mathematics

MDPI

Article

Quantized Graph Neural Networks for Image Classification

Xinbiao Xu [1], Liyan Ma [1,*], Tieyong Zeng [2] and Qinghua Huang [3]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; xbxu@shu.edu.cn
[2] Department of Mathematics, The Chinese University of Hong Kong, Hong Kong 999077, China; zeng@math.cuhk.edu.hk
[3] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; qinghua@shu.edu.cn
* Correspondence: liyanma@shu.edu.cn

Abstract: Researchers have resorted to model quantization to compress and accelerate graph neural networks (GNNs). Nevertheless, several challenges remain: (1) quantization functions overlook outliers in the distribution, leading to increased quantization errors; (2) the reliance on full-precision teacher models results in higher computational and memory overhead. To address these issues, this study introduces a novel framework called quantized graph neural networks for image classification (QGNN-IC), which incorporates a novel quantization function, Pauta quantization (PQ), and two innovative self-distillation methods, attention quantization distillation (AQD) and stochastic quantization distillation (SQD). Specifically, PQ utilizes the statistical characteristics of distribution to effectively eliminate outliers, thereby promoting fine-grained quantization and reducing quantization errors. AQD enhances the semantic information extraction capability by learning from beneficial channels via attention. SQD enhances the quantization robustness through stochastic quantization. AQD and SQD significantly improve the performance of the quantized model with minimal overhead. Extensive experiments show that QGNN-IC not only surpasses existing state-of-the-art quantization methods but also demonstrates robust generalizability.

Keywords: graph neural network;model quantization; knowledge distillation; image classification

MSC: 68T07

Citation: Xu, X.; Ma, L.; Zeng, T.; Huang, Q. Quantized Graph Neural Networks for Image Classification. *Mathematics* **2023**, *11*, 4927. https://doi.org/10.3390/math11244927

Academic Editors: Pedro A. Castillo Valdivieso and Florin Leon

Received: 3 November 2023
Revised: 29 November 2023
Accepted: 8 December 2023
Published: 11 December 2023

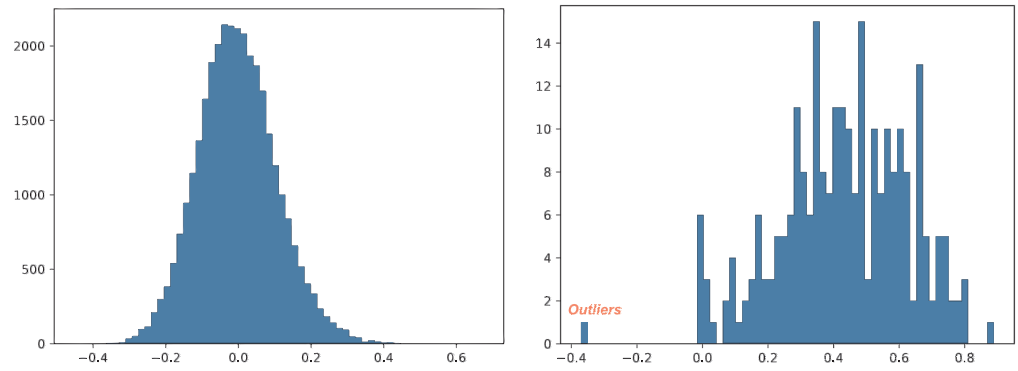
1. Introduction

Graph neural networks (GNNs) are considered the intersection of deep learning and graph theory [1,2]. Originally designed to handle data with graph structures, they capture relationships between nodes and propagate information through the graph structure [2]. In recent years, the application of graph neural networks (GNNs) in computer vision tasks has been extensively explored. The fusion of visual networks with GNNs has led to significant improvements in model comprehension and breakthroughs in various visual tasks, including image classification [3–5], object detection [6,7], and semantic segmentation [8,9]. However, the integration of these networks results in larger model sizes and higher computational costs, limiting their practical application on resource-constrained devices [10]. To address these challenges, several compression and acceleration methods have been proposed, including pruning [11–13], quantization [14–17], kernel decomposition [18], and efficient inference backends [19–21].

Among these methods, quantization has emerged as one of the most effective approaches. It uses quantization functions to reduce the bit width of network weights and activations, leading to faster inference and reduced memory usage [14,22]. For instance, LSQ [23] sets a learnable scale factor in the quantization function, achieving adaptive fine-grained quantization through backpropagation. N2UQ [24] realizes a better adaptation to distribution by learning non-uniform input thresholds to quantize inputs into equidistant

output levels. AdaQP [25] applies stochastic quantization to message vectors transmitted across devices, thereby reducing communication traffic. WGCN [26] proposes the integration of Haar wavelet transforms and quantization functions to compress graph channels, achieving computational savings through channel shrinkage. CoGNN [27] introduces the concept of reuse-aware sampling, performing node-level parallel-aware quantization to reduce the overhead of feature aggregation.

Despite advancements in designing better quantization methods, quantized GNNs for visual tasks still face significant challenges. The GNNs for visual tasks can be divided into two parts: visual networks for extracting semantic features from images and graph networks for identifying image relationships through message-passing mechanisms. In visual networks, weights often approximate a Gaussian distribution with a mean close to zero [28,29]. However, in graph networks, weights tend to deviate from zero and introduce outliers (see Figure 1b). These outliers expand the quantization range, leading to coarse-grained quantization and higher quantization errors. Consequently, the performance of quantized models significantly deteriorates. Although researchers have employed knowledge distillation (KD) [30–32], this approach has not fundamentally solved the problem of outliers. Moreover, KD introduces an additional teacher model, resulting in higher computational and memory costs for training, contradicting the intention of quantization to improve model efficiency.



(**a**) (**b**)

Figure 1. Weight distribution of pre-trained visual networks and graph networks. The x-axis represents the weights, and the y-axis represents the frequency. (**a**) Weight distribution of the 3rd layer of the visual network; (**b**) distribution of the weights of the 4th layer of the graph network.

