37.9	120.1	73.6
MinkowskiNet+LRP	38.5	157.0	**74.1**

Table 1. mIoU (%) scores for MinkowskiNet on ScanNet val set.

Method	Acc(%)	Method	Acc(%)
Baseline	91.2	PointNet++	92.2
LRPNet	**91.8**	PointNet++(LRP)	**92.6**

Table 2. Overall classification accuracy on ModelNet40.

	Cifar10				Cifar100			
Method	Params(M)	Flops(G)	Top-1	Top-5	Params(M)	Flops(G)	Top-1	Top-5
R18	11.17	0.56	94.82	**99.87**	11.22	0.56	78.80	93.91
R18*	12.25	0.61	95.04	99.82	12.30	0.61	78.14	94.13
R18+	12.22	0.61	**95.19**	99.85	12.27	0.61	**79.15**	**94.56**
R34	21.28	1.16	95.15	99.76	21.33	1.16	79.25	94.59
R34*	22.36	1.22	95.24	99.85	22.41	1.22	78.60	93.90
R34+	22.33	1.21	**95.65**	**99.86**	22.38	1.21	**79.49**	**94.78**

Table 3. Classification accuracy on Cifar10 and Cifar100. * means large kernel conv. + denotes our large range pooling.

Furthermore, the consistent improvements observed in Table 1, 2, and 3 confirm the effectiveness of the LRP module in improving performance across various backbone architectures and different fields and tasks. The versatility of the LRP module allows it to be easily incorporated into existing models without the need for significant adjustments or tuning of hyperparameters.

2. Further Exploration of Nonlinear Functions

As noted in our paper, incorporating non-linearity in feature aggregation can lead to improved network performance. While it can be challenging to quantify the degree of non-linearity, it is widely accepted that nonlinear opera-

Function	+AvgPool mIoU(%)	+Conv mIoU(%)
Baseline	73.9	73.8
+leaky relu	73.5	74.1
+relu	73.5	73.2
+elu	73.4	73.5
+gelu	73.4	72.9
+abs	73.4	73.7
+tanh	72.4	72.6
+sigmoid	71.7	72.1
+MaxPool (Ours)	75.0	

Table 4. Comparison of non-linearity functions with linear aggregation on the ScanNet val set. **Function**: the non-linearity functions. **AvgPool**: dilation average pooling. **Conv**: dilation convolution. **MaxPool**: dilation max pooling.

tors provide greater non-linearity than linear operators, such as average or convolution. Currently, we employ the max operator as a nonlinear operator in feature aggregation because other nonlinear functions, such as abs or tanh, are not well-suited for this task.

However, we still wish to investigate the impact of other nonlinear functions on network performance. To this end, we have tested common nonlinear functions followed by a convolution or an average pooling for feature aggregation on the ScanNet validation set. The results presented in Table 4 indicate that employing non-linearity functions followed by a convolution or an average pooling does not enhance network performance. This is to be expected, as this approach simply adds a nonlinear layer to the previous feature extraction module, which is similar to inserting two activation functions after convolution and then aggregating the results in a linear manner.

References

[1] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[3] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1