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

Sheng Xu†[1]
shengxu@buaa.edu.cn
Yanjing Li†[1]
yanjingli@buaa.edu.cn
Junhe Zhao[1]
jhzhao@buaa.edu.cn
Baochang Zhang*[1]
bczhang@buaa.edu.cn
Guodong Guo[2]
guoguodong01@baidu.com

[1] Beihang University,
Beijing, China

[2] National Engineering Laboratory for Deep Learning Technology and Application,
Institute of Deep Learning,
Baidu Research,
Beijing, China

Abstract

Real-time point cloud processing is fundamental for lots of computer vision tasks, while still challenged by the computational problem on resource-limited edge devices. To address this issue, we implement XNOR-Net-based binary neural networks (BNNs) for an efficient point cloud processing, but its performance is severely suffered due to two main drawbacks, Gaussian-distributed weights and non-learnable scale factor. In this paper, we introduce point-wise operations based on Expectation-Maximization (POEM) into BNNs for efficient point cloud processing. The EM algorithm can efficiently constrain weights for a robust bi-modal distribution. We lead a well-designed reconstruction loss to calculate learnable scale factors to enhance the representation capacity of 1-bit fully-connected (Bi-FC) layers. Extensive experiments demonstrate that our POEM surpasses existing the state-of-the-art binary point cloud networks by a significant margin, up to 6.7%.

1 Introduction

Compared with traditional 2D images, 3D data provides an opportunity to understand the surrounding environment for machines better. With the advancement of deep neural networks (DNNs) directly processing raw point clouds, great success has been achieved in PointNet [14], PointNet++ [15] and DGCNN [18]. However, existing methods are inefficient for real applications that require real-time inference and fast response, such as autonomous driving and augmented reality. Their deployed environments are often resource-constrained edge devices. To address the challenge, Grid-GCN [21], RandLA-Net [8], and PointVoxel [11], have been introduced for efficient point cloud processing using DNNs.

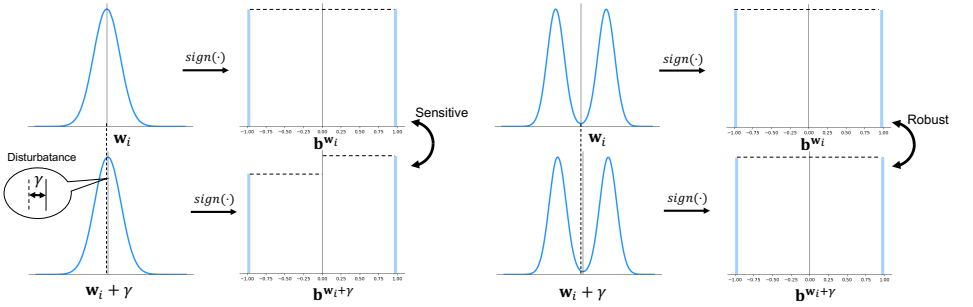


Figure 1: Subfigure (a) and (b) illustrates the robustness of Gaussian distribution and bi-modal distribution. From left to right in each subfigure, we plot the distribution of the unbinarized weights $\mathbf{w}_i$ and the binarized weights $\mathbf{b}^{\mathbf{w}_i}$.

While significant speedup and memory footprint reduction have been achieved, these works still rely on expensive floating-point operations, leaving room for further optimization of the performance from the model quantization perspective. Binarized neural network (BNNs) [3, 4, 5, 9, 10, 17, 19, 22, 23] compress weights and activations of DNNs into a single bit, which can decrease the storage requirements by 32× and computation cost by up to 58× [17]. However, using network binarization for point cloud processing tasks remains largely unexplored.

In this paper, we first implement a baseline, XNOR-Net-based [17] 1-bit point cloud network, which shows that the performance drop is mainly caused by two drawbacks. First, layer-wise weights of XNOR-Net roughly follow a Gaussian distribution with a mean value around 0. However, such a distribution is subjected to the disturbance aroused by the noise containing in the raw point cloud data [7]. As a result, the Gaussian-distributed weight (around 0) will accordingly change its sign, *i.e.*, the binarization result will be changed dramatically. This explains why the baseline network is ineffective to process the point cloud data and achieves a worse convergence, as shown in Figure 1 (a). In contrast, bi-modal distribution will gain more robustness against the noise. Second, XNOR-Net fails to adapt itself to the characteristics of cloud data, when computing the scale factor using a non-learning method.