To address these challenges, this paper explores the mechanisms of quantized GNNs used for image classification. We propose the quantized graph neural networks for image classification (QGNN-IC), which incorporates a novel quantization function called Pauta Quantization (PQ) and two plug-and-play self-distillation methods: attention quantization distillation (AQD) and stochastic quantization distillation (SQD). Specifically, PQ utilizes the mean and standard deviation of the input distribution to remove outliers, enabling fine-grained quantization and consequently reducing quantization errors, thereby enhancing the accuracy of the quantized model. AQD utilizes attention mechanisms to transfer more beneficial information to the quantized network, enhancing the visual network's ability to extract semantic information. SQD enhances the robustness of the model to quantization by minimizing the information discrepancy between the randomly quantized branch and the fully quantized branch. AQD and SQD significantly improve the performance of quantized models without external teacher models, resulting in minimal additional computational and memory overhead during the training phase. Experimental results indicate that the 2-bit quantized model exhibits enhancements in accuracy of 1.92% and 0.95% on CIFAR-FS and CUB-200-2011, respectively. Moreover, the experiments also validate the generalizability of QGNN-IC.

The remainder of this paper is organized as follows: Section 2 describes some related works to facilitate comprehension. Section 3 clarifies the details of QGNN-IC, including

quantization functions, self-distillation methods, loss functions, etc. Section 4 reports the experimental results of the quantized models across different datasets and provides an analysis of the results. Finally, Section 5 discusses the results and summarizes the article.

2. Related Work

2.1. Model Quantization

Model quantization aims to reduce the bit width of network weights and activations, leading to faster computation and reduced memory usage. It can be divided into post-training quantization (PTQ) and quantization-aware training (QAT). PTQ compresses a pre-trained full-precision model into a low-bit model without retraining or fine-tuning [33–35]. However, PTQ often suffers from significant accuracy degradation due to the lack of weight-aware training. Conversely, QAT quantizes the network and retrains it simultaneously to better adapt to the information loss caused by quantization [23,29,36–38]. QAT methods such as N2UQ [24] and LSQ [23] have been proposed to optimize the quantization process. N2UQ [24] quantizes the input into equidistant output levels by learning non-uniform input thresholds to better adapt to the distribution. LSQ [23] sets the scaling factor as a learnable parameter to provide finer optimization. However, these methods do not consider outliers in the data distribution, leading to coarser quantization and higher quantization errors, ultimately impacting the performance of the quantized model.

2.2. Graph Neural Networks for Computer Vision

GNNs have demonstrated strong capabilities in relational analysis [3,39], and researchers have attempted to transfer this idea to visual tasks. Firstly, the visual network is employed to extract semantic features from input images. Then, the extracted semantic features are used to initialize the graph nodes. Subsequently, deep semantic relations are extracted using graph networks. Finally, the results are outputted to modules related to downstream tasks. For instance, in image classification, graph networks are often used to compute the similarity between unlabeled query sets and labeled support sets for classification [5,40–42]. Moreover, GNNs have also been applied to object detection [6,7] and semantic segmentation [8,9], achieving promising results. Despite their remarkable performance, the fusion of visual networks and graph networks introduces high computational complexity, complex parameters, and frequent memory access, making it challenging to deploy the models to edge devices for practical applications. This paper is the first study on quantized GNNs for visual tasks, and we believe it will contribute to the deployment of state-of-the-art GNNs in a wider range of real-world computer vision applications.

2.3. Knowledge Distillation

Knowledge distillation (KD) aims to transfer knowledge from a teacher model to a student network [43]. Some quantization methods utilize knowledge distillation to recover the performance of quantized student models. For example, logit-level quantization distillation [44] and feature-level quantization distillation [45] have been proposed to leverage the predictions and intermediate features of the teacher model, respectively. Progressive distillation [46] has also been explored, where a full-precision model guides the training of a quantized model, which then becomes the new teacher for the next quantized model. Collaborative distillation [31] has been proposed to leverage multiple teacher models to improve the performance of the quantized student model. However, these methods require at least one additional full-precision teacher model, introducing additional computational and memory overheads, contradicting the goal of quantization. To address this limitation, we propose two plug-and-play self-distillation modules that extract knowledge from the quantized student model itself without a teacher model, significantly reducing the training time.

3. Methodology

This section provides a comprehensive overview of the proposed QGNN-IC (see Figure 2). Initially, we introduce the symbols and typical formulas used in model quantization (Section 3.1). Subsequently, we propose a novel quantization function, PQ, designed to minimize the quantization error caused by outliers (Section 3.2). In Sections 3.3 and 3.4, we present two distinct self-distillation methods: AQD and SQD. AQD aims to enhance the semantic representation capability of quantized visual networks, while SQD seeks to improve the quantization robustness of quantized graph networks. Lastly, in Section 3.5, we apply all the proposed methods to a graph neural network for image classification.

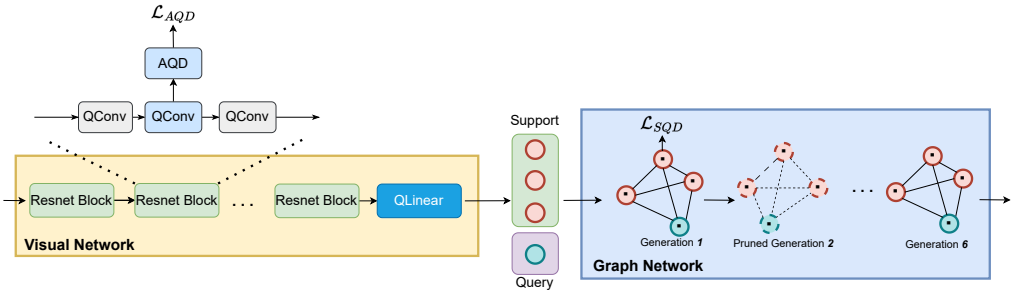


Figure 2. Illustration of the quantized graph neural networks for image classification (QGNN-IC). Support and query represent the features of labeled samples and unlabeled samples, respectively. QConv and QLinear represent the quantized convolutional and linear layers, respectively.

3.1. Preliminary

Model quantization converts the full-precision (32-bit) weights and activations to lower bit-width values, such as 8-bit fixed-point integers, using a quantization function. The n-bit quantization function is defined as follows:

$$Q(X) = Clip(round\left(\frac{X}{s} + z\right), -2^{n-1}, 2^{n-1} - 1), \tag{1}$$

where X represents the weights or activations, z is the zero-point, $round(\cdot)$ denotes the rounding function, $Clip(\cdot, -2^{n-1}, 2^{n-1} - 1)$ is a clipping function that limits the lower bound of the values to -2^{n-1} and the upper bound to $2^{n-1} - 1$, and $s = \frac{X_{\max} - X_{\min}}{2^n - 1}$ is the scaling factor, with $X_{\max}$ and $X_{\min}$ being the maximum and minimum values of X. The corresponding dequantized value $\hat{X}$ can be calculated as:

$$\hat{X} = s \cdot Q(X). \tag{2}$$

During backpropagation, the straight-through estimator (STE) [47] is used to mitigate the gradient vanishing problem caused by the rounding function.

