Reg-PTQ: Regression-specialized Post-training Quantization for Fully Quantized Object Detector

Yifu Ding[1, 2], Weilun Feng[3], Chuyan Chen[3], Jinyang Guo[1, 4], Xianglong Liu[1, 5, 6†]
[1] State Key Laboratory of Complex & Critical Software Environment, Beihang University
[2] Shen Yuan Honors College, Beihang University [3] Beihang University
[4] Institute of Artificial Intelligence, Beihang University [5] Zhongguancun Laboratory
[6] Institute of data space, Hefei Comprehensive National Science Center
{yifuding, fwl2023, ccy2006, jinyangguo, xlliu}@buaa.edu.cn

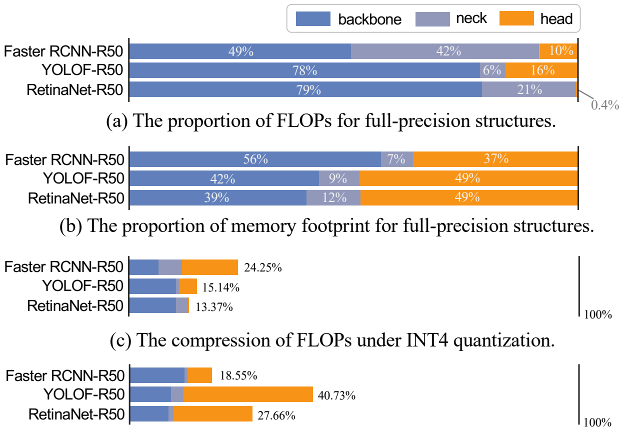
Abstract

Although deep learning based object detection is of great significance for various applications, it faces challenges when deployed on edge devices due to the computation and energy limitations. Post-training quantization (PTQ) can improve inference efficiency through integer computing. However, they suffer from severe performance degradation when performing full quantization due to overlooking the unique characteristics of regression tasks in object detection. In this paper, we are the first to explore regression-friendly quantization and conduct full quantization on various detectors. We reveal the intrinsic reason behind the difficulty of quantizing regressors with empirical and theoretical justifications, and introduce a novel ***Reg****ression-specialized* ***P****ost-****T****raining* ***Q****uantization (****Reg-PTQ****) scheme. It includes Filtered Global Loss Integration Calibration to combine the global loss with a two-step filtering mechanism, mitigating the adverse impact of false positive bounding boxes, and Learnable Logarithmic-Affine Quantizer tailored for the non-uniform distributed parameters in regression structures. Extensive experiments on prevalent detectors showcase the effectiveness of the well-designed Reg-PTQ. Notably, our Reg-PTQ achieves* 7.6× *and* 5.4× *reduction in computation and storage consumption under INT4 with little performance degradation, which indicates the immense potential of fully quantized detectors in real-world object detection applications.*

1. Introduction

Object detection [10, 11, 23, 42, 45, 57] is one of the most fundamental and challenging problems in computer vision. The current popular architectures, including convolution neural networks (CNNs) based [22, 46, 47, 50, 52, 53] and transformer-based [7, 14, 30, 33–35, 60] detection models, which are designed as powerful yet complex structures to deal with the detection of visual objects [61]. However, the existing detection models suffer from extremely high computational costs, making them infeasible to deploy on edge devices. This limits the broader application in practical scenarios. To mitigate this gap, several compression techniques [1, 15–17, 58] have been proposed to improve the efficiency of networks, among which quantization reduces the computational complexity and memory footprint by using lower bit-widths to represent network parameters. Post-training quantization (PTQ) is a widely used approach because of its wide versatility and low production cost, which directly applies quantization to a well-trained floating-point model without time-consuming retraining.

However, although quantization techniques are proven effective in classification, they still lack sufficient investi-



(a) The proportion of FLOPs for full-precision structures.

(b) The proportion of memory footprint for full-precision structures.

(c) The compression of FLOPs under INT4 quantization.

(d) The compression of memory footprint under INT4 quantization.

Figure 1. Comparison of FLOPs and parameters in full-precision and W4A4 quantized detection models. Head structures take non-neglectable percentage of computation and memory. And quantization significantly reduces the overall FLOPs and memory storage.

* Corresponding author: xlliu@buaa.edu.cn

gation for the object detection task. Most current PTQ approaches [12, 13, 26, 55] only evaluate the backbone and neck of the detector while avoiding quantizing the detection head. However, it is meaningful to quantize the head as well. Figure 1 (a) and (b) shows the FLOPs and memory footprints of different structures in typical detection models, including RetinaNet [29], YOLOF [5] and Faster RCNN [44]. We notice that detection heads take a considerable proportion of computation, and take almost half of the memory footprint in various detectors. After full quantization (Figure 1 (c) and (d)), we observe that both computation and memory footprint are significantly reduced. Compared to solely quantizing the backbone and neck, quantizing the head further achieves impressive compression and acceleration ratios with little performance degradation. However, directly applying classical PTQ algorithms on detection heads leads to significant performance drops, especially at extremely low bit-width.

To this end, we aim to explore how to generate fully quantized detectors with satisfactory performance, which is, to the best of our knowledge, the first universal PTQ framework to fully quantize various detection architectures. We first analyze and reveal the challenges in quantizing regression models with theoretical and experimental justifications. We observe that (1) compared with classification, regression is more sensitive to quantization noise, (2) minimizing local quantization error fails to select the optimal scaling factors for regressors. Hence, existing calibration metrics designed for classification lead to sub-optimal scaling factors, (3) regressors have non-uniform weight distribution, so uniform quantizers will cause coarse quantized representation and lose much information after quantization. And we theoretically prove that the sensitivity of the detection model with respect to quantization noise and the weight distribution is caused by the distance-based objective function (*i.e.*, L1/L2-norm) in the regressor.