To address these issues, we introduce 1-bit point-wise operations based on Expectation-Maximization (POEM) to efficiently process the point cloud data. First, we exploit Expectation-Maximization (EM) [12] to constrain the weights into a bi-modal distribution, which can be more robust to disturbances caused by the noise containing in the raw point cloud data [7], as shown in Figure 1 (b). We also introduce a learnable and adaptive scale factor for every 1-bit layer to enhance the feature representation capacity of our binarized networks. Finally, we lead a powerful 1-bit network for point cloud processing, which can well reconstruct real-valued counterparts' amplitude via a new learning-based method. Our contributions are summarized as follows:

- We introduce a new binarization approach of point-wise operations based on Expectation-Maximization (POEM), which can efficiently binarize network weights and activations for point cloud processing.

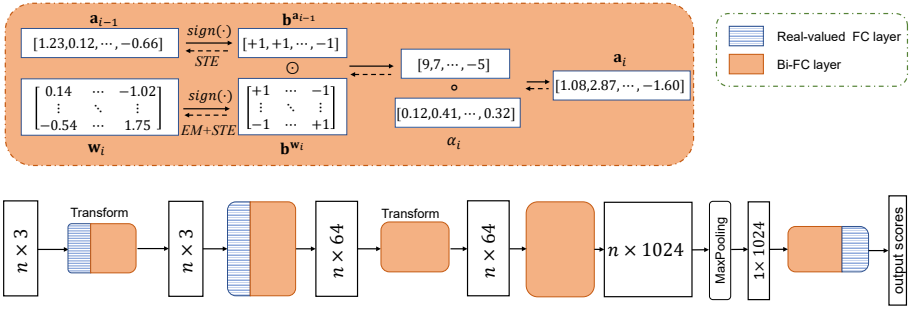


Figure 2: Outline of the 1-bit PointNet obtained by our POEM on the classification task. EM denotes Expectation-Maximization algorithm, and STE denotes Straight-Through-Estimator.

- We achieve a learnable scale factor to modulate the amplitude of real-valued weights in an end-to-end manner, which can significantly improve the representation ability of binarized networks.
- Our methods are generic and can be readily extendable to mainstream point cloud feature extractors. Extensive experiments on multiple fundamental point cloud tasks demonstrate the superiority of our POEM. For example, the 1-bit PointNet mounted by our POEM achieves 90.2% overall accuracy on the ModelNet40 dataset, which is even 0.9% higher than real-valued counterparts and promotes the state-of-the-arts.

2 Learning networks via POEM

This section elaborates our proposed POEM method, including the binarization framework, the supervision for learning a scale factor, and the optimization towards robust weights distribution through the EM method.

2.1 Binarization Framework of POEM

Our POEM framework is shown in Figure 2. We extend the binarization process from 2D convolution (XNOR-Net) to fully-connected layers (FCs) for feature extraction, termed 1-bit fully-connected (Bi-FC) layers, based on extremely efficient bit-wise operations (XNOR and Bit-count) via the lightweight binary weight and activation.

Given a conventional FC layer, we denote $\mathbf{w}_i \in \mathbb{R}_{C_i \times C_{i-1}}$ and $\mathbf{a}_i \in \mathbb{R}_{C_i}$ as its weights and features in the i-th layer. C_i represents the number of output channels of i-th layer. We then have $\mathbf{a}_i = \mathbf{a}_{i-1} \otimes \mathbf{w}_i$, where $\otimes$ denotes full-precision multiplication. As mentioned above, the BNN model aims to binarize $\mathbf{w}_i$ and $\mathbf{a}_i$ into $\mathbf{b}^{\mathbf{w}_i} \in \mathbb{B}_{m_i}$ and $\mathbf{b}^{\mathbf{a}_i} \in \mathbb{B}_{C_i}$ in this paper respectively, where $\mathbb{B}$ denotes discrete set $\{-1,+1\}$ for simplicity. Then, we apply XNOR and Bit-count operations to replace full-precision operations. Following [17], the forward process of the BNN is defined as

$$\mathbf{a}_i = \mathbf{b}^{\mathbf{a}_{i-1}} \odot \mathbf{b}^{\mathbf{w}_i}, \tag{1}$$

where $\odot$ represents efficient XNOR and Bit-count operations. Based on XNOR-Net [17], we introduce a learnable channel-wise scale factor to modulate the amplitude of real-valued