Finally, we define the quantization error with respect to X as:

$$E_Q(X) = ||X - \hat{X}||_2. \tag{3}$$

Generally, a smaller quantization error corresponds to a lesser decrease in model performance.

3.2. Pauta Quantization

GNNs for computer vision comprise two components: visual networks that extract semantic features from images and graph networks that identify image relationships using message-passing mechanisms. As depicted in Figure 1, the weights in visual networks are typically approximated by Gaussian distributions with a mean close to zero [28,29]. On the other hand, in the graph network, the weights often have means far from zero and may include some outliers. These outliers can expand the quantization range, leading to a larger scaling factor s and coarser quantization. Coarse quantization induces a high

quantization error near the means, which impairs the performance of the quantized model. Therefore, the identification and management of outliers are crucial for enhancing the quantized models.

The Pauta criterion, a method for identifying outliers in sample data, utilizes the mean and standard deviation as thresholds to detect significant errors. If the error exceeds this threshold, it is deemed a significant error rather than a random error. These outliers should be removed as they are deemed unacceptable.

Inspired by the above analysis and the Pauta criterion, we propose Pauta quantization (PQ). Specifically, we first calculate the mean and standard deviation of the distribution and determine the quantization range $[l, r]$:

$$l = \text{mean}(X) - 3\sigma(X), \tag{4}$$

$$r = \text{mean}(X) + 3\sigma(X), \tag{5}$$

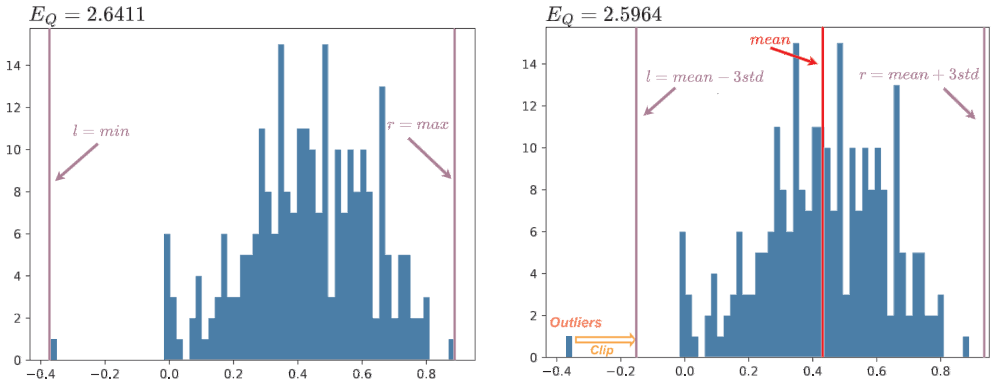
where $\text{mean}(\cdot)$ is the mean function, and $\sigma(\cdot)$ is the standard deviation function. Then, we perform precise quantization by excluding outliers. The formula for n-bit PQ is as follows:

$$s = \frac{r - l}{2^n - 1}, \tag{6}$$

$$PQ(X) = round(\frac{Clip(X - l, 0, r - l)}{s}), \tag{7}$$

where $Clip(\cdot, 0, r - l)$ is a clipping function that limits the lower bound of the values to 0 and the upper bound to $r - l$. l and r are set as learnable parameters.

As shown in Figure 3, PQ analyzes the data distribution and estimates the quantization range appropriately, resulting in fine-grained quantization. This allows a large number of values distributed around the mean to be quantized to more suitable integers. Compared to the original quantization function, PQ reduces the quantization error by approximately 1.7%.



(**a**) (**b**)

Figure 3. Illustration of the impact of different quantization functions on distributions with outliers. The x-axis represents the weights, and the y-axis represents the frequency. E_Q represents the quantization error. (**a**) Illustration of the original quantization function. (**b**) Illustration of Pauta quantization (PQ).

3.3. Attention Quantization Distillation

In visual tasks, GNNs leverage visual networks to extract rich semantic information from images and then use graph networks to perform message passing based on feature similarity. However, quantization compresses the data into low bit widths, leading to a loss of semantic information in visual networks. Knowledge distillation is a common method to assist in training quantized models, using a full-precision teacher model to enhance the semantic information extraction capability of the quantized model. However, the

additional teacher model introduces greater computational and memory burdens during the training phase.

To mitigate this, we propose AQD specifically for visual networks. AQD aims to enhance the semantic feature extraction capability of the quantized visual network by identifying beneficial high-level feature information from the latent full-precision activations. As shown in Figure 4, we allow the quantized convolutional layer (Qconv) to output both full-precision and quantized activations:

$$A_f = W \otimes A, \tag{8}$$

$$A_q = PQ(W) \otimes PQ(A), \tag{9}$$

where A represents the activations, subscripts f and q indicate whether it is full-precision or quantized; W represents the weights; $\otimes$ denotes the convolution operation; $PQ(\cdot)$ is the quantization function proposed in Section 3.2. Considering the distribution shift caused by the precision difference, it is necessary to normalize these two types of activations:

$$H(A) = Flatten(BN(A)), \tag{10}$$

where $Flatten(\cdot)$: $\mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{C\times HW}$ flattens the activations after normalization; H, W, and C are the height, width, and the number of channels, respectively; BN denotes the batch normalization layer.

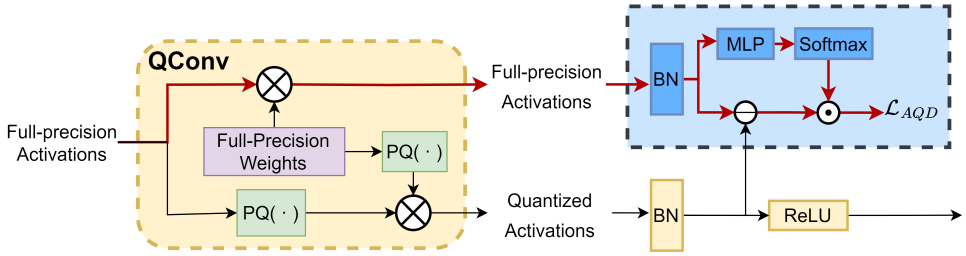


Figure 4. Illustration of proposed attention quantization distillation (AQD).

