electronics

MDPI

Article

AE-Qdrop: Towards Accurate and Efficient Low-Bit Post-Training Quantization for A Convolutional Neural Network

Jixing Li [1,2,3], Gang Chen [1,2,3,*], Min Jin [1,2,3], Wenyu Mao [1,2,3] and Huaxiang Lu [1,2,3]

[1] Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China; lijixing0425@semi.ac.cn (J.L.)
[2] University of Chinese Academy of Sciences, Beijing 100089, China
[3] Beijing Key Lab of Semiconductor Neural Network Intelligent Perception and Computing Technology, Beijing 100083, China
* Correspondence: chengang08@semi.ac.cn

Abstract: Blockwise reconstruction with adaptive rounding helps achieve acceptable 4-bit post-training quantization accuracy. However, adaptive rounding is time intensive, and the optimization space of weight elements is constrained to a binary set, thus limiting the performance of quantized models. The optimality of block-wise reconstruction requires that subsequent network blocks remain unquantized. To address this, we propose a two-stage post-training quantization scheme, AE-Qdrop, encompassing block-wise reconstruction and global fine-tuning. In the block-wise reconstruction stage, a progressive optimization strategy is introduced as a replacement for adaptive rounding, enhancing both quantization accuracy and efficiency. Additionally, the integration of randomly weighted quantized activation helps mitigate the risk of overfitting. In the global fine-tuning stage, the weights of each quantized network block are corrected simultaneously through logit matching and feature matching. Experiments in image classification and object detection tasks validate that AE-Qdrop achieves high precision and efficient quantization. For the 2-bit MobileNetV2, AE-Qdrop outperforms Qdrop in quantization accuracy by 6.26%, and its quantization efficiency is fivefold higher.

Keywords: post-training quantization; adaptive rounding; block-wise reconstruction; progressive optimization strategy; randomly weighted quantized activation; global fine-tuning

Citation: Li, J.; Chen, G.; Jin, M.; Mao, W.; Lu, H. AE-Qdrop: Towards Accurate and Efficient Low-Bit Post-Training Quantization for A Convolutional Neural Network. *Electronics* **2024**, *13*, 644. https://doi.org/10.3390/electronics13030644

Academic Editor: Luca Patanè

Received: 9 December 2023
Revised: 22 January 2024
Accepted: 1 February 2024
Published: 4 February 2024

1. Introduction

In recent years, Convolutional Neural Networks (CNNs) have produced remarkable performance across a variety of computer vision tasks. However, the extensive parameters and high computational complexity associated with CNNs present challenges for devices in terms of storage, power consumption, and computational capabilities. In resource-constrained mobile applications, such as intelligent wearable devices, unmanned aerial vehicles, and smart robots, CNNs quickly deplete storage, memory, battery, and computational resources. Therefore, reduction in the number of parameters and the computational complexity of CNNs is a crucial objective for their deployment in mobile applications.

Researchers have proposed a range of CNN compression and acceleration techniques, which include knowledge distillation [1,2], neural network architecture search [3,4], pruning [5,6], and quantization [7]. Knowledge distillation uses a large model as a 'teacher' to guide the training of a smaller 'student' model, enabling the smaller model to assimilate the knowledge contained in the larger model. For example, Anfu Zhu [1] achieved an 81.4% reduction in the number of parameters of the student model compared to the teacher model by utilizing multi-dimensional knowledge distillation, increasing accuracy by 2.5% over training the student model directly. Neural architecture search is a technique that employs automated methods to discover optimal neural network structures, effectively balancing network accuracy and efficiency. Recently, MANAS [3] approached

neural architecture search as a multi-agent problem and achieved a recognition rate of 74.74% on the ImageNet2012 dataset (https://image-net.org/challenges/LSVRC/2012/2012-downloads.php, accessed on 31 January 2024), with a network parameter size of only 2.6 MB. Weiguo Shen [6] applied pruning technology to the mixed signal identification network, removing 83.2% of the redundant weights in the network, and the accuracy only dropped by 3.72%. Knowledge distillation, neural architecture search, and pruning all involve adjusting network structures and retraining, making their time cost expensive. Quantization converts floating-point (FP) network parameters to lower-bit parameters without altering the network structure. This reduction in data bit-width directly decreases power consumption and storage requirements and improves computational speed. For example, INT8-based quantized models deliver 3.3× and 4× better performance over FP32 using OpenVINO on Intel CPU and TFLite on Raspberry Pi device, respectively, for the MLPerf offline scenario [8]. Therefore, quantization is an exceptionally effective technique for model compression and acceleration.

Due to the introduction of quantization noise through rounding and truncation operations in quantization computations, the accuracy of quantized models often experiences some degree of degradation. To mitigate precision loss, current quantization techniques are primarily categorized into Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). QAT relies on complete training data and labels, retraining network weights and quantization parameters through backpropagation. Research in QAT primarily focuses on gradient estimation [9–11]; optimization strategies [12–14]; binary networks [15–17]; quantization distillation [18–20]; etc. Although QAT can achieve higher quantization accuracy, it is not the primary choice for neural network quantization deployment due to its high time costs and close relation to specific tasks. Different tasks exhibit significant differences in model structure, loss functions, and training strategies. Therefore, QAT requires an in-depth understanding of the model's training details, increasing the manpower costs of model deployment (considering that model training and quantization may be handled by different personnel). In contrast, PTQ achieves network quantization based on a small sample (calibration dataset) without the need for retraining, making it more favored in the industry. However, PTQ faces a significant loss of accuracy. Through early research efforts such as optimizing quantization factor scale [21,22], PTQ can achieve nearly lossless 8-bit precision. Consequently, recent studies [23–25] have predominantly focused on low-bit quantization ($\leq$4 bits). In these studies, Qdrop [26] treats each network block as a fundamental unit and minimizes the output difference between floating-point and quantized network blocks (block-wise reconstruction) by applying adaptive rounding and randomly dropout quantized activation schemes, achieving acceptable 4-bit quantization accuracy.

However, Qdrop still harbors certain drawbacks stemming from adaptive rounding [24] and block-wise reconstruction [25]. The adaptive rounding technique confines the optimization space of weight elements to a binary set. As the bit-width of activation quantization reduces, the perturbation to weights due to activation quantization noise progressively intensifies. This restricted optimization space may not yield the optimal solution, thereby limiting the precision of quantization. More crucially, the adaptive rounding technique necessitates numerous iterative cycles to gradually weaken (through a linear annealing mechanism) the regularization constraints of rounding, which significantly diminishes quantization efficiency. For instance, early post-training quantization techniques such as MSE [21] only require approximately 3 min to complete the quantization of MobileNetV2, often deployed on mobile devices, whereas Qdrop demands about 40 min (running on the NVIDIA GeForce RTX 3090 GPU and Intel(R) Core(TM) i7-7700K CPU). Regarding block-wise reconstruction, its optimality relies on the assumption that subsequent network blocks are not quantized, which contradicts the reality where all network blocks are intended to be quantized. Consequently, there is still potential for further improvement in the accuracy of the quantized models.

Based on the above observations, we aim to achieve high-precision and efficient low-bit quantization, which prompts the proposal of AE-Qdrop. AE-Qdrop consists of block-wise

reconstruction and global fine-tuning. Considering the constraints of adaptive rounding on the optimization space and its time-intensive nature, we propose a progressive optimization strategy in block-wise reconstruction. By gradually constraining the optimization space, higher quantization accuracy and efficiency are to be achieved. A randomly weighted quantized activation scheme is to be introduced so as to reduce the risk of overfitting. To address the shortcomings of block-wise reconstruction, global fine-tuning synchronously corrects all weights through logit matching and feature matching, further improving quantization accuracy. Our contributions can be summarized as follows:

- We perform a theoretical analysis of the shortcomings associated with adaptive rounding and block-wise reconstruction.
- We introduce AE-Qdrop, a two-stage algorithm that includes block-wise reconstruction and global fine-tuning. AE-Qdrop combines a progressive optimization strategy with randomly weighted quantization activation, enhancing the accuracy and efficiency of block-wise reconstruction. Subsequently, global fine-tuning is applied to further optimize the weights, thereby improving the overall quantization accuracy.
- Extensive experiments are conducted to evaluate the quantization results of mainstream networks, demonstrating the excellent performance of AE-Qdrop in quantization accuracy and quantization efficiency.