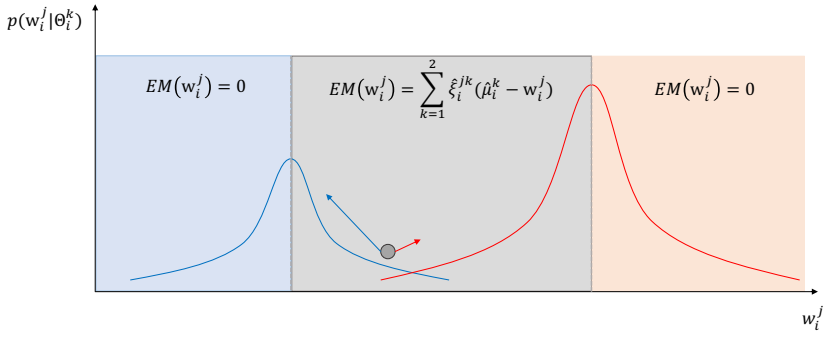


Figure 3: Illustration of training w_i^j via Expectation-Maximization. For the ones in the grey area (distribution not transparent), we apply $EM(\cdot)$ to constrain it to converge to a specific distribution.

convolution. Considering the Batch Normalization (BN) and activation layers, the forward process is formulated as

$$\mathbf{b}^{\mathbf{a}_i} = sign(\Phi(\alpha_i \circ \mathbf{b}^{\mathbf{a}_{i-1}} \odot \mathbf{b}^{\mathbf{w}_i})), \tag{2}$$

where we divide the data flow in POEM into units for detailed discussions. In POEM, the original output feature $\mathbf{a}_i$ is first scaled by a channel-wise scale factor (vector) $\alpha_i \in \mathbb{R}_{C_i}$ to modulate the amplitude of full-precision counterparts. It then enters $\Phi(\cdot)$, which represents a composite function built by stacking several layers, *e.g.*, BN layer, non-linear activation layer, and max-pooling layer. And then, the output is binarized to obtain the binary activations $\mathbf{b}^{\mathbf{a}_i} \in \mathbb{B}_{C_i}$, via sign function. $\text{sign}(\cdot)$ denotes the sign function which returns $+1$ if the input is greater than zeros, and -1 otherwise. Then, the 1-bit activation $\mathbf{b}^{\mathbf{a}_i}$ can be used for the efficient XNOR and Bit-count of $(i{+}1)$-th layer.

2.2 Supervision for POEM

To constrain the Bi-FC to have binarized weights with similar amplitudes as the real-valued counterparts, we introduce a new loss function in our supervision for POEM. We consider that unbinarized weights should be reconstructed based on binarized weights, and define the reconstruction loss as

$$L_R = \frac{1}{2}\|\mathbf{w}_i - \alpha_i \circ \mathbf{b}^{\mathbf{w}_i}\|_2^2, \tag{3}$$

where L_R is the reconstruction loss. Considering the impact of α_i on the output of the layer, we define the learning objective of our POEM as

$$\underset{\{\mathbf{w}_i,\alpha_i\},\forall i\in N}{\arg\min} \; L_S(\mathbf{w}_i,\alpha_i) + \lambda L_R(\mathbf{w}_i,\alpha_i), \tag{4}$$

where N denotes the number of layers in the network. L_S is the cross entropy, denoting learning from the ground truth. And λ is a hyper-parameter. Different from binarization methods (such as XNOR-Net [17] and Bi-Real Net [9]) where only the reconstruction loss is considered in the weight calculation. By fine-tuning the value of λ, our proposed POEM can achieve much better performance than XNOR-Net, which shows the effectiveness of combined loss against only softmax loss.

2.3 Optimization for POEM

In each Bi-FC layer, POEM sequentially update unbinarized weights $\mathbf{w}_i$ and scale factor α_i. **Updating $\mathbf{w}_i$ via Expectation-Maximization:** Under a binarization framework, the conventional back propagation process without necessary constraint will result in a Gaussian distribution of $\mathbf{w}_i$, which degrades the robustness of Bi-FCs. Our POEM takes another learning objective as

$$\arg\min_{\mathbf{w}_{i,j}} \mathbf{b}^{\mathbf{w}_{i,j}} - \mathbf{b}^{\mathbf{w}_{i,j}+\gamma}, \tag{5}$$

where γ denotes any disturbance caused by the noise containing in the raw point cloud data and $j = 1, \cdots, C_i$ is the channel index. To learn Bi-FCs capable of overcoming this obstacle, we introduce EM algorithm into the updating of $\mathbf{w}_{i,j}$. First, we assume the ideal distribution of $\mathbf{w}_{i,j}$ should be a bi-modal one.

Assumption 2.1 *For every channel of the i-th 1-bit layer, i.e., $\forall \mathbf{w}_{i,j} \in \mathbf{w}_i$, it can be constrained to follow a Gaussian Mixture Model (GMM).*

Based on our assumption, for the j-th channel $\mathbf{w}_{i,j}$ we formulate the ideal bi-modal distribution as

$$\mathcal{P}(\mathbf{w}_{i,j}|\Theta_{i,j}) = \beta_{i,j}^k \sum_{k=1}^{2} p(\mathbf{w}_{i,j}|\Theta_{i,j}^k), \tag{6}$$

where the number of distributions is set as 2 in this paper. $\Theta_{i,j}^k = \{\mu_{i,j}^k, \sigma_{i,j}^k\}$ denotes the parameters of the k-th distribution, *i.e.*, $\mu_{i,j}^k$ denotes the mean value and $\sigma_{i,j}^k$ denotes the variance respectively. To solve GMM with the observed data $\mathbf{w}_i$, *i.e.*, the ensemble of weights at the i-th layer.