Since different channels have different importance, it is crucial to further evaluate the importance of channels using a multi-layer perceptron (MLP) so that AQD can adaptively focus on channels with more discriminative features. We define the importance α of different channels as:

$$\alpha = softmax(MLP(H(A_f))), \tag{11}$$

where $\alpha \in \mathbb{R}^C$, $softmax(MLP(\cdot)) : \mathbb{R}^{C\times HW} \rightarrow \mathbb{R}^C$. With α, AQD emphasizes channels with more discriminative capabilities by minimizing the loss function $\mathcal{L}_{AQD}$, which can be expressed as:

$$\mathcal{L}_{AQD} = \sum_{i=1}^{C} \alpha_i || \frac{H_i(A_f)}{||H_i(A_f)||_2} - \frac{H_i(A_q)}{||H_i(A_q)||_2} ||_2^2, \tag{12}$$

where C represents the number of channels in the activations.

As shown in Figure 2, AQD is deployed in the shallow layers of the entire model, which not only improves the ability to extract semantic information but also alleviates the problem of gradient vanishing. Moreover, AQD does not require any additional teacher models and only introduces relatively low computational and memory overhead during the training phase.

3.4. Stochastic Quantization Distillation

Quantization for neural networks, especially below 4-bit, results in a significant degradation of model performance [23,48]. As shown in Table 1, this problem is more severe in

GNNs used for computer vision. The quantization of weights and activations introduces quantization errors and gradient errors, resulting in degraded model performance. Furthermore, as the model becomes deeper, these negative effects further accumulate in graph networks. Therefore, improving the robustness of graph networks to quantization is crucial for enhancing model performance.

Table 1. Quantitative comparison of 5-way 1-shot classification accuracy (%) on CIFAR-FS. N-bit represents bit widths of weights and activations. The accuracy of the full-precision model is 77.68%. Bold text represents the best performance for that column under the n-bit setting.

Method	Bit Width		
	2-bit	**3-bit**	**4-bit**
LSQ	35.81	62.60	71.67
DoReFa	23.96	34.07	67.88
PACT	27.29	63.17	70.48
EWGS	31.18	71.19	75.25
N2UQ	53.53	71.95	76.41
CMT-KD	54.52	71.84	75.51
QGNN-IC (ours)	**56.44**	**72.48**	**76.74**

Dropout [49] is a widely used strategy that randomly drops some neurons during the model training process to enhance the model's robustness. If we randomly quantize, i.e., randomly discard the quantization functions during the training process, can we obtain a more robust quantized model? Wei et al. [34] have shown that quantizing part of modules during PTQ can flatten the loss landscape. Therefore, it is feasible to enhance the robustness of quantized models through stochastic quantization.

Based on the above challenges and analysis, we propose SQD, which extracts more robust information from randomly quantized activations to guide the training of fully quantized activations. The formula for stochastic quantization is defined as:

$$SQ(X) = \begin{cases} PQ(X), & 0 < \xi < p, \\ X, & p < \xi < 1, \end{cases} \tag{13}$$

where ξ is a random sampling variable following a uniform distribution within the range (0,1); p is a hyperparameter; $PQ(\cdot)$ is the proposed quantization function in Section 3.2; X is a 32-bit full-precision tensor.

The matrix operation results of weights and activations are randomly quantized by $SQ(\cdot)$ to obtain P_{rand}. Simultaneously, P_{full} is obtained by using the function $PQ(\cdot)$. The equations are as follows:

$$P_{rand} = SQ(WA), \tag{14}$$

$$P_{full} = PQ(W)PQ(A). \tag{15}$$

Compared with P_{full}, P_{rand} contains more accurate and robust visual semantic information. Therefore, two special similarity pattern matrices can be constructed to help improve the visual characteristics of P_{full}:

$$\hat{P}_{rand} = \frac{P_{rand} \times P_{rand}^{\top}}{\left\| P_{rand} \times P_{rand}^{\top} \right\|}, \tag{16}$$

$$\hat{P}_{full} = \frac{P_{full} \times P_{full}^{\top}}{\left\| P_{full} \times P_{full}^{\top} \right\|}. \tag{17}$$

We distill the semantic matrix $\hat{P}_{rand}$ to enhance the performance of quantized graph networks:

$$\mathcal{L}_{SQD} = \left\| \hat{P}_{rand} - \hat{P}_{full} \right\|. \tag{18}$$

As shown in Figure 2, SQD is deployed on the nodes of the graph network to alleviate the cumulative effects of quantization errors. Furthermore, SQD serves as a self-distillation module, which does not require an additional teacher model and only incurs a small amount of computational and memory overhead during the training phase.

3.5. Quantized Graph Neural Networks for Image Classification

Without loss of generality, we apply the proposed methods to the distribution propagation graph network (DPGN) [5], which is a GNN used for image classification tasks, and obtain the QGNN-IC. As shown in Figure 2, QGNN-IC extracts image features through a visual network, initializes the nodes of the graph network with these features, and then utilizes message-passing mechanisms to compute the similarity between unlabeled and labeled samples for classification. Since the first layer in the visual network is directly related to the input and the last convolutional layer in each generation of the graph network is directly related to the prediction, we maintain these in full precision and quantize all other layers. AQD is inserted into the quantized convolutional layers (QConv) of the visual network, while SQD is deployed on the graph nodes of the graph network.

To further compress the model and improve training and inference speed, we need to strike a balance between accuracy and efficiency. Focusing on the graph network, we find that this part incurs significant computational overhead, so we perform generation-level pruning on the graph network. While this approach slightly reduces the performance of the quantized model, it greatly reduces the computational and parameter overhead. We believe this will facilitate the deployment of state-of-the-art graph neural networks in more real-world computer vision applications.

The overall training loss of $\mathcal{L}_{QGNN-IC}$ can be defined as:

$$\mathcal{L}_{QGNN-IC} = \mathcal{L}_{DPGN} + \lambda_1 \mathcal{L}_{AQD} + \lambda_2 \mathcal{L}_{SQD}, \tag{19}$$

where $\mathcal{L}_{DPGN}$ is the classification loss of DPGN; λ_1 and λ_2 are hyperparameters used to balance the loss function.

4. Experiments and Results

To validate the effectiveness of the proposed methods, we conducted experiments on image classification tasks. Specifically, we employed QGNN-IC to quantize DPGN, which uses ResNet [50] as a visual network and includes six generations of graph networks, and subsequently tested its performance on various datasets. Furthermore, to verify the universality of QGNN-IC, we also conducted experiments on different models.

