Supplementary Material for "POEM: 1-bit Point-wise Operations based on Expectation-Maximization for Efficient Point Cloud Processing"

1 Algorithm for POEM

Algorithm 1 POEM training. L is the loss function (summation of L_S and L_R) and N is the number of layers. Binarize() binarizes the filters, and Update() updates the parameters based on our update scheme.

Input: a minibatch of inputs and their labels, unbinarized weights $\mathbf{w}$, scale factor $\boldsymbol{\alpha}$, learning rates η.

Output: updated unbinarized weights $\mathbf{w}^{t+1}$, updated scale factor $\boldsymbol{\alpha}^{t+1}$.

1: {1. Computing gradients with aspect to the parameters:}
2: {1.1. Forward propagation:} i =1 to N
3: $\mathbf{b}^{\mathbf{w}_i} \leftarrow$ Binarize($\mathbf{w}_i$)
4: Bi-FC features calculation using Eq. 2
5: Loss calculation using Eq. 3 - 4
6: {1.2. Backward propagation:} i =N to 1
7: {Note that the gradients are not binary.}
8: Computing $\delta_{\mathbf{w}}$ using Eq. 6 - 15
9: Computing $\delta_{\boldsymbol{\alpha}}$ using Eq. 16 - 17
10: {Accumulating the parameters gradients:} i = 1 to N
11: $\mathbf{w}^{t+1} \leftarrow$ Update($\delta_{\mathbf{w}}$, η)
12: $\boldsymbol{\alpha}^{t+1} \leftarrow$ Update($\delta_{\boldsymbol{\alpha}}$, η)
13: $\eta^{t+1} \leftarrow$ Update(η) according to learning rate schedule

2 Efficiency Analysis

Memory usage is analyzed by comparing our approach with the corresponding real-valued network. The BOPs denotes the number of 1-bit operations in networks. For real-valued networks, all the operations are counted as FLOPs. For 1-bit networks, all fully-connected

Model	Method	W/A (bit)	FLOPs ($\times 10^6$)	BOPs ($\times 10^6$)	Computation ($\times 10^6$)	FLParams ($\times 10^6$)	BParams ($\times 10^6$)	Memory Usage ($\times 10^6$)
PointNet	Real-valued	32/32	3.47	0	222.08	3.47	0	111.04
	POEM	1/1	0.06	3.45	7.29	0.06	3.45	5.07
PointNet++	Real-valued	32/32	1.75	0	111.84	1.75	0	55.92
	POEM	1/1	0.04	1.73	4.29	0.04	1.73	3.01
DGCNN	Real-valued	32/32	1.81	0	115.84	1.81	0	57.92
	POEM	1/1	0.02	1.80	2.73	0.02	1.80	2.44

Table 1: Efficiency Analysis on the three models we employed. We report FLOPs, BOPs, Computation, FLParams, BParams and Memory Uasge. We analyze the architecture used with classification task in this table.

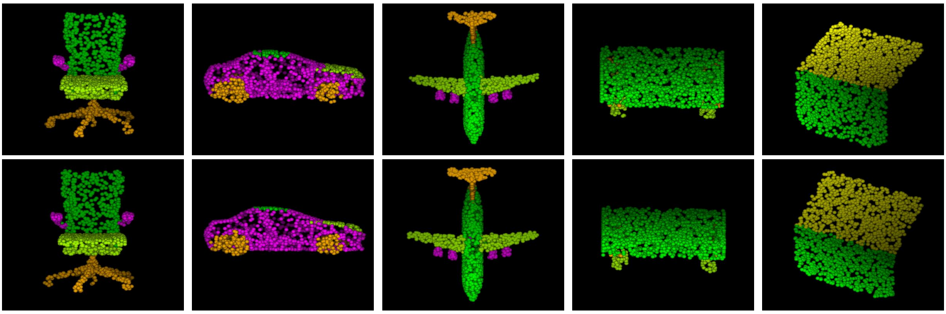

Figure 1: PointNet *vs.* 1-bit PointNet (POEM). We visualize the part segmentation results on ShapeNet Parts dataset. From left to right, we show the qualitative results of chair, car, airplane, table and laptop classes. The first row lie the results predicted by PointNet and the second row lie the results predicted by 1-bit PointNet (POEM) respectively. (Best viewed in color.)

layers are counted as BOPs, and the left ones (bias, PReLU, BN, and scale factor) are counted as FLOPs. The same record process is employed for the number of parameters. In Table 1, FLParam denotes the number of real-values parameters, and BParam denotes the number of 1-bit parameters. The computation is computed as the summation of 64-bit times the number of real-valued weight and 1-bit time the number of the 1-bit weight in the networks, following the guideline of [1]. The memory usage is computed as the summation of 32-bit times the number of real-valued weights and 1-bit times the number of the 1-bit weight in the networks. As shown in Table 1, our proposed POEM reduces the computation by $\frac{222.08}{7.29} = 30.5\times$ compared with the real-valued one on PointNet backbone. Also, memory usage is reduced by $21.9\times$ compared with the full-precision one. Similar efficiency improvement is realized on PointNet++ and DGCNN backbone.

3 Visualization

We visualize the part segmentation results on ShapeNet Parts dataset in Figure 1. The PointNet backbone and our 1-bit PointNet backbone are used for comparison. From left to right, we show the qualitative results on various classes. Figure 1 demonstrates our POEM achieves comparable results as real-valued counterparts (PointNet), which is consistent with the mIOU (81.1 *vs.* 83.7) in the Table 4 of our paper.

References

[1] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of European Conference on Computer Vision*, pages 722–737, 2018.