We introduce the hidden variable $\xi_{i,j}^{nk}$ to formulate the maximum likelihood estimation (MLE) of GMM as

$$\xi_{i,j}^{jk} = \begin{cases} 1, & \mathrm{w}_{i,j}^n \in p_{i,j}^k \\ 0, & \text{else} \end{cases}, \tag{7}$$

where $n = 1, , C_{i-1}$ is the input channel index of the i-th layer. $\xi_{i,j}^{jk}$ is the hidden variable describing the affiliation of $\mathrm{w}_{i,j}^n$ and $p_{i,j}^k$ (simplified denotation of $p(\mathbf{w}_{i,j}|\Theta_{i,j}^k)$). We then define the likelihood function $\mathcal{P}(\mathrm{w}_{i,j}^n, \xi_{i,j}^{nk}|\Theta_{i,j}^k)$ as

$$\mathcal{P}(\mathrm{w}_{i,j}^n, \xi_{i,j}^{nk}|\Theta_{i,j}^k) = \prod_{k=1}^{2} (\beta_{i,j}^k)^{|p_{i,j}^k|} \prod_{n=1}^{C_{i-1}} \left\{ \frac{1}{\Omega} f(\mathrm{w}_{i,j}^n, \mu_{i,j}^k, \sigma_{i,j}^k) \right\}^{\xi_{i,j}^{nk}}, \tag{8}$$

where $\Omega = \sqrt{2\pi|\sigma_{i,j}^k|}$, $|p_{i,j}^k| = \sum_{n=1}^{C_{i-1}} \xi_{i,j}^{nk}$, and $C_{i-1} = \sum_{k=1}^{2} |p_{i,j}^k|$. And $f(\mathrm{w}_{i,j}^n, \mu_{i,j}^k, \sigma_{i,j}^k)$ is defined as

$$f(\mathrm{w}_{i,j}^n, \mu_{i,j}^k, \sigma_{i,j}^k) = \exp(-\frac{1}{2\sigma_{i,j}^k}(\mathrm{w}_{i,j}^n - \mu_{i,j}^k)^2). \tag{9}$$

Hence, for every single weight $\mathrm{w}_{i,j}^n$, $\xi_{i,j}^{nk}$ can be computed by maximizing the likelihood as

$$\max_{\xi_{i,j}^{nk}, \forall n,k} \mathbb{E}\left[\log \mathcal{P}(\mathrm{w}_{i,j}^n, \xi_{i,j}^{nk}|\Theta_{i,j}^k) | \mathrm{w}_{i,j}^n, \Theta_{i,j}^k \right], \tag{10}$$

where $\mathbb{E}[\cdot]$ represents the estimation. Hence, the maximum likelihood estimation $\hat{\xi}_{i,j}^{nk}$ is calculated as

$$\hat{\xi}_{i,j}^{nk} = \frac{\beta_{i,j}^{k} p(\mathrm{w}_{i,j}^{n} | \Theta_{i,j}^{k})}{\sum_{k=1}^{2} \beta_{i,j}^{k} p(\mathrm{w}_{i,j}^{n} | \Theta_{i,j}^{k})}. \tag{11}$$

After the expectation step, we conduct the maximization step to compute $\Theta_{i,j}^{k}$ as

$$(\hat{\mu}_{i,j}^{k}, \hat{\sigma}_{i,j}^{k}, \hat{\beta}_{i,j}^{k}) = (\frac{\sum_{n=1}^{C_{i-1}} \hat{\xi}_{i,j}^{nk} \mathrm{w}_{i,j}^{n}}{\sum_{n=1}^{C_{i-1}} \hat{\xi}_{i,j}^{nk}}, \frac{\sum_{n=1}^{C_{i-1}} \hat{\xi}_{i,j}^{nk} (\mathrm{w}_{i,j}^{n} - \hat{\mu}_{i,j}^{k})^2}{\sum_{n=1}^{C_{i-1}} \hat{\xi}_{i,j}^{nk}}, \frac{\sum_{n=1}^{C_{i-1}} \hat{\xi}_{i,j}^{nk}}{C_{i-1}}). \tag{12}$$