4.1. Experimental Datasets

We followed the evaluation process of previous studies [5,40,51] and evaluated our methods on CIFAR-FS [52], CUB-200-2011 [53], and MiniImageNet [54]. CIFAR-FS is a dataset for few-shot classification, randomly sampled from CIFAR-100 [55]. It consists of 60,000 images of size 32 × 32, with 100 classes and 600 samples per class. CUB-200-2011 contains 11,788 images of size 84 × 84, with 200 classes. MiniImageNet consists of 60,000 images, divided into 100 categories, each with a size of 84 × 84. We adopted the data splits provided by DPGN [5].

4.2. Implementation Details

We adopted the N-way K-shot experimental setting used in DPGN [5]. In the N-way K-shot task, there is a support set and a query set. The support set contains N classes, with K labeled samples per class, while the query set contains several unlabeled samples from

the same N classes. The quantized model should correctly classify the query set based on the support set.

We first pre-trained a full-precision DPGN using the Adam optimizer, with an initial learning rate of 1×10^{-3} and weight decay of 1×10^{-5}. The learning rate is decayed by 0.1 every 15,000 iterations. Then, the pre-trained full-precision model was used to initialize the quantized one. For the 3/4-bit quantization experiments, we used the Adam [56] optimizer with an initial learning rate of 1×10^{-2}. For 2-bit quantization, we used the SGD optimizer with an initial learning rate of 3×10^{-3}. The weight decay and learning rate decay are the same as those used in training the full-precision DPGN. The hyperparameter p in SQD was set to 0.5 to ensure maximum entropy.

4.3. Quantitative Results

In this section, we explore the effectiveness of the proposed QGNN-IC from a quantitative perspective and compare it with state-of-the-art quantization methods on multiple datasets. We re-implement LSQ [23], DoReFa [36], PACT [37], N2UQ [24], EWGS [29], and CMT-KD [31] on DPGN. To ensure a fair comparison, DoReFa does not quantize gradients.

Tables 1 and 2 present the results of the 5-way 1-shot and 5-way 5-shot experiments conducted on the CIFAR-FS dataset, respectively. Except for N2UQ and CMT-KD, previous methods perform poorly. These methods solely focus on improving the distribution of a single visual network without considering the negative impact of outliers. The proposed methods significantly outperform others. Specifically, our approaches achieve an accuracy of 56.44% in the 2-bit 5-way 1-shot experiment, which is 3.11% higher than N2UQ and 1.92% higher than CMT-KD with knowledge distillation.

Table 2. Quantitative comparison of 5-way 5-shot classification accuracy (%) on CIFAR-FS. The accuracy of the full-precision model is 90.18%.

Method	Bit Width		
	2-bit	**3-bit**	**4-bit**
LSQ	36.57	68.26	79.03
DoReFa	31.85	67.25	77.74
PACT	34.61	64.67	78.21
EWGS	36.71	75.29	81.25
N2UQ	40.22	75.43	84.96
CMT-KD	40.92	76.88	85.14
QGNN-IC (ours)	**41.86**	**77.95**	**85.23**

Tables 3 and 4, respectively, display the results of the 5-way 1-shot experiments conducted on the CUB-200-2011 and MiniImageNet datasets. QGNN-IC achieves the best performance on the MiniImageNet dataset. Specifically, in the 2-bit experiment, QGNN-IC attains an accuracy of 39.86%, which is 0.61% higher than CMT-KD. Our methods significantly narrow the performance gap between quantized and full-precision models. In the 4-bit experiment on the CUB-200-2011 dataset, QGNN-IC's performance is slightly below N2UQ but surpasses other methods. However, N2UQ introduces multiple learnable parameters to aid quantization, resulting in higher computational complexity compared to QGNN-IC. Additionally, the self-distillation method proposed in this paper can be applied to N2UQ, further enhancing its performance.

Table 3. Quantitative comparison of 5-way 1-shot classification accuracy (%) on CUB-200-2011. The accuracy of the full-precision model is 75.62%.

Method	Bit Width		
	2-bit	**3-bit**	**4-bit**
LSQ	35.72	55.61	68.13
DoReFa	25.30	53.94	64.42
PACT	23.02	58.93	67.44
EWGS	47.89	70.95	74.25
N2UQ	47.53	71.47	**74.61**
CMT-KD	48.57	71.20	73.42
QGNN-IC (ours)	**49.53**	**71.76**	74.33

Table 4. Quantitative comparison of 5-way 1-shot classification accuracy (%) on MiniImageNet. The accuracy of the full-precision model is 66.58%.

Method	Bit Width		
	2-bit	**3-bit**	**4-bit**
LSQ	39.39	53.75	54.81
DoReFa	23.34	31.36	57.86
PACT	27.07	29.37	59.86
EWGS	37.51	60.01	63.23
N2UQ	38.25	60.96	63.40
CMT-KD	39.25	60.92	64.20
QGNN-IC (ours)	**39.86**	**61.58**	**64.42**

4.4. Ablation Experiments

In this section, we explore the ablation studies conducted on the hyperparameters and different components of QGNN-IC. All experiments were performed on the CIFAR-FS dataset.

Hyperparameters. We present the impact of different hyperparameters, including p in Equation (13) and λ_1 and λ_2 in Equation (19). The optimal values for these parameters, as observed in Figure 5, were found to be $p = 0.5$, $\lambda_1 = 0.7$, and $\lambda_2 = 1.0$. To avoid the complexity of an exhaustive search, we adopted these configurations for all experiments. Although these results may not be the absolute optimum, we have discovered that they already surpass the performance of the majority of existing methods.

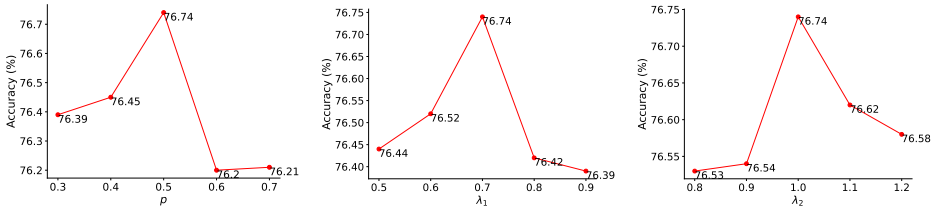


Figure 5. Influence of the hyperparameters on the accuracy of 4-bit QGNN-IC on 5-way 1-shot CIFAR-FS. (**a**) The impact of p on accuracy. (**b**) The impact of λ_1 on accuracy. (**c**) The impact of λ_2 on accuracy.