Based on our observations and findings, we propose a novel PTQ scheme named **Reg**ression-specialized **P**ost-**T**raining **Q**uantization (**Reg-PTQ**), the first universal PTQ framework for accurate full-quantized detection models. Reg-PTQ contains two novel techniques: Filtered Global-Loss Integration Calibration (FGIC) strategy and Learnable Logarithmic Affine Quantizer (LLAQ). To improve the calibration, FGIC combines the local reconstruction loss and global prediction loss with a two-step filtering mechanism, removing redundant predicted boxes and providing precise gradients in fine-tuning the scaling factors. As for the non-uniform distributed parameters, LLAQ applies logarithmic-affine transformation, which better learns the quantization factors for regression structures and preserves the characteristics of the original representation.

In summary, our contributions are as follows:

- We first give an experimental and theoretical analysis of quantizing regressors and discover the limitations of existing quantization methods for regression.
- We propose the first regression-specialized PTQ framework named Reg-PTQ, a universal PTQ scheme for full quantization on various detection architectures. Reg-PTQ presents a novel Filtered Global-Loss Integration Calibration strategy for efficient fine-tuning with filtered losses, and a Learnable Logarithmic Affine Quantizer for better representation of non-uniform distributed parameters.
- We conduct comprehensive experiments on various detection models. Efficiency comparisons show that the fully quantized detectors can achieve about $7.6\times$ and $5.4\times$ reduction in computation and storage without significant performance loss.

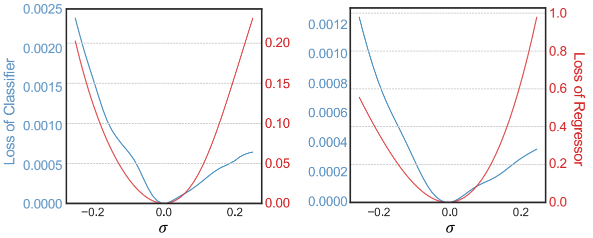
2. Related Works

2.1. Object Detection

In recent years, many frameworks have been proposed for object detection, which can be broadly classified into two categories: two-stage and one-stage detectors. Two-stage detectors propose candidate object regions first and refine the bounding boxes. For example, Faster R-CNN [44] introduces a region proposal network to remedy the cost and finetunes the bounding boxes in second stage. Mask R-CNN [18] adds a branch for predicting segmentation masks. For one-stage detectors, RetinaNet [28] utilizes the focal loss to address the class imbalance during training. YOLO [43] proposes to predict from full images in one evaluation, significantly speeding up the detection process. However, these classic approaches require significant computation, hindering the practical usage of in real-world implementation. Efficient detection models have emerged recently, such as EfficientDet [48], MobileDets [58], YOLOv7 [49]. But they use full-precision calculations, leaving room for further acceleration by quantization. Therefore, we propose a post-training quantization method to generate a fully quantized detector, which can significantly reduce both the computation and storage burden of these approaches.

2.2. Post-Training Quantization

The existing quantization methods can be classified into Quantization-Aware Training (QAT) [9, 25, 37, 54] and Post-Training Quantization (PTQ) [13, 19, 38]. QAT requires significant GPU resources for training or fine-tuning to achieve better accuracy performance. Instead, PTQ as a training-free method has gained widespread attention in real-world practice since it reduces production costs and time consumption. PTQ techniques tailored for CNNs [20, 21, 36, 40] and vision transformers [6, 33, 60] have made remarkable achievements in the field of classification. For example, Li *et al.* [24] proposed BRECQ to cal-



(a) Perturbation sampled from Gaussian distribution

(b) Perturbation sampled from uniform distribution

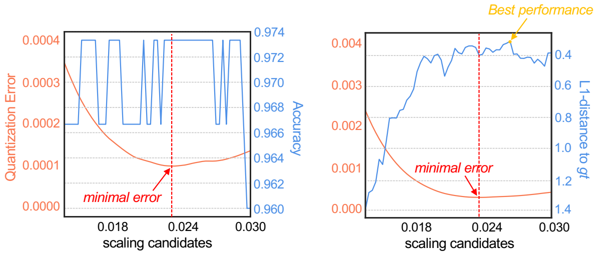
Figure 2. Loss landscapes of classifier (blue lines) and regressor (red lines) with (a) Gaussian distributed and (b) uniform distributed perturbations.

ibrate in the block-wise manner and reconstruct them sequentially. Wei *et al*. [56] studied the flatness of weight and proposed QDrop to consider activation information in quantizing weight. Cao *et al*. [3] introduced SSQL to pretrain quantization-friendly models in a self-supervised style. Li *et al*. [26] and Liu *et al*. [32] focused on the quantization of transformer-based architectures named RepQ-ViT and NoisyQuant. RepQ-ViT includes scale reparameterization for Layernorm and Softmax operations. And NoisyQuant analyzes the quantization noise theoretically, which provides novel insights in reducing the quantization error.

However, these works designed for the classification task have seldom been evaluated on detection models or only quantized the backbone and neck. Directly applying the methods to detectors leads to a significant performance drop since the architecture and objective of detection are different from that of classification. For the quantization of the detector, Q-DETR [59] proposed to quantize the DETR [4] by solving the bi-level optimization problem using rectified distillation. Q-YOLO [51] proposed a novel quantization approach to tackle the unilateral distributed activation for quantizing the YOLO [43] detector. DetPTQ [39] studied the influence of Lp metric and performed NMS before calculating the prediction loss. However, these works are dedicated to a specific detection architecture but not universal to general detection models, or only quantize the backbone and neck but leave the head structure unquantized. In contrast, our Reg-PTQ framework is the first universal PTQ framework for fully quantized detection models, which can be versatile to diverse detector architectures and bring significantly higher compression and acceleration ratios.