Then, we optimize $\mathrm{w}_{i,j}^{n}$ as

$$\delta_{\mathrm{w}_{i,j}^{n}} = \frac{\partial L_S}{\partial \mathrm{w}_{i,j}^{n}} + \lambda \frac{\partial L_R}{\partial \mathrm{w}_{i,j}^{n}} + \tau EM(\mathrm{w}_{i,j}^{n}), \tag{13}$$

where τ is hyper-parameter to control the proportion of Expectation-Maximization operator $EM(\mathrm{w}_{i,j}^{n})$. $EM(\mathrm{w}_{i,j}^{n})$ is defined as

$$EM(w_{i,j}^{n}) = \begin{cases} \sum_{k=1}^{2} \hat{\xi}_{i}^{jk} (\hat{\mu}_{i,j}^{k} - \mathrm{w}_{i,j}^{n}), & \hat{\mu}_{i,j}^{1} < \mathrm{w}_{i,j}^{n} < \hat{\mu}_{i,j}^{2} \\ 0, & \text{else} \end{cases}. \tag{14}$$

And further we have

$$\frac{\partial L_R}{\partial \mathbf{w}_i} = (\mathbf{w}_i - \alpha_i \circ \mathbf{b}^{\mathbf{w}_i}) \circ \alpha_i. \tag{15}$$

Updating α_i: We further update the scale factor α_i with $\mathbf{w}_i$ fixed. δ_{α_i} is defined as the gradient of α_i, and we have

$$\delta_{\alpha_i} = \frac{\partial L_S}{\partial \alpha_i} + \lambda \frac{\partial L_R}{\partial \alpha_i}. \tag{16}$$

The gradient derived from softmax loss can be easily calculated according to back propagation. Base on Eq. 3, we have

$$\frac{\partial L_R}{\partial \alpha_i} = (\mathbf{w}_i - \alpha_i \circ \mathbf{b}^{\mathbf{w}_i}) \cdot \mathbf{b}^{\mathbf{w}_i}. \tag{17}$$

The above derivations show that POEM is learnable with the BP algorithm based on a simple and effective reconstruction loss function. Moreover, we introduce EM to optimize unbinarized weights, which further constrain them to formulate a bi-modal distribution. We describe our algorithm in supplementary materials.

3 Implementation and Experiments

In this section, we conduct extensive experiments to validate the effectiveness of our proposed POEM for efficient learning on point clouds. We first ablate our method and demonstrate the contributions of our work on the most fundamental tasks: classification on ModelNet40 [20]. Moreover, we implement our POEM on mainstream models on three tasks, *i.e.*, classification on ModelNet40 [20], part segmentation on ShapeNet Parts [2], and semantic segmentation on S3DIS [1]. We compare POEM with existing binarization methods where our designs stand out.

1-bit PointNet		λ			
		1×10^{-3}	1×10^{-4}	1×10^{-5}	0
τ	1×10^{-2}	89.3	89.0	86.3	81.9
	1×10^{-3}	88.3	**90.2**	87.9	82.5
	1×10^{-4}	86.5	87.1	85.5	81.4
	0	82.7	85.3	83.7	80.1

Table 1: Ablation study on hyper-parameter λ and τ. We vary λ from 1×10^{-3} to 0 and τ from 1×10^{-2} to 0, respectively.

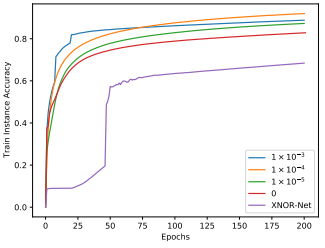


Figure 4: Training accuracies of POEM ($\tau = 1\times10^{-3}$) with different λ and XNOR-Net.

3.1 Datasets and Implementation Details

Datasets: ModelNet40 [20] is used for classification. The ModelNet40 dataset is the most frequently used datasets for shape classification. ModelNet is a popular benchmark for point cloud classification. It contains 12,311 CAD models from 40 representative classes of objects.

We employ ShapeNet Parts [2] for part segmentation. ShapeNet contains 16,881 shapes from 16 categories, 2,048 points are sampled from each training shape. Each shape is split into two to five parts depending on the category, making up to 50 parts in total.

For the semantic segmentation, S3DIS [1] is employed. S3DIS includes 3D scan point clouds for 6 indoor areas, including 272 rooms in total, and each point belongs to one of 13 semantic categories.

Implementation Details: We evaluate POEM on three mainstream models, including PointNet [14], PointNet++ [15] and DGCNN [18], on three main point cloud tasks, *i.e.*, classification, part segmentation and semantice segmentation. In our experiments, 4 NVIDIA GeForce TITAN V GPUs are used.