The subsequent content of this paper is organized as follows: Section 2 introduces related studies on PTQ. In Section 3, we introduce the linear quantization and adaptive rounding technique, then elucidate the limitations of adaptive rounding and block-wise reconstruction through theoretical analysis. Section 4 expounds on the specific details of AE-Qdrop. Section 5 delves into an in-depth analysis and discussion of the experimental results. Ultimately, the conclusions are presented in Section 6.

2. Related Work

PTQ is based on a small number of calibration samples to achieve network quantization. Compared to QAT, it is more efficient and convenient, which offers it brighter prospects for application in the industry. Table 1 summarizes some classical post-training quantization schemes. Early PTQ focused on minimizing the quantization error of network parameters through techniques such as optimizing quantization factor scale [21,22], bias correction [27,28], piecewise linear quantization [29,30], and outlier separation [31,32]. For example, Nvidia's TensorRT [22], a widely used quantization tool, searched for the optimal quantization factor scale by minimizing the Kullback–Leibler (KL) distance between FP activation and quantized activation. Although these PTQ schemes achieved near-lossless 8-bit quantization accuracy, they faced challenges at low bit-widths (such as 4 bit). This limitation can be attributed to the fact that minimizing the quantization error of network parameters does not necessarily result in optimal quantization.

Subsequently, LAPQ [23] proposed optimizing the quantization scale factor by minimizing the loss function. Inspired by this, AdaRound [24] designed an adaptive rounding technique to minimize the output activation difference between the quantized network layer and the FP network layer (layer-wise reconstruction), which significantly enhanced low-bit quantization accuracy. Adaptive rounding was subsequently widely adopted in post-training quantization. Among these developments, researchers using BrecQ [25] argued that layer-wise reconstruction ignored dependencies between network layers. It considered block-wise reconstruction as the optimal optimization target and employed the Fisher matrix to approximate the Hessian matrix. Building upon BrecQ, RAPQ [33] improved Power-of-Two low-bit quantization accuracy through Power-of-Two Scale Group and BN-based L-P Loss. Using Mr.BiQ [34], researchers explored the minimization of reconstruction error using nonlinear quantizers. Furthermore, scientists using Qdrop [26] pointed out that BrecQ overlooked the impact of activation quantization in weight optimization and demonstrated that activation quantization noise could be translated into weight perturbation. The algorithm designed a randomly dropout quantized activation strategy in block-wise reconstruction, achieving acceptable 4-bit quantization accuracy.

However, the limitation of adaptive rounding on the weight optimization space and the neglect of inter-block dependencies in block-wise reconstruction limited the quantization accuracy of Qdrop, and the time-intensive nature of adaptive rounding led to low quantization efficiency. Therefore, our goal is to achieve accurate and efficient low-bit quantization. AdaQuant [35] directly optimized quantized weights through the Straight-Through Estimator (STE). The optimization process was efficient, but the STE faced serious gradient mismatch problem at low bit-widths; in addition, an insufficient number of calibration samples can easily lead to overfitting. The proposed progressive optimization strategy in AE-Qdrop gradually confines the optimization space of weight values from arbitrary values to a binary set. This approach enhances quantization efficiency while, to some extent, mitigating issues such as gradient mismatch and overfitting. Although BrecQ assumes that network-wise reconstruction is an ideal optimization goal, constrained by the limited number of calibration samples, the quantization model based on network-wise reconstruction had poor generalization performance. AE-Qdrop effectively avoids this problem through a two-stage process of block-wise reconstruction followed by global fine-tuning. In addition, global fine-tuning is also efficient, and the formal derivation of the optimal quantization for network blocks provides a more rigorous explanation for the optimization objective of global fine-tuning.

Table 1. Overview of the classic PTQ.

Bit-Width	Optimization Goal	Related Work
≥6 bit	The Quantization Error of Network Parameters	Optimizing Quantization Factor Scale [21,22] Bias Correction [27,28] Piecewise Linear Quantization [29,30] Outlier Separation [31,32]
≤4 bit	Layer-wise Reconstruction	LAPQ [23] AdaRound [24] AdaQuant [35]
	Block-wise Reconstruction	BrecQ [25] RAPQ [33] Mr.BiQ [34] Qdrop [26]

3. Background and Theoretical Analysis

3.1. Quantizer

For a FP tensor (activation or weight) x, we can map it to an integer tensor q according to the following equation [36]:

$$\begin{aligned} q &= \text{clip}\left(\text{round}\left(\frac{x}{s}\right) + z, q_{\min}, q_{\max}\right), \\ s &= \frac{x_{\max} - x_{\min}}{2^b - 1}, \\ z &= q_{\min} - \text{round}\left(\frac{x_{\min}}{s}\right), \\ x_q &= s(q - z). \end{aligned} \tag{1}$$

clip$(\cdot)$ and round$(\cdot)$ denote truncation and rounding operations, respectively. Variable s represents the quantization scaling factor, indicating the proportional relationship between FP values and integers. Variable z is defined as the offset corresponding to the zero point. The maximum and minimum values in the vector are denoted by $x_{\max}$ and $x_{\min}$, respectively. The quantization range, specified by $[q_{\min}, q_{\max}]$, is determined by bit-width b. This paper focuses solely on uniform unsigned symmetric quantization, the most common quantization setting, where $q_{\min}$ equals 0 and $q_{\max}$ is $2^b - 1$. Non-linear quantization is

not considered due to its challenges in hardware deployment. x_q refers to the FP tensor, also known as the fake-quantized tensor. In the FP domain, the quantizer discretizes continuous FP values into 2^b distinct values. The difference between x_q and x is defined as the parameter quantization error.

3.2. AdaRound

AdaRound [24] reexamines the effect of weight quantization on the loss function using a second-order Taylor expansion. For the lth layer of the network $f_l(\cdot)$, the change in the loss function $L(\cdot)$ due to weight quantization is defined as

$$\begin{aligned}
&L\Big(f_n(f_{n-1}\dots f_l(x_f, w_q))\Big) - L\Big(f_n(f_{n-1}\dots f_l(x_f, w_f))\Big)\\
=&\mathcal{L}\Big(x_f, w_f + \Delta w\Big) - \mathcal{L}\Big(x_f, w_f\Big)\\
\approx&\Delta w^T \mathbf{g}_w\Big(x_f, w_f\Big) + \frac{1}{2}\Delta w^T \mathbf{H}_w\Big(x_f, w_f\Big)\Delta w\\
\approx&\frac{1}{2}\Delta w^T \mathbf{J}_{y:w}^T\Big(x_f, w_f\Big)\mathbf{H}_y\Big(x_f, w_f\Big)\mathbf{J}_{y:w}\Big(x_f, w_f\Big)\Delta w\\
=&\frac{1}{2}\Delta y^T \mathbf{H}_y\Big(x_f, w_f\Big)\Delta y\\
\approx&\frac{1}{2}\Delta y^T \mathbf{I}\Delta y\\
=&\frac{1}{2}\|\Delta y\|_2^2,
\end{aligned} \tag{2}$$

where $\mathcal{L}(\cdot)$ is equal to $L(f_n(f_{n-1}\dots f_l(\cdot)))$, $\mathbf{g}_w\Big(x_f, w_f\Big)$ represents the gradient matrix and $\mathbf{H}_w\Big(x_f, w_f\Big)$ represents the Hessian matrix. Considering that the FP model has converged, $\mathbf{g}_w\Big(x_f, w_f\Big)$ is approximately zero. By introducing network layer output y and the Jacobian matrix, $\mathbf{J}_{y:w}\Big(x_f, w_f\Big)$ of y with respect to w, $\mathbf{H}_w\Big(x_f, w_f\Big)$ can be decomposed into $\mathbf{J}_{y:w}^T\Big(x_f, w_f\Big)\mathbf{H}_y\Big(x_f, w_f\Big)\mathbf{J}_{y:w}\Big(x_f, w_f\Big)$ [25]. $\Delta y = \mathbf{J}_{y:w}\Big(x_f, w_f\Big)\Delta w = y - y_q$ is the difference in network layer output before and after weight quantization. By approximating the Hessian matrix, $\mathbf{H}_w\Big(x_f, w_f\Big)$ with the identity matrix, $\mathbf{I}$, the change in the loss function caused by weight quantization is approximately equal to the change in the output of the network layer. Therefore, the optimization goal of weight quantization can be defined as layer-wise reconstruction:

$$\min_{\Delta w} \mathcal{L}\Big(x_f, w_f + \Delta w\Big) - \mathcal{L}\Big(x_f, w_f\Big) \to \min_{\Delta w} \|\Delta y\|_2^2. \tag{3}$$