Components. We further investigated the PQ proposed in Section 3.2, the AQD in Section 3.3, and the SQD in Section 3.4. The experimental results are presented in Table 5. We used EWGS as the baseline. It can be observed that when PQ is used alone, the accuracy improvement over the baseline is 21.72% in the 2-bit 5-way 1-shot experiment. This highlights the importance of removing outliers to reduce quantization errors. Furthermore, when PQ is combined with AQD or SQD, the performance continues to improve. This confirms the feasibility of utilizing the knowledge of the quantized model itself to enhance performance. With all three strategies employed, there is a significant increase in accuracy.

Table 5. Ablation study for QGNN-IC on 5-way 1-shot and 5-way 5-shot CIFAR-FS.

N-Way K-Shot	Method	Bit Width		
		2-bit	**3-bit**	**4-bit**
5-way 1-shot	baseline (EWGS)	31.18	71.19	75.25
	+PQ	52.90	71.44	76.32
	+PQ, AQD	56.02	72.21	76.49
	+PQ, SQD	53.32	71.71	76.21
	+PQ, AQD, SQD	**56.44**	**72.48**	**76.74**
5-way 5-shot	baseline (EWGS)	36.71	75.29	81.25
	+PQ	38.85	76.16	84.16
	+PQ, AQD	41.70	76.84	85.06
	+PQ, SQD	39.92	76.05	84.30
	+PQ, AQD, SQD	**41.86**	**77.95**	**85.23**

4.5. Computational Cost

In this section, we quantitatively evaluate the proposed methods in terms of parameter count and computational cost. We used the bit-floating point operations (Bit-FLOPs) metric as a measure of computational cost [57]. Table 6 demonstrates that during the training phase, knowledge distillation introduces an additional full-precision teacher model to assist in training, resulting in a significant increase in Bit-FLOPs. On the other hand, AQD and SQD are both self-distillation submodules, thus requiring much lower costs compared to knowledge distillation. During the inference phase, the teacher model and additional modules are removed, resulting in the same parameter count and Bit-FLOPs for all methods.

Table 6. Quantitative comparison of parameters and bit-floating point operations (Bit-FLOPs). FP means the 32-bit full-precision models. The others are quantized to 4-bit. KD represents the knowledge distillation.

Method	Training		Inference	
	Param (M)	**Bit-FLOPs (G)**	**Param (M)**	**Bit-FLOPs (G)**
FP	5.1031	503.9534	5.1031	503.9534
PQ	5.1032	503.9677	5.1032	1.0005
PQ + KD	10.2063	984.3012	5.1032	1.0005
PQ + AQD + SQD	5.1212	651.7423	5.1032	1.0005

4.6. Visualization

To provide a more intuitive demonstration of the effectiveness of our methods, we extracted features from CIFAR-FS and utilized t-SNE [58] to visualize the feature embeddings, as shown in Figure 6. The baseline can accurately identify categories, but the clustering

at the feature embedding level is not compact. For example, the yellow, blue, and cyan categories are mixed together. After applying our proposed methods, the quantized models extract rich semantic information and robust visual features with smaller quantization errors. As expected, it can successfully distinguish the yellow, blue, and cyan categories.

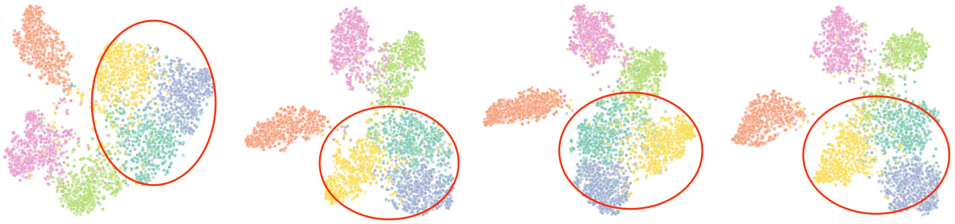


Figure 6. Feature visualization. Different colors represent different categories, and the gradual separation of the feature embedding within the red circle can verify the effectiveness of the proposed methods. (**a**) Baseline (EWGS). (**b**) PQ. (**c**) PQ + AQD. (**d**) PQ + AQD + SQD.

4.7. Generality

In order to investigate the universality of QGNN-IC, we applied the proposed approaches to another GNN, an edge-labeling graph neural network (EGNN) [40]. We quantized the model using the proposed quantization function, PQ, and incorporated AQD into the visual network and SDQ into the graph network.

In Table 7, we present a comparison to evaluate the performance on the MiniImageNet. Compared with the state-of-the-art method CMT-KD, QGNN-IC achieved a 0.61% improvement in the 2-bit experiment. These results demonstrate the general applicability of our methods.

Table 7. Quantitative comparison of quantized edge-labeling graph neural network (EGNN) on 5-way 1-shot classification accuracy (%) on MiniImageNet. The accuracy of the full-precision model is 59.21%.

Method	**Bit Width**		
	2-bit	**3-bit**	**4-bit**
LSQ	30.21	48.26	50.72
DoReFa	31.22	45.23	49.88
PACT	33.74	46.49	51.05
EWGS	35.66	48.91	51.83
N2UQ	36.24	50.26	53.79
CMT-KD	35.85	51.47	54.20
QGNN-IC (ours)	**36.46**	**51.74**	**54.42**

5. Discussion and Conclusions

In this paper, we propose QGNN-IC, a method for quantizing GNNs for visual tasks. It consists of a quantization function, PQ, along with two self-distillation methods, AQD and SQD. PQ eliminates outliers based on data distribution, allowing for fine-grained quantization. SQD and AQD guide the learning of low-bit activations by extracting information from the quantized model itself.

The concept of PQ shares similarities with PAMS [59] and EWGS [29], both of which establish a learnable truncation threshold, thus enabling the model to autonomously determine its own quantization range. Nevertheless, PQ takes into account the distribution discrepancies between visual networks and graph neural networks, initializing the quantization range using mean and standard deviation and reducing quantization errors

by truncating outliers. Experimental outcomes and statistical data reveal that PQ effectively reduces quantization errors by nearly 1.7%, significantly enhancing the accuracy of quantized models.

The essence of AQD and SQD is to utilize modules that contain rich knowledge to assist in training quantized models. This is similar to some previous methods that use knowledge distillation to aid quantization, such as the approach proposed by Zhuang et al. [60], which utilizes full-precision auxiliary modules to assist in training binary networks, and the approach proposed by Xie et al. [45], which uses an attentive transfer module to train quantized models under a knowledge distillation system. However, in comparison, AQD and SQD are self-knowledge distillation modules that fully consider the quantization mechanism and utilize their own implicitly stored full-precision activation values to assist in training the quantized models themselves. They introduce only a small amount of training overhead while significantly enhancing the performance of the quantized models.