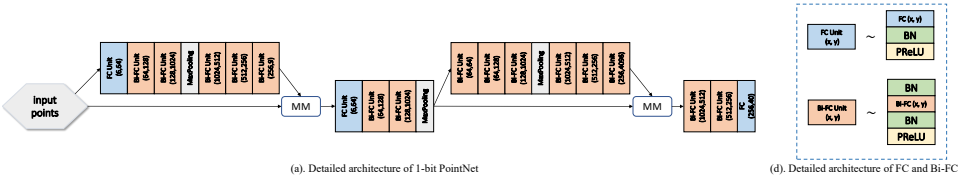
On the classification task, 1-bit PointNet is built by binarizing the full-precision PointNet via POEM. All fully-connected layers in PointNet except the first and last one are binarized to the Bi-FC layer, and we select PReLU [6] instead of ReLU as the activation function when binarizing the activation before the next Bi-FC layer. We also extend this binarization setting to other tasks. We also provide our PointNet baseline under this setting. For other 1-bit networks, we also follow this implementation. In all tables, we use the bold typeface to denote the best result.

For the part segmentation task, we follow the convention [14] to train a model for each of the 16 classes of ShapeNet Parts [2]. For semantic segmentation task on S3DIS [1], we also follow the same setups an [14].

Following PointNet [14], we train 200 epochs, 250 epochs, 128 epochs on point cloud classification, part segmentation, semantic segmentation respectively. To stably train the 1-bit networks, we use learning rate 0.001 with Adam and Cosine Annealing learning rate decay for all 1-bit models on all tasks.

3.2 Ablation Study

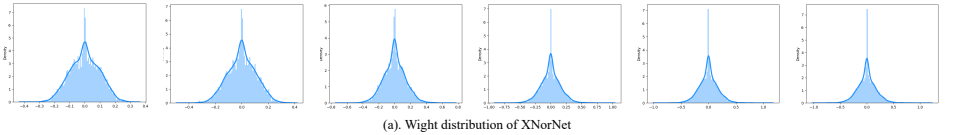
Hyper-parameter selection: Hyper-parameters λ and τ in Eq. 4 and 13 are related to the reconstruction loss and EM algorithm. The effect of parameters λ and τ are evaluated on ModelNet40 for 1-bit PointNet. The Adam optimization algorithm is used during the training process, with per batch sized as 592. Using different values of λ and τ, the performance of



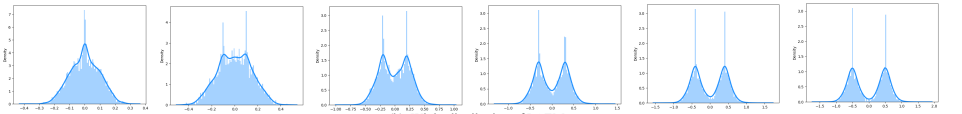
(a). Detailed architecture of 1-bit PointNet

(d). Detailed architecture of FC and Bi-FC

Figure 5: Detailed architecture of 1-bit PointNet implemented by us. MM denotes matrix multiplication in short.



(a). Wight distribution of XNorNet



(b). Wight distribution of POEM

Figure 6: (a) and (b) illustrate the distribution of the unbinarized weights $\mathbf{w}_i$ of the 6-th 1-bit layer in 1-bit PointNet backbone when being trained under XNOR-Net and our POEM, respectively. From left to right, we report the weight distribution of initialization, 40-th, 80-th, 120-th, 160-th, and 200-th epoch. Our POEM obtains an apparent bi-modal distribution, which is much more robust.

POEM is shown in Table 1. In Table 1, from left to right lie the overall accuracies (OAs) with different λ from 1×10^{-3} to 0.

And the OAs with different τ from 1×10^{-2} to 0 lie from top to bottom. With the decrease of λ, the OA increases first and then drops dramatically. The same trend is shown when we decrease τ. We get the optimal 1-bit PointNet with POEM with $\{\lambda,\tau\}$ set as $\{1\times10^{-4},1\times10^{-3}\}$. Hence, we extend this hyper-parameter set to the other experiments involved in this paper.

We also set τ as 1×10^{-3} and plot the growth curve of training accuracies of POEM with different λ and XNOR-Net. As shown in Figure 4, 1-bit PointNet obtained by POEM achieves the optimal training accuracy when λ is set as 1×10^{-4}. Also, with the EM optimized back propagation, the convergence of weights becomes better than XNOR-Net (in purple), as shown in Figure 4.

Evaluating the components of POEM: In this part, we evaluate every critical part of POEM to show how we compose the novel and effective POEM.