AdaRound introduces the trainable tensor v, which has the same dimension as w, into the weight quantizer to indirectly optimize Δw:

$$\begin{aligned}
q_w &= \text{clip}\left(\text{floor}\left(\frac{w_f}{s_w}\right) + h(v) + z, q_{\min}, q_{\max}\right),\\
h(v) &= \text{clip}\left(\frac{1.2}{1+\exp(-v)} - 0.1, 0, 1\right),\\
\Delta w &= w_f - s_w(q_w - z).
\end{aligned} \tag{4}$$

This scheme is called adaptive rounding. To ensure that $h(v)$ converges to zero or one, AdaRound introduces a regularization term for Equation (3). Therefore, the final optimization goal is

$$\min_v \|\Delta y\|_2^2 + \lambda \sum (1 - |2h(v) - 1|)^\beta, \tag{5}$$

where β and λ are parameters governing the regularization. The value of β is adjusted based on linear annealing. In the initial stages, the value of β is higher, which weakens

the constraint of the regularization term, facilitating the reduction in the reconstruction loss, $\|\Delta y\|_2^2$. In the later stages, the value of β is lower, enhancing the constraint of the regularization term, encouraging $h(v)$ to converge to zero or one. If w_f contains M elements, then adaptive rounding provides Δw with a solution space of size 2^M. Nearest neighbor rounding is only one set of solutions, but there may be better solutions that minimize Equation (3). Therefore, adaptive rounding can effectively improve quantization accuracy.

3.3. Drawbacks of Adaptive Rounding

Considering that both weight and activation are quantized simultaneously, Equation (2) can be generalized as follows:

$$\begin{aligned}
&\mathbf{L}\left(f_n(f_{n-1}\dots f_l(x_q,w_q)))\right)-\mathbf{L}\Big(f_n(f_{n-1}\dots f_l(x_f,w_f)))\Big)\\
&=\mathcal{L}\Big(x_f+\Delta x,w_f+\Delta w\Big)-\mathcal{L}\Big(x_f,w_f\Big)\\
&\approx\Delta x^T\mathbf{g}_x\Big(x_f,w_f\Big)+\tfrac{1}{2}\Delta x^T\mathbf{H}_x\Big(x_f,w_f\Big)\Delta w+\\
&\Delta w^T\mathbf{g}_w\Big(x_f,w_f\Big)+\tfrac{1}{2}\Delta w^T\mathbf{H}_w\Big(x_f,w_f\Big)\Delta w+\Delta x^T\mathbf{H}_{xw}\Big(x_f,w_f\Big)\Delta w\\
&\approx\tfrac{1}{2}\Delta x^T\mathbf{H}_x\Big(x_f,w_f\Big)\Delta x+\tfrac{1}{2}\Delta w^T\mathbf{H}_w\Big(x_f,w_f\Big)\Delta w+\Delta x^T\mathbf{H}_{xw}\Big(x_f,w_f\Big)\Delta w\\
&=\tfrac{1}{2}\Delta x^T\mathbf{J}_{y:x}^T\Big(x_f,w_f\Big)\mathbf{H}_y\Big(x_f,w_f\Big)\mathbf{J}_{y:x}\Big(x_f,w_f\Big)\Delta x+\\
&\tfrac{1}{2}\Delta w^T\mathbf{J}_{y:w}^T\Big(x_f,w_f\Big)\mathbf{H}_y\Big(x_f,w_f\Big)\mathbf{J}_{y:w}\Big(x_f,w_f\Big)\Delta w+\\
&\Delta x^T\mathbf{J}_{y:x}^T\Big(x_f,w_f\Big)\mathbf{H}_y\Big(x_f,w_f\Big)\mathbf{J}_{y:w}\Big(x_f,w_f\Big)\Delta w\\
&=2\cdot\Delta y^T\mathbf{H}_y\Big(x_f,w_f\Big)\Delta y\\
&\approx 2\cdot\Delta y^T\mathbf{I}\Delta y.
\end{aligned}\tag{6}$$

Qdrop demonstrates that activation quantization noise can be converted into weight perturbation $\tau(x)$:

$$\begin{aligned}
&\mathcal{L}\Big(x_f+\Delta x,w_f+\Delta w\Big)-\mathcal{L}\Big(x_f,w_f\Big)\\
&=\mathcal{L}\Big(x_f,(w_f+\Delta w)\odot(1+\tau(x)\Big)-\mathcal{L}\Big(x_f,w_f\Big)\\
&=\mathcal{L}\Big(x_f,w_f+\Delta w+w_f\odot\tau(x)+\Delta w\odot\tau(x)\Big)-\mathcal{L}\Big(x_f,w_f\Big).
\end{aligned}\tag{7}$$

Considering $f_l(\cdot)$ as the lth network block, Equation (6) can be generalized for block-wise reconstruction, as derived in [25]. The optimization goal for block-wise reconstruction can be derived based on Equations (6) and (7):

$$\min_{\Delta w_f}\mathcal{L}\left(x_f,w_f+\underbrace{\Delta w+w_f\odot\tau(x)+\Delta w\odot\tau(x)}_{(\Delta w_f)}\right)-\mathcal{L}\Big(x_f,w_f\Big)\rightarrow\min_{\Delta w_f}\|\Delta y\|_2^2.\tag{8}$$

In Qdrop, the optimization of Equation (8) is achieved through adaptive rounding. However, adaptive rounding constrains the optimization space of each element in w_q to a binary set. As the activation quantization bit-width decreases, weight perturbation $\tau(x)$ incrementally intensifies, making the variation of Δw_f more complex. This escalation can lead to a situation where the binary set may not provide the optimal Δw_f, potentially compromising the quantization performance. To illustrate this point, we consider the example depicted in Figure 1. In this scenario, FP output y_f equals 0.98. While the best quantized output, y_q^{ada}, achievable through adaptive rounding stands at 0.85, the actual ideal quantized output, y_q^{best}, is found to be 0.95. The linear annealing mechanism of parameter β results in adaptive rounding requiring a significant number of iterative cycles, which considerably increases the time cost of network quantization. Lastly, an adversarial

relationship exists between the regularization term and the reconstruction loss. This antagonism can impede the efficient optimization of the reconstruction error.

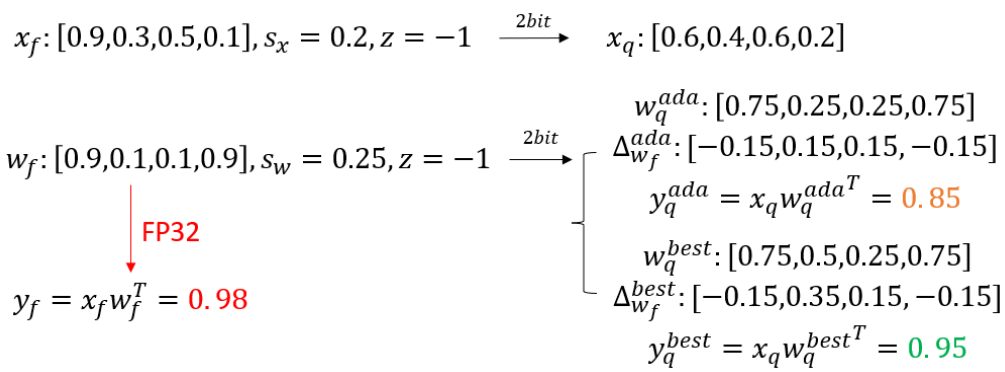

Figure 1. A simple calculation example illustrates that adaptive rounding cannot provide an optimal solution.

3.4. Drawbacks of Block-Wise Reconstruction

The derivation of Equation (6) actually relies on the assumption that the network blocks from the lth to the nth are not quantized. If all network blocks are quantized, then the optimization objective for the lth network block can be transformed into

$$\min_{\Delta w_f} \mathbf{L}\left(f_n^q(f_{n-1}^q \ldots f_l(x_q, w_q))\right) - \mathbf{L}\left(f_n(f_{n-1} \ldots f_l(x_f, w_f))\right). \tag{9}$$

It is evident that the optimal quantization of the l^{th} network block is correlated with subsequent quantized network blocks. The applicability of the optimal solution for Equations (8)–(13) depends on the discrepancy between $f_n^q(f^q n-1 \ldots f_l(\cdot))$ and $f_n(f_{n-1} \ldots f_l(\cdot))$. Consequently, block-wise quantization disregards the influence of subsequent quantized network blocks, potentially leading to suboptimal quantization of each network block.