3. Motivation

In this section, we elaborate the motivation for studying the quantization of regression tasks. We demonstrate the current quantization principles designed for the classification task are ineffective for regression models.



(a) Quantization error and accuracy of quantized classifier

(b) Quantization error and L1-distance to *gt* (ground truth) of quantized regressor

Figure 3. Quantization error measured by MSE (red lines) and model performance (blue lines) with different scaling factors in quantized classifer and regressor.

3.1. Empirical Observation

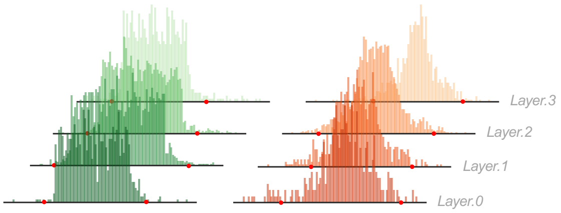
To find the difference in quantizing regressor and classifier, we first design a toy experiment which can reach the same conclusion with the real one, and eliminate irrelevant variables to focus on the fundamental differences. We provide the detailed advantage of toy models in Supplementary **??**. Specifically, we designed 4-layer networks for both regression and classification while controlling approximately the same amount of parameters. For regression, we generate training data from a quadratic equation. For classification, we use the Iris flower dataset [1] with three species, and each sample has four features. We train both models on corresponding datasets until convergence.

Observation 1: Regressor is more sensitive to perturbation than classifier. To evaluate the robustness of the regressor and classifier, we superpose perturbations on the well-trained models in two cases. In the vanilla hard quantization, the rounding operation can be considered as a random noise on the parameters, which is sampled from a uniform distribution on the interval $[-0.5, 0.5)$. On the other hand, soft quantization means training with differentiable functions before rounding and can be approximated as a noise approximated to the Gaussian distribution.

As shown in Figure 2, we add perturbation on the model parameters sampled from (a) Gaussian distribution $N(0, \sigma)$ and (b) uniform distribution on the interval $[-\sigma, \sigma)$. We use mean squared error (MSE) to measure the output error between the perturbed models and the original ones. The blue line represents the loss curve of the classifier, and the red line represents the regressor's. It can be seen from the figure that (1) both Gaussian and uniform noise have larger impacts on the regressor compared to the classifier, which means that the regressor is more sensitive to perturbations than the classifier, (2) the regressor is more sensitive to uniform noise than Gaussian noise, which means that the vanilla rounding operation may lead to severe quantization error in regressors.

Observation 2: Minimizing local quantization error

[1] https://www.kaggle.com/datasets/arshid/iris-flower-dataset



(a) Weight in classifier (b) Weight in regressor

Figure 4. Histogram of the original weight of (a) classifier and (b) regressor. The red points are truncated position for the minimal reconstruction error in INT4 quantization.

selects sub-optimal scaling factors for regressor. We apply quantization on classifier and regressor, calculate the local quantization error, and test the model performance for each scaling candidate in Figure 3. The red lines are the quantization error, and the blue lines are the final performance of the network. Figure 3 (a) shows the quantization error and accuracy of the classifier. We can see that the classifier is robust to quantization, and the scaling factor with minimal quantization error results in the best accuracy performance. However, for the regressor, the scaling factor with minimal error cannot guarantee the best performance, which indicates that the local quantization error is not the optimal calibration strategy for detection models. It reminds us to rethink the calibration principles for regression models to also consider the global quantization error.

Observation 3: Regressor has non-uniform weight distributions, which differs from the classifier. To analyze the difference in weight distribution between classifier and regressor, we further visualize them in Figure 4 with truncated positions selected by minimized quantization error. As shown in the figure, it is evident that the weight distribution of the classifier and regressor is significantly different. The classifier's weight distribution approximates the uniform distribution with few margins. However, the regressor's appears non-uniform, with more values concentrated to the middle. Furthermore, when taking the minimal quantization error as the calibration metric, a large number of weights would be truncated in the regressor (see red points in the figure), resulting in performance degradation. Therefore, it is desirable to design a novel quantizer tailored to the unique distribution of regressors.

3.2. Theoretical Analysis

In the following, we provide a theoretical analysis to explain the cause of non-uniform weight distribution.

To simplify the demonstration, we take a dataset (X, Y) and consider the simple regression function: $f(X) = XW^\top + b$. We use distance-based metrics such as least squares or absolute values to train the network weight W. We expect the weights to maximize the probability of getting the *gt* (ground truth), equivalent to minimizing the distance between the model output and the *gt*:

$$W^* = \arg\max_W \left(P\left(Y|X,W\right)\right) = \arg\min_W \left(\|f(X)-Y\|_p\right), \tag{1}$$

where $p \in \{1,2\}$ represents the least absolute and least squares value. It is easy to derive that the predicted $f(X)$ has an zero-mean error towards Y, *i.e.*, $P(Y_i|X_i,W) = \frac{1}{\lambda_1}\exp\left(-\frac{\|f(X_i)-Y_i\|_p}{\lambda_2}\right)$, where λ_1 and λ_2 describe the probability density function of Y.