We first introduce our baseline network by adding a single BN layer ahead of the 1-bit convolutions of XNOR-Net, which brings 1.2% improvement on OA. As shown in Table 2, the introduction of PReLU, EM, and learnable scale factor improves the accuracy by 1.9%, 3.1% and 3.4% respectively over the baseline network, as shown in the second section of Table 2. By adding all the PReLU, EM, and the learnable scale factor, our POEM achieves 7.1% higher accuracy than the baseline, even surpassing the corresponding real-valued network's accuracy.

Compared to merely using the PReLU, our main contributions, EM and learnable scale

1-bit PointNet	OA (%)
XNOR-Net	81.9
Proposed baseline network	83.1
Proposed baseline network + PReLU	85.0
Proposed baseline network + EM	86.2
Proposed baseline network + LSF	86.5
Proposed baseline network + PReLU + EM + LSF (POEM)	90.2
Real-valued Counterpart	89.2

Table 2: The effects of different components in POEM on the OA. PReLU, EM and LSF denote components of proposed baseline network.

Model	Method	W/A (bit)	OA (%)
PointNet	Real-valued	32/32	89.2
	XNOR-Net	1/1	81.9
	Bi-Real Net		77.5
	BiPointNet		86.4
	POEM		**90.2**
PointNet++	Real-valued	32/32	91.9
	XNOR-Net	1/1	83.8
	BiPointNet		87.8
	POEM		**91.2**
DGCNN	Real-valued	32/32	89.2
	XNOR-Net	1/1	81.5
	BiPointNet		83.4
	POEM		**91.1**

Table 3: Our methods on mainstream networks on classification task with ModelNet40 dataset.

factor, boost the accuracy by 5.2%, which is very significant on point cloud task. The 1-bit PointNet achieves the performance, which even surpasses the real-valued PointNet baseline with 1.0% (90.2% vs. 89.2%).

Weight distribution: The POEM-based model is based on an Expectation-Maximization process implemented on PyTorch [13] platform. We analyze the weight distribution of training XNOR-Net and POEM for comparison to confirm our motivation. For a 1-bit PointNet model, we analyze the 6-th 1-bit layer sized $(64, 64)$ and having 4096 elements. We plot its weight distribution at the $\{0, 40, 60, 120, 160, 200\}$-th epochs. As seen in Figure 6, the initialization (0-th epoch) is the same for XNOR-Net and POEM. However, our POEM efficiently employs the Expectation-Maximization algorithm to supervise the back propagation process, leading to an effective and robust bi-modal distribution. This analysis also compiles with the performance comparison in Table 2.

3.3 Comparison with State-of-the-arts

Classification on ModelNet40: Table 3 shows that our POEM outperforms other binarization methods such as XNOR-Net [17], Bi-Real Net [9] and BiPointNet [16] on classification task with ModelNet40 dataset. We implement comparative experiments on three mainstream backbones: PointNet [14], PointNet++ [15] and DGCNN [18]. XNOR-Net and Bi-Real Net have been proven effective in 2D vision, and we successfully transfer them to point clouds.

Specifically, on PointNet, our POEM outperforms XNOR-Net, Bi-Real Net, and BiPointNet by 9.3%, 12.7%, and 3.8% respectively. Moreover, 1-bit PointNet obtained by POEM even surpasses the real-valued PointNet by 1.0% OA. On PointNet++, POEM stands out from all other 1-bit methods by a sizable performance advance. For example, POEM outperforms BiPointNet by 3.4% OA improvement. Similar circumstances arise on the DGCNN backbone, and our POEM surpasses BiPointNet by 7.7%. All the results demonstrate that our POEM promotes the state-of-the-art in 1-bit point cloud classification.

Part Segmentation on ShapeNet Parts: We demonstrate the superiority of our POEM on part segmentation task in Table 4. We have two observations from the impressive results: 1). POEM can achieve the best performance (mIOU) compared with other 1-bit methods on all three backbones; 2). Compared with real-valued counterparts, acceptable performance drops are achieved (2.6%, 2.2% and 2.1%) with significant compression rates.

Model	Method	W/A (bit)	mIOU
PointNet	Real-valued	32/32	83.7
	XNOR-Net	1/1	75.3
	Bi-Real Net		70.0
	BiPointNet		80.6
	POEM		**81.1**
PointNet++	Real-valued	32/32	85.1
	XNOR-Net	1/1	77.7
	POEM		**82.9**
DGCNN	Real-valued	32/32	85.2
	XNOR-Net	1/1	77.4
	POEM		**83.1**

Table 4: Our methods on mainstream networks on part segmentation task with ShapeNet Part dataset.