4. AE-Qdrop

To address the issues with adaptive rounding and block-wise reconstruction, we propose a two-stage post-training quantization algorithm, AE-Qdrop. In the block-wise reconstruction phase (as shown in Figure 2), a progressive optimization strategy is designed to replace adaptive rounding. This strategy offers a larger space for weight optimization and higher optimization efficiency. Additionally, randomly weighted quantized activation is introduced to enhance the diversity of activations, effectively improving the generalization performance of the quantized model. Block-wise reconstruction provides a pre-quantized model for global fine-tuning. Global fine-tuning (as shown in Figure 3) aims to correct suboptimal weights caused by block-wise reconstruction through feature matching and logit matching. It should be emphasized that, similar to network-wise reconstruction, global fine-tuning cannot achieve high-precision quantization directly using only a small number of unlabeled samples. Therefore, the block-wise reconstruction process is crucial and indispensable, which is the rationale behind adopting a two-stage quantization design.

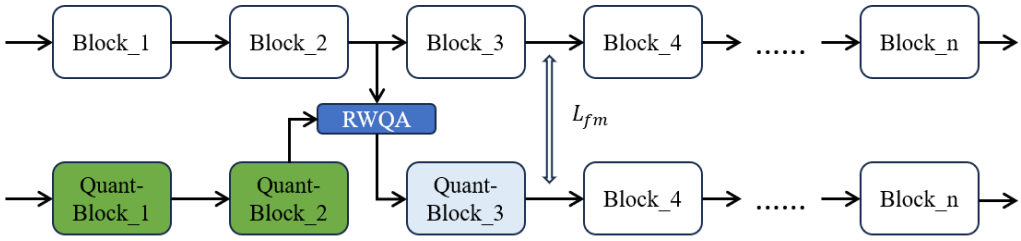

Figure 2. The block-wise reconstruction stage of AE-Qdrop.

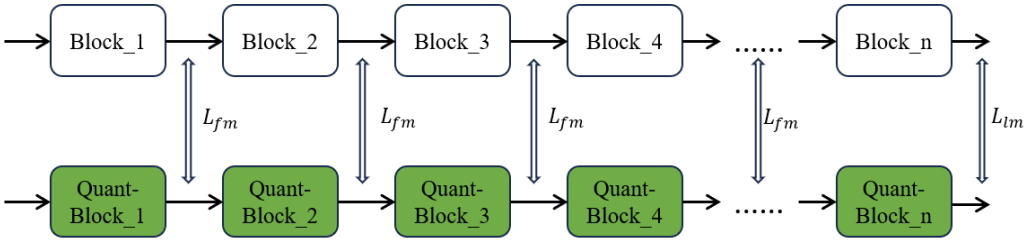

Figure 3. The global fine-tuning stage of AE-Qdrop.

4.1. Block-Wise Reconstruction: Progressive Optimization Strategy

AE-Qdrop indirectly optimizes Δw_f by adjusting w_f. However, there are some challenges in directly using STE to optimize weights, as Adaquant [35] does. On the one hand, the gradient mismatch problem of the straight-through estimator becomes prominent when weights and activations are quantized to low bit-widths, potentially causing confusion in the optimization direction. On the other hand, a limited number of calibration samples can easily lead to overfitting. To address these challenges, we propose a progressive optimization strategy (POS).

At first, we quantize the activation but do not quantize the weight. Weight w_f can be updated along the correct gradient direction since the STE is not required. It is worth noting that the updated weight can absorb the weight perturbation generated by activation quantization, i.e., $w_f \leftarrow w_f \odot (1 + \tau(x))$.

Next, the new weight is quantized. Considering that the previous optimization has significantly reduced weight perturbation, Equation (8) is close to Equation (3). Therefore, we only optimize the weight rounding direction to avoid overfitting. Unlike AdaRound, POS achieve optimal rounding by setting the upper and lower bounds of w_f:

$$\begin{aligned} w_f^* \in [&s_w(\text{clip}(\text{floor}\left(\frac{w_f}{s_w}\right) + z, q_{min}, q_{max}) - z), \\ &s_w(\text{clip}(\text{floor}\left(\frac{w_f}{s_w}\right) + 1 + z, q_{min}, q_{max}) - z)]. \end{aligned} \tag{10}$$

If the updated weight $w_f^* \geq s_w(\text{clip}\left(\text{floor}\left(\frac{w_f^1}{s_w}\right) + 0.5 + z, q_{min}, q_{max}\right) - z)$, rounding up is the best rounding. Conversely, rounding down is preferable. This scheme does not introduce additional optimization tensors to the weight quantizer and does not require additional regularization terms. It focuses entirely on minimizing the reconstruction error. Due to the stepwise nature of rounding operations, minor weight updates may not alter the computed results of reconstruction loss, thereby increasing the difficulty of rounding optimization. To address this, in the early stages of rounding optimization, we retain the truncation operation of the weight quantizer but eliminate the rounding operation. This approach ensures that weight updates of any magnitude are promptly reflected in the computation results. It also avoids the gradient mismatch issue caused by rounding operations, thus accelerating the convergence of weights in the rounding direction.

In conclusion, POS can be divided into three stages:

1. Quantize the activation while keeping the weight unquantized. Optimize w_f to absorb weight perturbations caused by activation quantization and then set the upper and lower bounds of w_f according to Equation (10).
2. Quantize the activation and maintain truncation calculation of the weight quantizer but disable the rounding calculation.
3. Quantize both the activation and the weight.

In the first phase, the weights are not quantized, which means that the optimization space for Δw_f is unconstrained, allowing for w_f to be efficiently optimized to absorb perturbations caused by activation quantization. In the second phase, since the upper and lower bounds of w_f are set and the weight rounding operation is canceled, the value space for each element in w_q and Δw_f is restricted to a continuous interval. In the third phase, with the restoration of the rounding operation, the value space for each element in Δw_f is limited to a binary set. Compared to adaptive rounding or AdaQuant, POS progressively increases the constraints on weight optimization, considering both the effective reduction in reconstruction error and the potential overfitting issues caused by excessive optimization space. Furthermore, POS directly adjusts the weights, eliminating the need to balance regularization terms with reconstruction loss through parameter annealing, thus offering higher optimization efficiency.

4.2. Block-Wise Reconstruction: Randomly Weighted Quantized Activation

Considering the limited number of samples available for post-training quantization, Qdrop introduces a random dropout quantized activation (RDQA) scheme to alleviate overfitting in block-wise reconstruction:

$$q_y \leftarrow y\left(1 + u(y)\frac{q_y - y}{y}\right). \tag{11}$$

Here, $u(y)$ is a binary tensor randomly sampled from the Bernoulli distribution $B(1, 0.5)$ with the same dimension as y. Similar to the Dropout mechanism [37], RDQA is only employed during the quantization phase and is not applied during the inference stage, which can also be regarded as a data augmentation scheme.

Inspired by Mixup [38], we propose a randomly weighted quantized activation (RWQA) scheme in AE-Qdrop:

$$q_y \leftarrow t(y)q_y + (1 - t(y))y = y(1 + t(y)\frac{q_y - y}{y}). \tag{12}$$

Here, $t(y)$ is a FP tensor sampled from a uniform distribution $U(0, 1)$ with the same dimensions as y. In comparison to RDQA, RWAQ offers a more diverse set of feature inputs for block reconstruction, further enhancing the generalization capability of the quantization model.