Assuming that the weight W is an unknown variable. Therefore, maximizing the posterior probability density of W can be $P(W|X,Y) \propto P(Y|X,W)P(W)$, where $P(W)$ is the priori of the distribution of W. We usually initialize W from the uniform or Gaussian probability. We take the uniform distribution as the priori to simplify the demonstration. The case of non-uniform distribution is in Supplementary **??**. Since each data in (X, Y) is independent, the posterior probability of W is

$$P(W|X,Y) \approx \prod_i \frac{1}{\lambda_1}\exp\left(-\frac{\|f(X_i)-Y_i\|_p}{\lambda_2}\right). \tag{2}$$

It indicates that given training data (X, Y), the weight W have an arbitrary distribution that is more likely to gather around the center. Typically, when $p = 1, \lambda_2 = 2\lambda_1$, it is a standard Laplace distribution. When $p = 2, \lambda_2 = \lambda_1^2/\pi$, it is a Gaussian distribution. And p can be bigger than two in the loss function. As for the classifier, the prediction output is the probability of independent events. It is irrelevant to distance but relative to magnitude counts. Therefore, the weight in the classifier hardly exhibits the phenomenon of clustering towards the center. However, regression is a task that converges in the continuous numerical space and has geometric meaning from the spatial perspective, such as the distance-based Lp metric in Eq. 1. The exponential function in Eq. 2 indicates the non-uniform weight distribution if no additional regularization is applied.

4. Method

4.1. Overall Framework

Based on above observations and theoretical analysis, we design a novel regression-specialized PTQ framework named Reg-PTQ. Figure 5 shows the overall framework, which consists of the Filtered Global Loss Integration Calibration (FGIC) strategy and Learnable Logarithmic Affine Quantizer (LLAQ). When quantization, we first apply the LLAQ to detection head, then initialize all the scaling factors following the typical search strategy [55, 60], and fine-tune all the learnable parameters by FGIC. We describe the two proposed methods in the following.

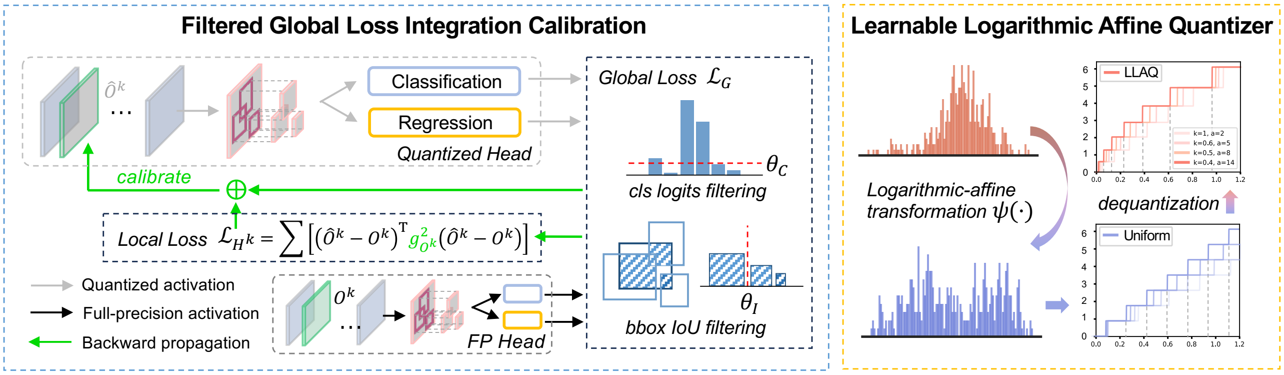

Filtered Global Loss Integration Calibration
$\hat{O}^k$
Classification
Regression
Quantized Head
calibrate
Global Loss $\mathcal{L}_G$
θ_C
cls logits filtering
Local Loss $\mathcal{L}_{H^k} = \sum\left[(\hat{O}^k - O^k)^{\top} g^2_{O^k}(\hat{O}^k - O^k)\right]$
Quantized activation
Full-precision activation
Backward propagation
O^k
FP Head
θ_I
bbox IoU filtering
Learnable Logarithmic Affine Quantizer
LLAQ
k=1, a=2
k=0.6, a=5
k=0.5, a=8
k=0.4, a=14
Logarithmic-affine transformation $\psi(\cdot)$
dequantization
Uniform

Figure 5. Overview of the regression-specialized post-training quantization framework. Left is the Filtered Global Loss Integration Calibration (FGIC), which combines the local and global loss with a two-step filtering mechanism. Right is the Learnable Logarithmic Affine Quantizer (LLAQ), which projects variables to the logarithmic space and learns the quantization factors for regression structures.

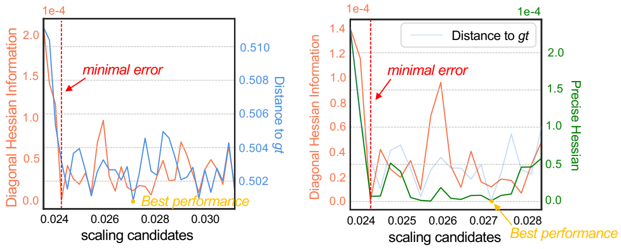
4.2. Filtered Global Loss Integration Calibration

As discussed in **Observation 2**, minimizing the local quantization error cannot guarantee the optimal scaling factor for quantization. To overcome this problem, a typical practice is to use the Hessian-guided metric calibration. We use Diagonal Fisher Information to accelerate the calculation of the Hessian matrix, which is one of the widely used approximation methods [24, 60]:

$$\mathcal{L}_{H^k} = \sum_i \left[(\hat{O}_i^k - O_i^k)^{\top} \left(\frac{\partial \mathcal{L}}{\partial O_i^k} \right)^2 (\hat{O}_i^k - O_i^k) \right], \quad (3)$$

where O_k and $\hat{O}_k$ are the output of the k-th layer in full-precision and quantized networks, respectively. $\mathcal{L}_{H^k}$ denotes Hessian-guided loss which accumulates the second-order term in Taylor expansion of all the values.