Model	Method	W/A (bit)	mIOU	OA (%)
PointNet	Real-valued	32/32	47.7	78.6
	XNOR-Net	1/1	39.1	70.4
	Bi-Real Net		35.5	65.0
	BiPointNet		44.3	76.7
	POEM		**45.8**	**77.9**
PointNet++	Real-valued	32/32	53.2	82.7
	XNOR-Net	1/1	43.1	75.9
	POEM		**49.8**	**80.4**
DGCNN	Real-valued	32/32	56.1	84.2
	XNOR-Net	1/1	45.6	78.0
	POEM		**50.1**	**81.3**

Table 5: Our methods on mainstream networks on semantic segmentation task with S3DIS dataset.

Semantic Segmentation on S3DIS: As listed Table 5, our POEM outperforms all other 1-bit methods on part segmentation tasks. First, POEM can achieve the best mIOU and OA compared with other 1-bit methods on all employed backbones. Second, POEM can inference the implemented backbones with efficient XNOR and Bit-count operations with acceptable mIOU drops (1.9%, 3.4% and 2.9%) achieved. Moreover, our POEM can achieve OA over 80% with 1-bit weights and activations, which promotes the state-of-the-art.

Except for these performance comparison, we also provide efficiency analysis and results visualization in the supplementary material to sufficiently evaluate our POEM.

4 Conclusion

We have developed a new deep learning model for point cloud processing, 1-bit point-wise operations based on Expectation-Maximization (POEM), which can significantly reduce the storage requirement for computationally limited devices. POEM is employed in point cloud networks to formulate 1-bit fully-connected layer (Bi-FC), which mainly works with binary weights, proposed scale factor, and Expectation-Maximization (EM) algorithm. In POEMs, we use the learnable scale factor to build an end-to-end framework and a new architecture to calculate the network model. To further enhance the robustness of unbinarized weights, we employ the EM algorithm to learning a bi-modal distribution of unbinarized weights. All the parameters of our POEM can be obtained in the same pipeline as in the back propagation algorithm. Extensive experiments demonstrate that our POEM surpasses existing binarization methods by significant margins. For more real-world applications, we will implement our POEM on ARM CPUs for future work.

5 Acknowledgement

Sheng Xu and Yanjing Li are the co-first authors. Baochang Zhang is the corresponding author. The work was supported by the National Natural Science Foundation of China (62076016, 61972016). This study was supported by Grant NO.2019JZZY011101 from the Key Research and Development Program of Shandong Province to Dianmin Sun.

References

[1] 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*, pages 1534–1543, 2016.

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[3] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.

[4] Jiaxin Gu, Ce Li, Baochang Zhang, Jungong Han, Xianbin Cao, Jianzhuang Liu, and David Doermann. Projection convolutional neural networks for 1-bit cnns via discrete back propagation. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 8344–8351, .

[5] Jiaxin Gu, Junhe Zhao, Xiaolong Jiang, Baochang Zhang, Jianzhuang Liu, Guodong Guo, and Rongrong Ji. Bayesian optimized 1-bit cnns. In *Proceedings of IEEE International Conference on Computer Vision*, pages 4909–4917, .

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[7] Pedro Hermosilla, Tobias Ritschel, and Timo Ropinski. Total denoising: Unsupervised learning of 3d point cloud cleaning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 52–60, 2019.

[8] 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 Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020.

[9] 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.

[10] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In *Proceedings of European Conference on Computer Vision*, pages 143–159, 2020.

[11] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Proceedings of Advances in Neural Information Processing Systems*, pages 965–975, 2019.

[12] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019.

[14] 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*, pages 652–660, 2017.

[15] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[16] Haotong Qin, Zhongang Cai, Mingyuan Zhang, Yifu Ding, Haiyu Zhao, Shuai Yi, Xianglong Liu, and Hao Su. Bipointnet: Binary neural network for point clouds. In *Proceedings of International Conference on Learning Representations*, pages 1–24, 2020.

[17] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proceedings of European Conference on Computer Vision*, pages 525–542, 2016.

[18] 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*, 38(5):1–12, 2019.

[19] Ziwei Wang, Ziyi Wu, Jiwen Lu, and Jie Zhou. Bidet: An efficient binarized object detector. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2049–2058, 2020.

[20] 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*, pages 1912–1920, 2015.

[21] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020.

[22] Sheng Xu, Zhendong Liu, Xuan Gong, Chunlei Liu, Mingyuan Mao, and Baochang Zhang. Amplitude suppression and direction activation in networks for 1-bit faster r-cnn. In *Proceedings of the International Workshop on Embedded and Mobile Deep Learning*, pages 19–24, 2020.

[23] Sheng Xu, Junhe Zhao, Jinhu Lu, Baochang Zhang, Shumin Han, and David Doermann. Layer-wise searching for 1-bit detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5682–5691, 2021.