4.3. Global Fine-Tuning

Block-wise reconstruction produces a pre-quantized model. The analysis presented in Section 3.4 reveals that each quantized network layer still has the potential for further optimization. As a result, AE-QDrop introduces global fine-tuning. We further analyze the optimization in the lth network block in the pre-quantized model:

$$
\begin{aligned}
&\mathbf{L}\left(f_n^q(f_{n-1}^q \ldots f_l(x_q^l, w_q^l))\right) - \mathbf{L}\left(f_n(f_{n-1} \ldots f_l(x_f^l, w_f^l))\right) \\
=&\mathbf{L}\left(f_n^q(f_{n-1}^q \ldots f_l(x_q^l, w_q^l))\right) - \mathbf{L}\left(f_n^q(f_{n-1}^q \ldots f_l(x_f^l, w_f^l))\right) \\
+&\mathbf{L}\left(f_n^q(f_{n-1}^q \ldots f_l(x_f^l, w_f^l))\right) - \mathbf{L}\left(f_n(f_{n-1} \ldots f_l(x_f^l, w_f^l))\right) \\
\approx&\Delta x_l^T \mathbf{g}_x^{(q,l)}\left(x_f^l, w_f^l\right) + \Delta w_l^T \mathbf{g}_w^{(q,l)}\left(x_f^l, w_f^l\right) + 2 \cdot \Delta y_l^T \mathbf{I} \Delta y_l \\
+&\mathbf{L}\left(f_n^q(f_{n-1}^q \ldots f_{l+1}(x_q^{l+1}, w_q^{l+1}))\right) - \mathbf{L}\left(f_n(f_{n-1} \ldots f_{l+1}(x_f^{l+1}, w_f^{l+1}))\right) \\
\approx&\sum_{i=l}^{n} \Delta x_i^T \mathbf{g}_x^{(q,i)}\left(x_f^i, w_f^i\right) + \sum_{i=l}^{n} \Delta w_i^T \mathbf{g}_w^{(q,i)}\left(x_f^i, w_f^i\right) + 2 \sum_{i=l}^{n} \Delta y_i^T \mathbf{I} \Delta y_i.
\end{aligned} \tag{13}
$$

Therefore, global fine-tuning aims to minimize $\sum_{i=1}^{n} \Delta y_i^T \mathbf{I} \Delta y_i$, $\mathbf{g}_x^{(q,i)}$, and $\mathbf{g}_w^{(q,i)}$. $\sum_{i=1}^{n} \Delta y_i^T \mathbf{I} \Delta y_i$ can be calculated directly, indicating that the output of each quantized network block should simultaneously match the output of the FP network block. We refer to this as feature matching. However, $\mathbf{g}_x^{(q,i)}$ and $\mathbf{g}_w^{(q,i)}$ cannot be calculated directly because the label of the sample is unknown.

If the quantized network converges just like the FP network, $\mathbf{g}_x^{(q,i)}$ and $\mathbf{g}_w^{(q,i)}$ are approximately equal to zero. Qualitatively speaking, for the same input sample, if the output of the quantized network aligns with that of the FP network, it is posited that the quantized network is also in a state of convergence. Therefore, we optimize $\sum_{i=l}^{n} \Delta x_i^T \mathbf{g}_x^{(q,i)}\left(x_f^i, w_f^i\right)$ and $\sum_{i=l}^{n} \Delta w_i^T \mathbf{g}_w^{(q,i)}\left(x_f^i, w_f^i\right)$ based on the perspective of knowledge distillation. The FP model is regarded as the teacher, while the quantized model is perceived as the student. The KL distance between the output $\mathbf{z}$ of the FP network and the output $\mathbf{z}_q$ of the quantized network is minimized, which is called logit matching. In order to avoid overfitting, we only correct the rounding direction of the weight. To sum up, the loss function $\mathbf{L}_{gf}$ of global fine-tuning is defined as

$$
\begin{aligned}
\mathbf{L}_{fm}^i =& \Delta y_i^T \mathbf{I} \Delta y_i = \|y_q^i - y^i\|_2^2, \\
\mathbf{L}_{lm} =& \sum_{i=1}^{m} p_i(z; \mathcal{T}) \log\left(\frac{p_i(z; \mathcal{T})}{p_i(z^q; \mathcal{T})}\right), \mathbf{p}_i(z; \mathcal{T}) = \frac{e^{z_i/\mathcal{T}}}{\sum_j^m e^{z_j/\mathcal{T}}}, \\
\mathbf{L}_{gf} =& L_{lm} + \frac{\theta}{n} \sum_{i=1}^{n} \mathbf{L}_{fm}^i,
\end{aligned} \tag{14}
$$

where $\mathcal{T}$ and θ represent the distillation temperature and the hyperparameter, respectively.

5. Experimental Result

5.1. Experimental Setup and Implementation Details

Experiments are organized in both image recognition and object detection tasks to verify the performance of AE-Qdrop. The ImageNet2012 dataset and the PASCAL VOC2007 dataset (http://host.robots.ox.ac.uk/pascal/VOC, accessed on 31 January 2024) are used for image recognition and object detection tasks, respectively. The ImageNet2012 dataset includes 1.2 million training images and 50,000 test images, with the top-1 recognition accuracy as the evaluation metric. The VOC2007 dataset consists of 2501 training images and 4951 test images, with MAP0.5 as the evaluation metric.

For the image recognition task, the quantized networks include ResNet18 (Res18), ResNet50 (Res50), MobileNetV2 (MV2), RegNet-600MF (Reg600M), RegNet-3.2GF (Reg3.2G), and MnasNetx2 (MNx2). For the object detection task, the quantized networks are MobileNetV1-SSD and MobileNetV2-SSD. All experiments are based on the hardware platform of GeForce RTX 3090 Ti GPU and Intel(R) Core(TM) i7-7700K CPU. The software environment mainly includes Python 3.8 and Pytorch 2.1. The pre-trained models of the image recognition networks are sourced from Pytorchcv (https://github.com/donnyyou/

PyTorchCV, accessed on 31 January 2024), and the SSD pre-trained models are obtained from the open-source project pytorch-ssd (https://github.com/qfgaohao/pytorch-ssd, accessed on 31 January 2024).

The code of AE-Qdrop is based on the open-source implementation of Qdrop and follows its related settings (https://github.com/wimh966/QDrop, accessed on 31 January 2024). For example, the BN layers in the network are merged with the convolutional layers; the first and last layers of the quantized network are kept at 8 bit (we note that only the Backbone part is quantized for SSD networks); the activation quantization scale factors are also optimized simultaneously, among others. A total of 1024 random samples from the training set are selected as the calibration dataset for the image recognition task (the calibration dataset for the object detection task includes 256 random training samples).

During the block-wise reconstruction phase, each network block undergoes 2000 optimization iterations. The three phases of POS consist of 800, 400, and 800 iterations, respectively. The Adam optimizer is used with an initial learning rate of 4×10^{-5} (the initial learning rate for 2-bit quantization is 4×10^{-4}), and it varies based on a cosine decay strategy. The global fine-tuning iteration number is set to 2000, with a distillation temperature of $\mathcal{T} = 20$ and a hyperparameter of $\theta = 0.1$. The SGD optimizer is utilized with an initial learning rate of 1×10^{-7} (the initial learning rate for 2-bit quantization is 1×10^{-6}), also varying based on a cosine decay strategy. For the object detection task, the logit matching loss is adjusted to $\|z_q - z\|_2^2$.

5.2. Comprehensive Comparison

Table 2 presents the quantization results of various post-training quantization techniques in image recognition networks. Under W4A4, LAPQ only optimizes the quantization scale factor without adjusting weights, resulting in significantly lower quantization accuracy compared to other schemes that involve weight adjustment. For MV2, Reg600M, and Reg3.2G, its average quantization accuracy loss exceeds 20%. AdaRound, BrecQ, and Qdrop all employ adaptive rounding technique. Due to the neglect of inter-layer dependencies, the quantization accuracy of AdaRound, which is based on layer-wise reconstruction, is significantly lower than that of BrecQ, which is based on block-wise reconstruction. However, BrecQ does not quantize activations during the adaptive rounding process, resulting in its inability to perceive the weight perturbation caused by activation quantization noise, which leads to a significant loss in 4-bit quantization accuracy. Particularly for MobileNetV2, the accuracy loss in BrecQ reaches 10.54%. The average accuracy loss for Qdrop is only 2.76%, achieving acceptable quantization accuracy. Under W4A4, the weight perturbation caused by activation quantization is relatively small, and the impact of subsequent quantized network blocks on the current block is also lower, thus the disadvantages of adaptive rounding and block-wise reconstruction are less evident. Consequently, the quantization accuracy of Qdrop and AE-Qdrop is comparable, but AE-Qdrop offers higher quantization efficiency (as detailed in Table 3).