However, directly using existing Hessian-guided metrics does not work well on detector quantization. Figure 6 (a) shows the curves of Diagonal Fisher Information (red line) and the performance of the quantized regression model (blue line) under different scaling candidates. We find that the scaling factor with minimal Hessian metric also cannot guarantee the best performance. Since the toy model has only <1k parameters in each layer, it is practical to calculate the precise Hessian matrix $(w - \hat{w})^{\top} H^{(w)} (w - \hat{w})$ using the second-order gradient of w in the neural network. In Figure 6 (b), we zoom in on a clip of scaling candidates and compare the approximated (red line) and precise Hessian-guided metrics (green line). We observe that the precise Hessian is consistent with the model performance, while the approximate Diagonal Hessian Information fails to reflect the impact of quantization. It indicates that the approximation error brought by Hessian calculation cannot be ignored in the regression tasks.



(a) Diagonal Hessian Information and distance to *gt* (ground truth)

(b) Diagonal Hessian Information and precise Hessian

Figure 6. Quantization error measured by Diagonal Fisher Information (red line) and the performance of regressor (blue line) and the precise Hessain (green line).

Therefore, we first introduce a Global-Loss Integration Calibration (GIC) strategy, which combines the local layerwise/block-wise reconstruction loss $\mathcal{L}_H$ and the global regression-aware loss $\mathcal{L}_G$ to finetune the parameters. Typically, $\mathcal{L}_G$ compares the head output between full-precision and quantized detectors. It consists of classification loss $\mathcal{L}_{cls}$ by the cross-entropy (CE) of classification logits y and $\hat{y}$, and the regression loss $\mathcal{L}_{reg}$ by L_p loss of regressed bounding boxes b and $\hat{b}$, respectively:

$$\mathcal{L}_G = \frac{1}{n} \sum_{i=1}^{n} \left(L_{CE}(y_i, \hat{y}_i) + \lambda L_p(b_i, \hat{b}_i) \right), \quad (4)$$

where λ is used to balance the two losses, n is the number of predicted bounding boxes. Therefore, the complete loss to calibrate the k-th layer is:

$$\mathcal{L}_{tot^k} = \mathcal{L}_{H^k} + \mathcal{L}_G. \quad (5)$$

However, integrating the global loss $\mathcal{L}_G$ is non-trivial. The detection head outputs a large amount of bounding boxes with classification scores, including those with low confidence scores and low IoUs. These false positive boxes will introduce useless information or even harmful noise in optimizing the scaling factors. Specifically, boxes with low confidence scores will be ignored in the post-process, so they bring useless information in gradients. And boxes with low IoUs from quantized and original networks have high probabilities that they may correspond to different objects. Directly aligning them is also unreasonable.

To solve the two issues, we propose a two-step bounding boxes filtering mechanism to select high-confidence and high-intersection boxes. As shown in Figure 5, when using the full-precision detector to calibrate the quantized one, we only preserve the boxes with high classification confidence above a predefined threshold θ_C. In this way, we can effectively remove the bounding boxes with low confidence scores. After that, we calculate the Intersection-of-Union (IoU) of corresponding boxes from the quantized and full-precision models filtered by the first step. IoUs larger than the threshold θ_I are more likely to bound the same object and will be preserved to calculate the global loss. Moreover, those that have low intersections will be removed since they may provide meaningless information or even perturbation to the finetuning. Let us suppose $b, \hat{b}$ are the bounding boxes obtained from full-precision and quantized detectors. The whole process can be expressed as:

$$b' = b \cdot \mathcal{I}_{HC} \cdot \mathcal{I}_{HI}, \quad \hat{b}' = \hat{b} \cdot \mathcal{I}_{HC} \cdot \mathcal{I}_{HI}, \tag{6}$$

where $\mathcal{I}_{HC}$ and $\mathcal{I}_{HI}$ are position indicators indicating whether the bounding box b and $\hat{b}$ is filtered:

$$\mathcal{I}_{HC} = \begin{cases} 1, & \text{if } y \geq \theta_C \\ 0, & \text{otherwise}, \end{cases} \quad \mathcal{I}_{HI} = \begin{cases} 1, & \text{if IoU}(b, \hat{b}) \geq \theta_I \\ 0, & \text{otherwise}. \end{cases} \tag{7}$$

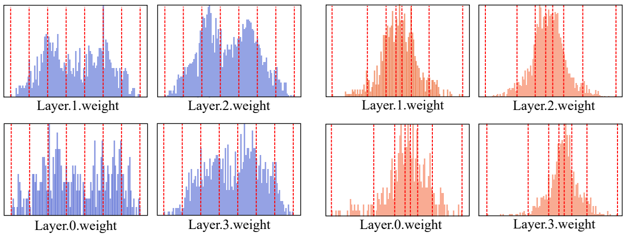
Therefore, the global loss in Eq. 4 becomes

$$\begin{aligned} \mathcal{L}_G &= \frac{1}{n} \sum_{i=1}^{n} \Big(L_{CE}(y_i, \hat{y}_i) + \lambda L_p(b_i^{'}, \hat{b}_i^{'}) \Big), \\ &= \frac{1}{n} \sum_{i=1}^{n} \Big(L_{CE}(y_i, \hat{y}_i) + \lambda L_p(b_i \cdot \mathcal{I}_{HC} \cdot \mathcal{I}_{HI}, \hat{b}_i \cdot \mathcal{I}_{HC} \cdot \mathcal{I}_{HI}) \Big). \end{aligned} \tag{8}$$

In this way, we can filter out the negative effect from the low-confidence and low-intersection bounding boxes in the calibration process, which can provide more accurate gradients and help find optimal scaling factors for quantization. The complete calibration process is in Algorithm 1. Φ and ϕ^k are the full-precision network and a single k-th layer. $\hat{\Phi}$ and $\hat{\phi}^k$ are the quantized counterparts. $hook(\cdot)$ is the hook function in forward and backward propagation. N_i is the total number of iterations.