This work conducts a focused investigation into quantized graph neural networks applied to image classification tasks. The effectiveness of the proposed approaches has been substantiated on both DPGN and EGNN. However, its suitability for other tasks, such as point cloud classification, remains unexplored. In the future, we will study the performance of our methods in more tasks.

Author Contributions: Conceptualization, X.X., L.M. and T.Z.; methodology, X.X. and L.M.; software, X.X.; validation, X.X. and L.M.; formal analysis, X.X. and L.M.; investigation, X.X.; resources, T.Z.; data curation, Q.H.; writing—original draft preparation, X.X.; writing—review and editing, X.X. and L.M.; visualization, X.X.; supervision, L.M. and T.Z.; project administration, L.M., T.Z. and Q.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Shanghai Municipal Natural Science Foundation under Grant 21ZR1423300.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

$\mathbb{R}$	real space
PTQ	post-training quantization
QAT	quantization-aware training
KD	knowledge distillation
GNN	graph neural network
QGNN-IC	quantized graph neural networks for image classification
PQ	Pauta quantization
QConv	quantized convolutional layer
AQD	attention quantization distillation
SQD	stochastic quantization distillation
DPGN	distribution propagation graph network

References

1. Ben Makhlouf, A.; Farooq, M.U.; Rehman, A.u.; Ibrahim, T.Q.; Hussain, M.; Ali, A.H.; Rashwani, B. Metric dimension of line graphs of bakelite and subdivided bakelite network. *Discret. Dyn. Nat. Soc.* **2023**, *2023*, 7656214. [CrossRef]
2. Waikhom, L.; Patgiri, R. A survey of graph neural networks in various learning paradigms: Methods, applications, and challenges. *Artif. Intell. Rev.* **2023**, *56*, 6295–6364. [CrossRef]
3. Vasudevan, V.; Bassenne, M.; Islam, M.T.; Xing, L. Image classification using graph neural network and multiscale wavelet superpixels. *Pattern Recognit. Lett.* **2023**, *166*, 89–96. [CrossRef]
4. Fei, Z.; Guo, J.; Gong, H.; Ye, L.; Attahi, E.; Huang, B. A GNN architecture with local and global-attention feature for image classification. *IEEE Access* **2023**, *11*, 110221–110233. [CrossRef]

5. Yang, L.; Li, L.; Zhang, Z.; Zhou, X.; Zhou, E.; Liu, Y. Dpgn: Distribution propagation graph network for few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 13390–13399. [CrossRef]
6. Tang, Z.; Liu, Y.; Shang, Y. A new GNN-based object detection method for multiple small objects in aerial images. In Proceedings of the 2023 IEEE/ACIS 23rd International Conference on Computer and Information Science (ICIS), Wuxi, China, 23–25 June 2023; pp. 14–19. [CrossRef]
7. Zhao, G.; Ge, W.; Yu, Y. GraphFPN: Graph feature pyramid network for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 2763–2772. [CrossRef]
8. Shi, P.; Guo, X.; Yang, Y.; Ye, C.; Ma, T. NexToU: Efficient topology-aware U-Net for medical image segmentation. *arXiv* **2023**, arXiv:2305.15911.
9. Xie, G.S.; Liu, J.; Xiong, H.; Shao, L. Scale-aware graph neural network for few-shot semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 5475–5484. [CrossRef]
10. Zhang, S.; Sohrabizadeh, A.; Wan, C.; Huang, Z.; Hu, Z.; Wang, Y.; Li, Y.; Cong, J.; Sun, Y. A survey on graph neural network acceleration: Algorithms, systems, and customized hardware. *arXiv* **2023**, arXiv:2306.14052.
11. Chang, J.; Lu, Y.; Xue, P.; Xu, Y.; Wei, Z. Global balanced iterative pruning for efficient convolutional neural networks. *Neural Comput. Appl.* **2022**, *34*, 21119–21138. [CrossRef]
12. Yu, S.; Nguyen, P.; Anwar, A.; Jannesari, A. Heterogeneous federated learning using dynamic model pruning and adaptive gradient. In Proceedings of the 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Bangalore, India, 1–4 May 2023; pp. 322–330. [CrossRef]
13. Liu, C.; Ma, X.; Zhan, Y.; Ding, L.; Tao, D.; Du, B.; Hu, W.; Mandic, D.P. Comprehensive graph gradual pruning for sparse training in graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–15. [CrossRef]
14. Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M.W.; Keutzer, K. A survey of quantization methods for efficient neural network inference. *arXiv* **2021**, arXiv:2103.13630.
15. Wang, Y.; Han, Y.; Wang, C.; Song, S.; Tian, Q.; Huang, G. Computation-efficient deep learning for computer vision: A survey. *arXiv* **2023**, arXiv:2308.13998.
16. Shang, Y.; Yuan, Z.; Xie, B.; Wu, B.; Yan, Y. Post-training quantization on diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 1972–1981. [CrossRef]
17. Qin, H.; Ma, X.; Ding, Y.; Li, X.; Zhang, Y.; Ma, Z.; Wang, J.; Luo, J.; Liu, X. BiFSMNv2: Pushing binary neural networks for keyword spotting to real-network performance. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–13. [CrossRef] [PubMed]
18. Li, S.; Hanson, E.; Qian, X.; Li, H.H.; Chen, Y. ESCALATE: Boosting the efficiency of sparse CNN accelerator with kernel decomposition. In Proceedings of the 54th Annual IEEE/ACM International Symposium on Microarchitecture, New York, NY, USA, 18–22 October 2021; pp. 992–1004. [CrossRef]
19. Chen, J.A.; Sung, H.H.; Shen, X.; Choudhury, S.; Li, A. BitGNN: Unleashing the performance potential of binary graph neural networks on GPUs. In Proceedings of the 37th International Conference on Supercomputing, New York, NY, USA, 21–23 June 2023; pp. 264–276. [CrossRef]
20. Shen, H.; Meng, H.; Dong, B.; Wang, Z.; Zafrir, O.; Ding, Y.; Luo, Y.; Chang, H.; Gao, Q.; Wang, Z.; et al. An efficient sparse inference software accelerator for transformer-based language models on CPUs. *arXiv* **2023**, arXiv:2306.16601.
21. Gu, J.; Zhu, Y.; Wang, P.; Chadha, M.; Gerndt, M. FaST-GShare: Enabling efficient spatio-temporal GPU sharing in serverless computing for deep learning inference. In Proceedings of the 52nd International Conference on Parallel Processing, ICPP '23, New York, NY, USA, 7–10 August 2023; pp. 635–644. [CrossRef]
22. Qian, B.; Wang, Y.; Hong, R.; Wang, M. Adaptive data-free quantization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 7960–7968. [CrossRef]
23. Esser, S.K.; McKinstry, J.L.; Bablani, D.; Appuswamy, R.; Modha, D.S. Learned step size quantization. *arXiv* **2019**, arXiv:1902.08153.
24. Liu, Z.; Cheng, K.T.; Huang, D.; Xing, E.P.; Shen, Z. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 4942–4952. [CrossRef]
25. Wan, B.; Zhao, J.; Wu, C. Adaptive message quantization and parallelization for distributed full-graph GNN training. *arXiv* **2019**, arXiv:2306.01381.
26. Eliasof, M.; Bodner, B.J.; Treister, E. Haar wavelet feature compression for quantized graph convolutional networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–12. [CrossRef] [PubMed]
27. Zhong, K.; Zeng, S.; Hou, W.; Dai, G.; Zhu, Z.; Zhang, X.; Xiao, S.; Yang, H.; Wang, Y. CoGNN: An algorithm-hardware co-design approach to accelerate GNN inference with mini-batch sampling. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 4883–4896. [CrossRef]
28. Sharma, P.K.; Abraham, A.; Rajendiran, V.N. A generalized zero-shot quantization of deep convolutional neural networks via learned weights statistics. *IEEE Trans. Multimed.* **2023**, *25*, 953–965. [CrossRef]
29. Lee, J.; Kim, D.; Ham, B. Network quantization with element-wise gradient scaling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 6448–6457. [CrossRef]