As bit-width decreases, AE-Qdrop demonstrates a significant accuracy advantage on lightweight networks such as MV2 and MNx2. For example, under W2A2, the quantization accuracy of AE-Qdrop exceeds that of Qdrop by 6.49% (MV2) and 3.22% (MNx2). Typically, lightweight networks feature broader numerical distribution ranges and fewer weights. The former leads to larger parameter quantization errors, exacerbating the discrepancy between the optimal solutions of Equations (7) and (8). The latter diminishes the efficacy of adaptive rounding, analogous to a consensus that fewer neural network parameters result in weaker fitting optimization capability. Notably, under W4A2, the performance advantage of AE-Qdrop is most pronounced, exceeding Qdrop by 12.07% (MV2) and 8.36% (MNx2). Compared to 2-bit weights, 4-bit weights can encapsulate more information. AE-Qdrop's progressive optimization strategy relaxes constraints on weights, enabling 4-bit weights to fully exploit their representational capacity. This allows for them to absorb the perturbations caused by activation quantization and minimize reconstruction loss. Consequently, under W4A2, AE-Qdrop achieves its greatest performance advantage.

Table 2. Quantization results of various post-training quantization techniques for image recognition networks.

Method	Bits(W/A)	Res18	Res50	MV2	Reg600M	Reg3.2G	MNx2
FP32	32/32	71.01	76.63	72.62	73.52	78.46	76.52
LAPQ	4/4	60.30	70.00	49.70	57.71	55.89	65.32
AdaRound		67.96	73.88	61.52	68.20	73.85	68.86
BrecQ		68.16	72.95	62.08	68.94	73.94	71.01
Qdrop-4k		69.05	74.79	67.72	70.60	76.21	72.57
Qdrop		69.16	74.91	67.86	70.95	76.45	72.81
AE-Qdrop		69.24	74.98	67.93	70.83	76.54	72.68
AdaRound	4/2	0.44	0.17	0.29	2.14	0.10	0.93
BrecQ		31.19	16.95	0.28	4.22	3.47	6.34
Qdrop-4k		56.46	61.87	10.26	46.68	59.58	16.71
Qdrop		58.10	63.26	17.03	49.78	61.87	33.96
AE-Qdrop		58.48	64.53	29.10	52.71	64.29	42.32
AdaRound	2/2	0.39	0.13	0.12	0.79	0.11	0.40
BrecQ		25.91	8.26	0.19	2.49	1.72	0.38
Qdrop-4k		46.12	48.81	6.18	31.30	48.38	16.37
Qdrop		51.55	55.21	9.97	39.31	53.88	24.21
AE-Qdrop		52.24	55.55	16.46	40.58	54.56	27.43

Table 3. Comparison of quantization time (minute).

	Res18	Res50	MV2	Reg600M	Reg3.2G	MNx2
AdaRound	19.4	65.1	42.8	38.6	75.6	58.9
BrecQ	17.3	51.2	28.7	28.9	60.4	44.7
Qdrop	19.1	64.4	37.7	32.8	74.8	58.4
Qdrop-4k	4.6	15.1	8.6	7.4	16.8	13.8
BR	2.4	8.7	4.7	3.8	9.7	7.8
GF	1.9	5.8	2.9	2.8	7.1	5.1
AE-Qdrop	4.3	13.5	7.6	6.6	16.8	12.9

Table 3 displays the quantization times for AdaRound, BrecQ, Qdrop, and AE-Qdrop. Among these, AdaRound exhibits the lowest quantization efficiency. While BrecQ demonstrates slightly greater efficiency in quantization time compared to Qdrop, it fails to offset its significant disadvantage in quantization accuracy. For MV2, commonly used in mobile deployment, AE-Qdrop requires only 7.6 min to complete quantization, with block-wise reconstruction (BR) and global fine-tuning (GF) taking 4.7 min and 2.9 min, respectively. The efficiency of AE-Qdrop is fivefold that of Qdrop, and it achieves higher quantization accuracy as shown in Table 2, thereby confirming that it is a high-accuracy and efficient quantization scheme. Qdrop-4k represents the quantization results with the number of adaptive rounding iterations set at 4000. Its efficiency is comparable to that of AE-Qdrop, but the reduced iteration count leads to a noticeable decrease in accuracy. As indicated in Table 2, under W2A2, the accuracy of Qdrop-4k falls by 3.79%∼8.01% compared to Qdrop, and by 6.06%∼11.06% compared to AE-Qdrop. These results highlight the dependence of adaptive rounding technology on sufficient iteration cycles.

The quantization results of the object detection network are presented in Table 4. Under W4A4, the quantization accuracy loss of AE-Qdrop is only 3.39% and 3.6%, respectively, surpassing Qdrop by 0.73% and 0.46%. The reduction in quantization bit-width further highlights the accuracy advantage of AE-Qdrop. Under W2A2, AE-Qdrop's quantization accuracy surpasses Qdrop by 1.49% and 1.59%, respectively. Notably, although Qdrop-4k shows a small difference from Qdrop under 4w4a, its accuracy loss is quite significant under W4A2 or W2A2. In particular, for the W2A2 MobileNetV1-SSD, its quantization accuracy drops to only 21.98%, a decline of 9.71% compared to Qdrop. Therefore, simply reducing the number of iterations for adaptive rounding can directly improve quantization

efficiency, but it brings catastrophic consequences to low-bit quantization accuracy. Similar to the results of quantizing image recognition networks, AE-Qdrop shows the most significant accuracy advantage under 4w2a. For MobileNetV1-SSD and MobileNetV2-SSD, AE-Qdrop's quantization accuracy improves by 6.06% and 9.79%, respectively, compared to Qdrop.

Table 4. The low-bit quantization accuracy (MAP0.5) of SSD.

Method	Bits (W/A)	MobileNetV1-SSD	MobileNetV2-SSD
FP32	32/32	67.60	68.70
Qdrop-4k	4/4	63.46	63.91
Qdrop		63.48	64.09
AE-Qdrop		64.21	65.10
Qdrop-4k	4/2	36.77	24.81
Qdrop		38.61	28.10
AE-Qdrop		44.67	37.89
Qdrop-4k	2/2	21.98	19.58
Qdrop		30.18	26.45
AE-Qdrop		31.69	28.04

Figures 4 and 5 visualize the detection results of quantized SSD networks. Clearly, the detection results of AE-Qdrop are closer to those of the FP networks in terms of category confidence and the positioning of object bounding boxes. Taking Figure 4 as an example, under W4A4, AE-Qdrop shows a confidence of 82% for the train, while Qdrop and Qdrop-4k show confidences of 70% and 57%, respectively, for the train. Under W4A2, compared to the detection results of the FP network, AE-Qdrop only leads to a decrease in object confidence but does not cause any missed or false detections. However, Qdrop fails to detect the dining table and incorrectly detects a human target. Under W2A2, both Qdrop and Qdrop-4k result in greater decreases in confidence.

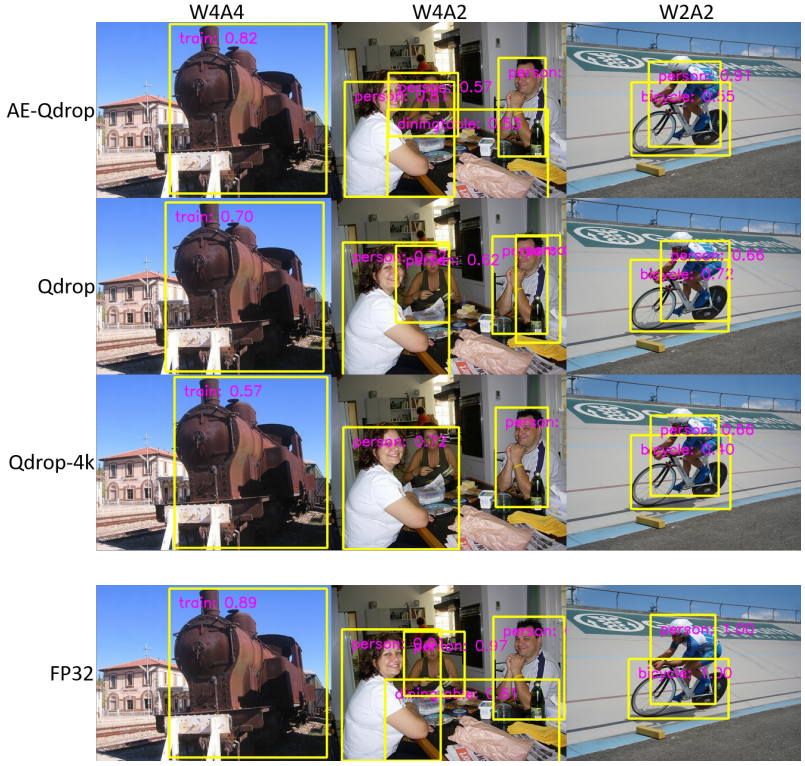

Figure 4. Detection results of MobileNetV1-SSD under different bit-widths.