Algorithm 1 Local and Global Loss-Alignment Calibration

```
Input: Full-precision detector $\Phi$, calibration set $\mathbf{S}_{calib}$, $\mathbf{S}_{prep} = \emptyset$.
Output: Quantized model $\hat{\Phi}$.
for $\phi_q^k$ in $\Phi_q$ do
    $\hat{\phi}_q^k \leftarrow \phi_q^k$.open_quant()
    for s in $\mathbf{S}_{calib}$ do
        $(y, b), (\hat{y}, \hat{b}) \leftarrow \Phi(\mathbf{s}), \hat{\Phi}(\mathbf{s})$
        $(I^k, O^k), (\hat{I}^k, \hat{O}^k) \leftarrow hook(\phi^k), hook(\hat{\phi}^k)$
        $g_{o^k} \leftarrow \frac{\partial \mathcal{L}_G}{\partial \hat{O}^k}$
        $\mathbf{S}_{prep} \leftarrow \mathbf{S}_{prep} \cup \{(\mathbf{s}, y, b, O^k, \hat{I}^k, g_{o^k})\}$
    end for
    for $i \leftarrow 1$ to $N_i$ do
        sampling $\mathbf{s}, y, b, O^k, \hat{I}^k, g_{o^k}$ from $\mathbf{S}_{prep}$
        $\hat{y}, \hat{b}, \hat{O}^k \leftarrow \hat{\Phi}(\mathbf{s}), \hat{\phi}^k(\hat{I}^k)$
        use Eq. 3 to update $\mathcal{L}_{H^k}$
        use Eq. 8 to update $\mathcal{L}_G$
        $\mathcal{L}_{tot} \leftarrow \mathcal{L}_{H^k} + \mathcal{L}_G$
        $\mathcal{L}_{tot}$.backward()
    end for
end for
```



(a) Weight distribution after logarithmic-affine

(b) Weights after dequantization

Figure 7. Visualizations of the weight (a) after logarithmic-affine transformation and (b) in the original linear space. It can be seen that weight in (a) is more uniformly distributed. The red lines are the quantization levels.

4.3. Learnable Logarithmic Affine Quantizer

As discussed in **Observation 3** in Sec. 3.1 and theoretical analysis in Sec. 3.2, the regression of object location is typically trained by distance-based loss functions, in which the weight distribution is more likely to gather around and becomes a *quasi* Laplace or Gaussian distribution, and is not friendly for uniform quantizer.

Therefore, we propose the Learnable Logarithmic-Affine Quantizer (LLAQ) to handle the non-uniform distribution. Specifically, consider a variable with Laplace distribution $f(x|\mu, \lambda) = \frac{1}{2\lambda} \exp\left(-\frac{|x-\mu|}{\lambda}\right)$, where μ, b are location and scale parameters. We first divide the parameters into two parts according to the position parameter μ, and apply a logarithmic-affine transformation $\psi(\cdot)$ to project them into a logarithmic space:

$$\psi(x) = k^* \text{log} x + a^*, \tag{9}$$

where k^*, a^* are learnable scale and offset that we use k^+, a^+ when $x \geq \mu$ and k^-, a^- otherwise. We update k^+, a^+, k^-, a^- during finetuning to learn better logarithmic-affine transformations. Theoretically, applying $\psi(\cdot)$ on Laplace probability density function, we have

$$\psi(f(x|\mu, \lambda)) = \begin{cases} \frac{k^+}{\lambda}(\mu - x) - k^+ \text{log} 2\lambda + a^+ & \text{if } x \geq \mu, \\ \frac{k^-}{\lambda}(x - \mu) - k^- \text{log} 2\lambda + a^- & \text{otherwise}. \end{cases} \tag{10}$$

Method	#Bit(W/A)	RetinaNet		YOLOF	Faster RCNN		Mask RCNN	
		ResNet-50	ResNet-101	ResNet-50	ResNet-50	ResNet-101	ResNet-50	ResNet-101
Full-precision	32/32	37.4	38.9	37.5	38.5	39.8	39.2	40.8
BRECQ	2/4	14.0	18.7	10.8	12.5	13.0	11.0	12.0
PD-Quant	2/4	19.3	20.6	15.4	1.7	6.1	11.8	10.8
QDrop	2/4	19.9	22.9	17.4	17.8	19.9	18.2	19.9
Reg-PTQ (Ours)	**2/4**	**23.9**	**24.8**	**19.3**	**19.1**	**21.5**	**19.1**	**20.7**
AdaRound	3/3	19.3	20.7	7.7	21.2	22.8	21.6	22.6
AdaQuant	3/3	21.1	19.9	13.3	4.8	5.8	4.5	4.4
BRECQ	3/3	22.8	24.6	18.4	16.7	16.5	15.9	15.2
PD-Quant	3/3	24.5	25.6	22.2	14.0	14.0	18.7	17.3
QDrop	3/3	26.5	26.8	25.8	23.6	24.1	24.4	24.7
Reg-PTQ (Ours)	**3/3**	**28.1**	**28.3**	**27.3**	**28.1**	**29.1**	**28.4**	**28.8**
AdaRound	4/4	20.5	20.8	17.1	0.6	23.8	24.3	24.8
AdaQuant	4/4	33.5	34.5	25.6	12.8	14.5	12.0	14.6
BRECQ	4/4	34.2	35.8	29.0	28.8	30.8	31.7	30.1
PD-Quant	4/4	33.2	33.4	31.4	25.7	28.3	27.6	27.5
QDrop	4/4	34.1	35.1	33.4	33.7	34.4	34.5	35.6
SubSetQ	4/4	33.4	35.0	31.8	33.3	35.4	34.9	36.8
Reg-PTQ (Ours)	**4/4**	**36.7**	**35.9**	**34.3**	**36.7**	**36.2**	**36.4**	**37.2**
AdaQuant	4/8	36.5	38.1	35.0	16.9	19.2	14.2	18.4
BRECQ	4/8	36.8	38.6	36.2	20.0	22.0	21.2	23.4
PD-Quant	4/8	36.8	38.5	36.5	24.1	24.2	27.4	26.9
QDrop	4/8	37.0	38.5	36.7	37.6	38.9	38.2	39.9
SubSetQ	4/8	36.7	38.3	36.2	36.1	38.7	38.1	39.8
Reg-PTQ (Ours)	**4/8**	**37.4**	**38.6**	**36.8**	**37.8**	**39.1**	**38.3**	**40.0**