30. Pei, Z.; Yao, X.; Zhao, W.; Yu, B. Quantization via distillation and contrastive learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–13. [CrossRef]
31. Pham, C.; Hoang, T.; Do, T.T. Collaborative Multi-Teacher Knowledge Distillation for Learning Low Bit-width Deep Neural Networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 6435–6443. [CrossRef]
32. Zhu, K.; He, Y.Y.; Wu, J. Quantized feature distillation for network quantization. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, Washington, DC, USA, 7–14 February 2023. [CrossRef]
33. Liu, J.; Niu, L.; Yuan, Z.; Yang, D.; Wang, X.; Liu, W. PD-Quant: Post-training quantization based on prediction difference metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 24427–24437. [CrossRef]
34. Wei, X.; Gong, R.; Li, Y.; Liu, X.; Yu, F. QDrop: Randomly dropping quantization for extremely low-bit post-training quantization. *arXiv* **2022**, arXiv:2203.05740.
35. Lin, C.; Peng, B.; Li, Z.; Tan, W.; Ren, Y.; Xiao, J.; Pu, S. Bit-shrinking: Limiting instantaneous sharpness for improving post-training quantization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 16196–16205. [CrossRef]
36. Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; Zou, Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv* **2016**, arXiv:1606.06160.
37. Choi, J.; Wang, Z.; Venkataramani, S.; Chuang, P.I.J.; Srinivasan, V.; Gopalakrishnan, K. Pact: Parameterized clipping activation for quantized neural networks. *arXiv* **2018**, arXiv:1805.06085.
38. Nagel, M.; Fournarakis, M.; Amjad, R.A.; Bondarenko, Y.; Van Baalen, M.; Blankevoort, T. A white paper on neural network quantization. *arXiv* **2021**, arXiv:2106.08295.
39. Gao, C.; Wang, X.; He, X.; Li, Y. Graph neural networks for recommender system. In Proceedings of the the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM), New York, NY, USA, 21–25 February 2022; pp. 1623–1625. [CrossRef]
40. Kim, J.; Kim, T.; Kim, S.; Yoo, C.D. Edge-labeling graph neural network for few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 11–20. [CrossRef]
41. Zhang, R.; Yang, S.; Zhang, Q.; Xu, L.; He, Y.; Zhang, F. Graph-based few-shot learning with transformed feature propagation and optimal class allocation. *Neurocomputing* **2022**, *470*, 247–256. [CrossRef]
42. Zhang, Y.; Li, W.; Zhang, M.; Wang, S.; Tao, R.; Du, Q. Graph information aggregation cross-domain few-shot learning for hyperspectral image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–14. [CrossRef] [PubMed]
43. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
44. Mishra, A.; Marr, D. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv* **2017**, arXiv:1711.05852.
45. Xie, Z.; Wen, Z.; Liu, J.; Liu, Z.; Wu, X.; Tan, M. Deep transferring quantization. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 625–642. [CrossRef]
46. Martinez, B.; Yang, J.; Bulat, A.; Tzimiropoulos, G. Training binary neural networks with real-to-binary convolutions. *arXiv* **2020**, arXiv:2003.11535.
47. Bengio, Y.; Léonard, N.; Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv* **2013**, arXiv:1308.3432.
48. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713. [CrossRef]
49. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
50. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
51. Rusu, A.A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; Hadsell, R. Meta-learning with latent embedding optimization. *arXiv* **2018**, arXiv:1807.05960.
52. Bertinetto, L.; Henriques, J.F.; Torr, P.H.; Vedaldi, A. Meta-learning with differentiable closed-form solvers. *arXiv* **2018**, arXiv:1805.08136.
53. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *The Caltech-Ucsd Birds-200-2011 Dataset*; Technical Report, CNS-TR-2011-001; California Institute of Technology: Pasadena, CA, USA, 2011.
54. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3637–3645.
55. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, Department of Computer Science, University of Toronto, Toronto, ON, Canada, 2009.
56. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

57. Liu, Z.; Wang, Y.; Han, K.; Ma, S.; Gao, W. Instance-aware dynamic neural network quantization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12424–12433. [CrossRef]
58. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 11.
59. Li, H.; Yan, C.; Lin, S.; Zheng, X.; Zhang, B.; Yang, F.; Ji, R. PAMS: Quantized super-resolution via parameterized max scale. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 564–580. [CrossRef]
60. Zhuang, B.; Liu, L.; Tan, M.; Shen, C.; Reid, I. Training quantized neural networks with a full-precision auxiliary module. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1488–1497. [CrossRef]