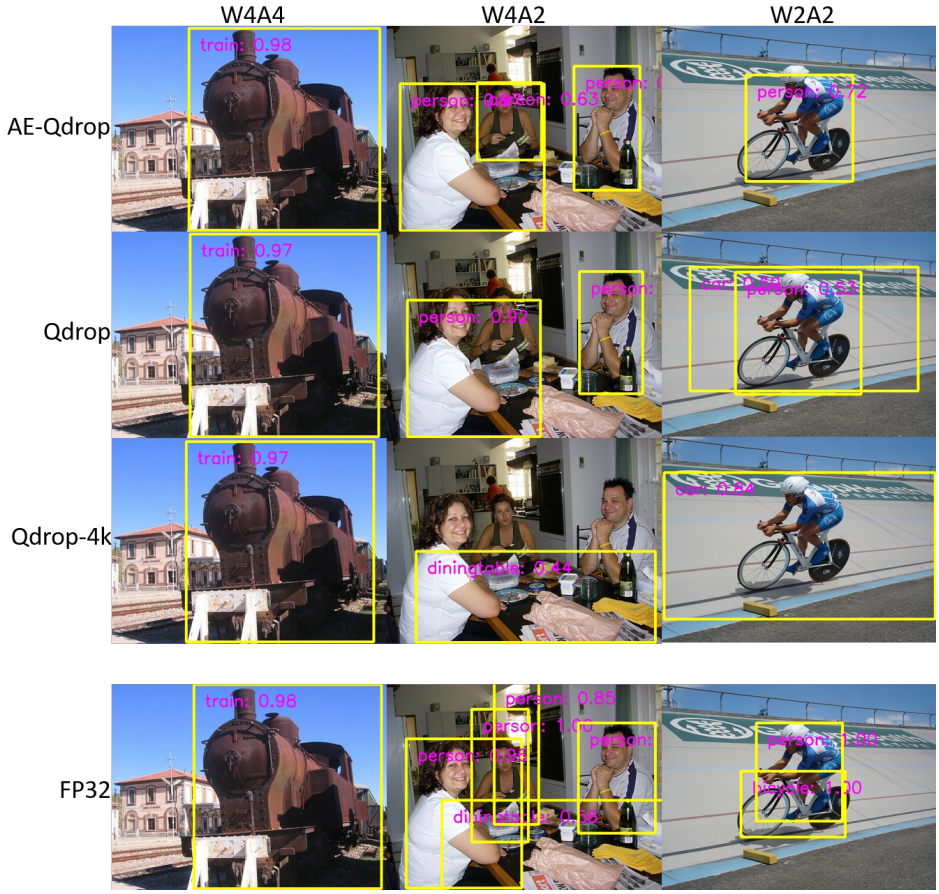

Figure 5. Detection results of MobileNetV2-SSD under different bit widths.

5.3. Ablation Study

In Table 5, we explore the impact of various design components on quantization performance under W2A2. The baseline represents the result of minimizing the block reconstruction error by directly optimizing the weights using STE (consistent with AdaRound), without employing progressive optimization strategies and data augmentation. The limited number of calibration samples tends to make block reconstruction susceptible to overfitting; therefore, the benefits of data augmentation become quite substantial, particularly for larger networks such as ResNet and RegNet, which see an accuracy improvement of over 3.6%. Compared to RDQA proposed in Qdrop, RWQA offers a performance gain of 0.6% to 3.7%, which means that RWQA can better reduce the risk of overfitting. In contrast to the direct optimization of weights using the STE, the implementation of POS effectively mitigates challenges like gradient mismatch and overfitting by gradually shrinking the optimization space. This approach results in a widespread enhancement of quantization precision. Notably, in the cases of MV2 and MNx2, there are marked accuracy increments of 4.66% and 4.77% respectively. The Baseline+RWQA+POS configuration epitomizes the quantization accuracy achieved in the first phase (block-wise reconstruction) of AE-Qdrop. The combination of RWQA and POS results in an accuracy enhancement ranging from 5.3% to 13.15%, which further corroborates the efficacy of RWQA and POS. The quantization results of AE-Qdrop are obtained by conducting global fine-tuning subsequent to block-wise reconstruction. As observed, GF significantly enhances the quantization precision for MV2, MNx2, and Reg600M. We hypothesize that the larger parameter volume of ResNet and Reg3.2G imparts robustness against quantization noise, diminishing the disparity between $f^q n(f^q n-1 \ldots f_l(\cdot))$ and $f_n(f_{n-1} \ldots f_l(\cdot))$, thereby resulting in a comparatively lower gain from global fine-tuning.

In Table 6, we explore the quantization results achieved solely through single-stage global fine-tuning. MSE [21] represents an early approach to post-training quantization. It determines quantization scale factors by minimizing the L2 norm of parameter quantization errors. In recent works [25,26] and in AE-Qdrop, MSE is utilized to provide a pre-quantized network for block-wise reconstruction. Under W4A4, although the quantization results of MSE lead to significant accuracy loss, the quantized models retain some image recognition capability. Global fine-tuning significantly enhances performance, but there is a notable disparity compared to the results of Brecq and Qdrop. As the quantization bit-width decreases, especially under W2A2, the MSE-derived quantized model completely fails, and global fine-tuning offers no benefits, indicating that global fine-tuning alone cannot retrain a quantized network with just 1024 calibration samples. Therefore, for optimal quantization performance, the block-wise reconstruction phase in AE-Qdrop is indispensable, providing a favorable initial state for global fine-tuning.

Table 5. Under W2A2, the impact of various design components on quantization performance.

Method	Res18	Res50	MV2	Reg600M	Reg3.2G	MNx2
Baseline	46.40	47.90	6.44	27.73	41.17	15.72
Baseline+RDQA	50.00	52.29	7.52	36.29	52.89	16.64
Baseline+RWQA	51.05	52.89	8.78	36.92	53.51	20.35
Baseline+POS	47.12	49.55	11.10	28.75	41.74	20.49
Baseline+RWQA+POS	51.73	55.36	13.33	39.06	54.32	24.61
Baseline+RWQA+POS+GF	52.24	55.55	16.46	40.58	54.56	27.43

Table 6. The quantization results achieved solely through single-stage global fine-tuning.

	Method	Res18	Res50	MV2	Reg600M	Reg3.2G	MNx2
W4A4	MSE	49.75	65.54	22.40	51.70	66.75	49.71
	MSE+GF	65.05	69.10	36.69	60.05	70.24	56.68
W4A2	MSE	9.33	4.35	0.11	1.9	2.01	0.27
	MSE+GF	25.18	6.98	0.18	3.3	4.43	0.28
W2A2	MSE	0.08	0.16	0.11	0.15	0.11	0.10
	MSE+GF	0.08	0.10	0.09	0.16	0.17	0.10

6. Conclusions

This paper theoretically analyzes the deficiencies of adaptive rounding and block-wise reconstruction and proposes a highly precise and efficient quantization scheme-AE-Qdrop. Addressing the constraints in the weight optimization space imposed by adaptive rounding and its time-consuming nature, AE-Qdrop introduces a progressive optimization strategy to enhance the optimization space and efficiency. Moreover, the randomly weighted quantized activation diversifies the activation inputs for block-wise reconstruction, further mitigating the overfitting issue caused by insufficient samples. To counter the shortcomings of block-wise reconstruction, the proposed global fine-tuning considers the dependencies between network blocks and enhances quantization accuracy through feature matching and logit matching. The precision advantages of AE-Qdrop are effectively evaluated in image recognition and object detection tasks, and its quantization efficiency is five times that of Qdrop. Ablation experiments further assess the performance gains of the progressive optimization strategy, random weighted quantized activation, and block-wise reconstruction.