Table 1. Comparison with other PTQ methods on various detectors with ResNet-50/101 as the backbone on COCO dataset.

It means that by Eq. 9, we transform the probability density function from an exponential function to a segmented linear space.

We apply the transformation on the weights of the toy regressor and visualize them in Figure 7. It can be seen that the logarithmic-affine transformation makes the parameters close to a uniform distribution compared to that in Figure 4 (b). It satisfies the presupposition of uniform quantization that the variables are expected to spread evenly. And then, we apply uniform quantization on them (red dotted lines in Figure 7 (a)). We prestore the quantized $\lfloor\psi(W)\rceil$ and $\boldsymbol{s}_w^*$ in checkpoint to avoid on-the-fly computation. In inference, the product of activation and weight is $X^\top W \approx X^\top 2^{\lfloor\psi(W)\rceil}\boldsymbol{s}_w^* = \boldsymbol{s}_w^* X^\top << \lfloor\psi(W)\rceil$, where $\boldsymbol{s}_w^* = \frac{1}{HW}\Sigma_i\Sigma_j\frac{W_{ij}}{W_{ij}^{k^*}2^{a^*}}$ are channel-wise vectors. $<<$ can be implemented by `BitShift` operator.

5. Experiments

5.1. Settings

Detection Task and Networks: To demonstrate the versatility of our Reg-PTQ, we evaluate our Reg-PTQ framework on the most representative one-stage and two-stage detection frameworks in the literature, including RetinaNet [28], YOLOF [2], Faster RCNN [44] and Mask RCNN [18] with ResNet-50/101 as backbones. We showcase the mAP result of the above detectors on the large-scale COCO object detection dataset [27]. Due to limited space, we also provide experimental results on small-scale PASCAL VOC [8] dataset and transformers-architecture detectors in the supplementary material.

Implementation Details: We follow [56] to adopt the widespread PTQ pipeline, and follow [36] to use the standard pipeline for weight-tuning. We adopt the uniform settings for all methods, use block-wise calibration for the backbone and layer-wise calibration for other structures, and finetune each block or layer for 2k iterations with a batch size of 256. The quantized bit-width is denoted as WwAa, meaning w-bit weight and a-bit activation. Unlike previous works that only quantize the backbone, we quantize all the intermediate layers to WwAa, the first layer to 8-bit, and keep the last prediction layer to full-precision. Besides, additive operators are full-precision and Batch Normalization is folded into previous convolution. We dubbed it full quantization.

5.2. Results on COCO Object Detection Dataset

We compare our Reg-PTQ framework with several classical and recent PTQ methods including AdaRound [36], AdaQuant [21], BRECQ [24], QDrop [55], PD-Quant [31] and SubSetQ [41]. Table 1 showcases results on COCO object detection dataset. We evaluate under diverse bit-width settings, including W2A4, W3A3, W4A4 and W4A8. We do not report the results of baseline methods if their performance collapse after quantization. Numbers in bold are the results of our method.

As Table 1 shows, Reg-PTQ has consistent improvements over various detection models and bitwidth. It achieves almost lossless accuracy under W4A8, which is 38.6% on RetinaNet ResNet-101, with only a 0.3% drop

baseline: 23.0		θ_C				
		0 (w/o)	5e-5	2e-4	1e-3	1e-2
θ_I	0 (w/o)	23.5	23.9	23.8	23.5	23.2
	0.1	23.8	23.9	23.9	23.8	23.5
	0.5	23.8	23.8	23.8	23.8	23.2

Table 2. Ablation study of FGIC and sensitivity analysis of its hyperparameters, θ_C and θ_I, on RetinaNet ResNet-50 on COCO under W2A4 quantization. Baseline means solely using local loss.

Model	Quantizer	W2A4	W3A3	W4A4
One-Stage	Uniform	23.0	27.2	35.2
(Bbox Head)	LLAQ	23.6	28.0	35.7
Two-Stage	Uniform	22.3	28.8	34.3
(Rpn+Roi Heads)	LLAQ	23.7	31.7	36.4

Table 3. Comparison of uniform and LLAQ quantizers under various bitwidth on COCO. Models used here are RetinaNet ResNet-50 and Faster RCNN ResNet-50.

#Bit$_{\text{(W/A)}}$	**Quantize Backbone & Neck**		**Fully Quantize**	
	FLOPs (G)	**Storage (M)**	**FLOPs (G)**	**Storage (M)**
2/4	25.48	21.78	12.14	5.97
4/4	35.24	23.46	22.65	8.70
4/8	54.75	23.46	43.95	8.70

(a) Faster RCNN ResNet-50. The full-precision one has 171.8 GFLOPs and 46.91 M Storage while processing one sample.

#Bit$_{\text{(W/A)}}$	**Quantize Backbone & Neck**		**Fully Quantize**	
	FLOPs (G)	**# Storage (M)**	**FLOPs (G)**	**Storage (M)**
2/4	8.06	37.11	7.89	14.9
4/4	14.73	39.08	14.46	18.42
4/8	28.07	39.08	27.84	18.42

(b) RetinaNet ResNet-50. The full-precision one has 108.10 GFLOPs and 66.60 M Storage while processing one sample.