However, there remains a significant gap in accuracy between AE-Qdrop and QAT. Observing the significant improvements in quantization performance due to data augmentation, we plan to delve deeper into the integration of various data augmentation techniques with block-wise reconstruction and global fine-tuning in our future work, aiming to further enhance quantization accuracy.

Author Contributions: Conceptualization, Methodology, Software, Writing—original draft preparation, J.L.; Conceptualization, Methodology, G.C.; Resources, Supervision, Funding acquisition, M.J. Validation, Investigation, W.M.; Supervision, Project administration, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China 92364202 and in part by the CAS Strategic Leading Science and Technology Project XDA18040400, XDB44000000.

Data Availability Statement: Imagenet Dataset: https://image-net.org/, accessed on 31 January 2024. Other data will be made available on request.

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

References

1. Zhu, A.; Wang, B.; Xie, J.; Ma, C. Lightweight Tunnel Defect Detection Algorithm Based on Knowledge Distillation. *Electronics* **2023**, *12*, 3222. [CrossRef]
2. Wu, P.; Wang, Z.; Li, H.; Zeng, N. KD-PAR: A knowledge distillation-based pedestrian attribute recognition model with multi-label mixed feature learning network. *Expert Syst. Appl.* **2024**, *237*, 121305. [CrossRef]
3. Lopes, V.; Carlucci, F.M.; Esperança, P.M.; Singh, M.; Yang, A.; Gabillon, V.; Xu, H.; Chen, Z.; Wang, J. Manas: Multi-agent neural architecture search. *Mach. Learn.* **2024**, *113*, 73–96. [CrossRef]
4. Song, Y.; Wang, A.; Zhao, Y.; Wu, H.; Iwahori, Y. Multi-Scale Spatial–Spectral Attention-Based Neural Architecture Search for Hyperspectral Image Classification. *Electronics* **2023**, *12*, 3641. [CrossRef]
5. Li, Y.; Adamczewski, K.; Li, W.; Gu, S.; Timofte, R.; Van Gool, L. Revisiting random channel pruning for neural network compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 191–201.
6. Shen, W.; Wang, W.; Zhu, J.; Zhou, H.; Wang, S. Pruning-and Quantization-Based Compression Algorithm for Number of Mixed Signals Identification Network. *Electronics* **2023**, *12*, 1694. [CrossRef]
7. Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M.W.; Keutzer, K. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2022; pp. 291–326.
8. Ahn, H.; Chen, T.; Alnaasan, N.; Shafi, A.; Abduljabbar, M.; Subramoni, H. Performance Characterization of using Quantization for DNN Inference on Edge Devices: Extended Version. *arXiv* **2023**, arXiv:2303.05016.
9. 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, New Orleans, LA, USA, 18–24 June 2022; pp. 4942–4952.
10. Kim, D.; Lee, J.; Ham, B. Distance-aware quantization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5271–5280.
11. Peng, H.; Wu, J.; Zhang, Z.; Chen, S.; Zhang, H.T. Deep network quantization via error compensation. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 4960–4970. [CrossRef] [PubMed]
12. Esser, S.K.; McKinstry, J.L.; Bablani, D.; Appuswamy, R.; Modha, D.S. Learned Step Size quantization. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
13. Bhalgat, Y.; Lee, J.; Nagel, M.; Blankevoort, T.; Kwak, N. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 696–697.
14. 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, Nashville, TN, USA, 20–25 June 2021; pp. 6448–6457.
15. Li, Z.; Ni, B.; Li, T.; Yang, X.; Zhang, W.; Gao, W. Residual quantization for low bit-width neural networks. *IEEE Trans. Multimed.* **2021**, *25*, 214–227. [CrossRef]
16. Xu, W.; Li, F.; Jiang, Y.; Yong, A.; He, X.; Wang, P.; Cheng, J. Improving extreme low-bit quantization with soft threshold. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 1549–1563. [CrossRef]
17. Guo, N.; Bethge, J.; Meinel, C.; Yang, H. Join the high accuracy club on ImageNet with a binary neural network ticket. *arXiv* **2022**, arXiv:2211.12933.
18. 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.
19. Pei, Z.; Yao, X.; Zhao, W.; Yu, B. Quantization via distillation and contrastive learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–13. [CrossRef] [PubMed]
20. Li, Z.; Yang, B.; Yin, P.; Qi, Y.; Xin, J. Feature Affinity Assisted Knowledge Distillation and Quantization of Deep Neural Networks on Label-Free Data. *arXiv* **2023**, arXiv:2302.10899.

21. Choukroun, Y.; Kravchik, E.; Yang, F.; Kisilev, P. Low-bit quantization of neural networks for efficient inference. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 3009–3018.
22. Jeong, E.; Kim, J.; Tan, S.; Lee, J.; Ha, S. Deep learning inference parallelization on heterogeneous processors with tensorrt. *IEEE Embed. Syst. Lett.* **2021**, *14*, 15–18. [CrossRef]
23. Nahshan, Y.; Chmiel, B.; Baskin, C.; Zheltonozhskii, E.; Banner, R.; Bronstein, A.M.; Mendelson, A. Loss aware post-training quantization. *Mach. Learn.* **2021**, *110*, 3245–3262. [CrossRef]
24. Nagel, M.; Amjad, R.A.; Van Baalen, M.; Louizos, C.; Blankevoort, T. Up or down? adaptive rounding for post-training quantization. In Proceedings of the International Conference on Machine Learning. PMLR, Virtual, 13–18 July 2020; pp. 7197–7206.
25. Li, Y.; Gong, R.; Tan, X.; Yang, Y.; Hu, P.; Zhang, Q.; Yu, F.; Wang, W.; Gu, S. BRECQ: Pushing the Limit of Post-Training Quantization by Block Reconstruction. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual, 3–7 May 2021.
26. Wei, X.; Gong, R.; Li, Y.; Liu, X.; Yu, F. QDrop: Randomly Dropping Quantization for Extremely Low-bit Post-Training Quantization. In Proceedings of the Tenth International Conference on Learning Representations, ICLR 2022, Virtual, 25–29 April 2022.
27. Nagel, M.; Baalen, M.v.; Blankevoort, T.; Welling, M. Data-free quantization through weight equalization and bias correction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1325–1334.
28. Banner, R.; Nahshan, Y.; Soudry, D. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Adv. Neural Inf. Process. Syst.* **2019**, 7948–7956.
29. Fang, J.; Shafiee, A.; Abdel-Aziz, H.; Thorsley, D.; Georgiadis, G.; Hassoun, J.H. Post-training piecewise linear quantization for deep neural networks. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 69–86.
30. Park, D.; Lim, S.G.; Oh, K.J.; Lee, G.; Kim, J.G. Nonlinear depth quantization using piecewise linear scaling for immersive video coding. *IEEE Access* **2022**, *10*, 4483–4494. [CrossRef]
31. Zhao, M.; Ning, K.; Yu, S.; Liu, L.; Wu, N. Quantizing Oriented Object Detection Network via Outlier-Aware Quantization and IoU Approximation. *IEEE Signal Process. Lett.* **2020**, *27*, 1914–1918. [CrossRef]
32. Zhao, R.; Hu, Y.; Dotzel, J.; De Sa, C.; Zhang, Z. Improving neural network quantization without retraining using outlier channel splitting. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 7543–7552.
33. Yao, H.; Li, P.; Cao, J.; Liu, X.; Xie, C.; Wang, B. RAPQ: Rescuing Accuracy for Power-of-Two Low-bit Post-training Quantization. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23–29 July 2022; pp. 1573–1579. [CrossRef]
34. Jeon, Y.; Lee, C.; Cho, E.; Ro, Y. Mr.BiQ: Post-Training Non-Uniform Quantization based on Minimizing the Reconstruction Error. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 12319–12328. [CrossRef]
35. Hubara, I.; Nahshan, Y.; Hanani, Y.; Banner, R.; Soudry, D. Accurate post training quantization with small calibration sets. In Proceedings of the International Conference on Machine Learning, PMLR, Online, 18–24 July 2021; pp. 4466–4475.
36. Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv* **2018**, arXiv:1806.08342.
37. Baldi, P.; Sadowski, P.J. Understanding dropout. *Adv. Neural Inf. Process. Syst.* **2013**, 2814–2822.
38. Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6438–6447.