Table 4. The FLOPs (G) and the Storage (M) of different detectors under different bit-width settings.

compared with the full-precision counterpart. And the performance of Faster RCNN is also noteworthy with only 0.7% drops, which is 37.8% and 39.1% using ResNet-50/101 as backbones. Under more challenging W4A4 setting, Reg-PTQ also achieves comparable performance that less than 1.0% drop when quantizing RetinaNet ResNet50, and about 3.0% drop on other detection models.

We highlight that Reg-PTQ framework outperforms other PTQ methods by a wide margin, especially under lower bit-width, such as W2A4 and W3A3. For instance, for one-stage detectors, Reg-PTQ achieves 23.9% on RetinaNet ResNet-50 under W2A4, up to 4.0% higher than current state-of-the-art methods. As for two-stage detectors, Reg-PTQ achieves remarkable 4.0-5.0% improvements on Faster RCNN and Mask RCNN under W3A3. Experimental results forcefully demonstrate that our Reg-PTQ framework retains the accuracy for quantized detectors, which means that the proposed two regression-specialized techniques are more suitable for quantizing detection models.

5.3. Ablation Study

We perform detailed ablation study and analysis for the proposed techniques, *i.e.*, FGIC and LLAQ.

Effect of FGIC. We use two hyperparameters θ_C and θ_I as the thresholds to filter out the low confidence and low IoU boxes. To better understand the effect, in Table 2, we report the performance of FGIC, and the impact of different θ_C and θ_I. From the results, we observe: (1) Compared with solely using local loss, introducing global loss without filtering can improve 0.5% of the performance. (2) It is effective to filter out the low-confidence and low-intersection bounding boxes in calculating the loss, which is 0.4% better than that without using the filtering strategy. (3) Our Reg-PTQ is not sensitive to the value of θ_C and θ_I. We find that when $\theta_C = 2e-4, \theta_I = 0.1$, our Reg-PTQ can achieve the best result. Therefore, we use these values as default.

Effect of LLAQ. We report the results when applying uniform quantizer and LLAQ quantizers on detection head only for one/two-step detectors, and the results are shown in Table 3. Compared with the others, LLAQ exhibits consistent improvement in various bit-width settings and detection heads. It shows impressive improvements, especially under lower bit-width settings. For example, it increases 1.6 mAP under W2A4 for the one-stage detector, and 2.9 mAP under W3A3 for the two-stage detector.

5.4. Efficiency

We evaluate the computational complexity and storage of fully quantized detectors using thop [2]. We take the Faster RCNN and RetinaNet with ResNet-50 as examples. Table 4 compares the FLOPs (G) and storage (M) between quantizing backbone and neck and full quantization. Full quantization pushes the acceleration and compression ratio to the impressive 14.2× and 7.8× on Faster RCNN ResNet-50 under W2A4, which has an additional 2.1× and 3.6× reduction in FLOPs and storage compared to only quantize the backbone and neck. For RetinaNet, the computation decrease in FLOPs is not noticeable, but the storage consumption is significantly reduced. Therefore, detectors can benefit a lot from full quantization and have great potential to achieve efficient inference if deployed on hardware.

6. Conclusion

In this paper, we devote ourselves to the regression-specialized PTQ methods. We reveal the fundamental cause of quantizing detectors and propose the first universal full quantization framework for detection named Reg-PTQ. The impressive accuracy and efficiency performance demonstrate the effectiveness of our method. We provide primary attempts and insights in quantizing the regression structures, making it promising to produce and implement full quantized detectors in real object detection scenarios.

[2] https://github.com/Lyken17/pytorch-OpCounter

References

[1] Abhishek Balasubramaniam, Febin Sunny, and Sudeep Pasricha. R-toss: A framework for real-time object detection using semi-structured pruning. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2023. 1

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019. 7

[3] Yun-Hao Cao, Peiqin Sun, Yechang Huang, Jianxin Wu, and Shuchang Zhou. Synergistic self-supervised and quantization learning. In *European Conference on Computer Vision*, pages 587–604. Springer, 2022. 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[5] Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, and Jian Sun. You only look one-level feature. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[6] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5380–5388, 2022. 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 7

[9] Angela Fan, Pierre Stock, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand Joulin. Training with quantization noise for extreme model compression. *arXiv preprint arXiv:2004.07320*, 2020. 2

[10] Ross Girshick. Fast r-cnn. In *IEEE ICCV*, 2015. 1

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE CVPR*, 2014. 1

[12] Ruihao Gong, Yang Yong, Zining Wang, Jinyang Guo, Xiuying Wei, Yuqing Ma, and Xianglong Liu. Fast and controllable post-training sparsity: Learning optimal sparsity allocation with global constraint in minutes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12190–12198, 2024. 2

[13] Ofir Gordon, Hai Victor Habi, and Arnon Netzer. Eptq: Enhanced post-training quantization via label-free hessian. *arXiv preprint arXiv:2309.11531*, 2023. 2

[14] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12259–12269, 2021. 1

[15] Jinyang Guo, Jiaheng Liu, and Dong Xu. Jointpruning: Pruning networks along multiple dimensions for efficient point cloud processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 1

[16] Jinyang Guo, Jiaheng Liu, Zining Wang, Yuqing Ma, Ruihao Gong, Ke Xu, and Xianglong Liu. Adaptive contrastive knowledge distillation for bert compression. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8941–8953, 2023.

[17] Jinyang Guo, Dong Xu, and Wanli Ouyang. Multidimensional pruning and its extension: A unified framework for model compression. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 1

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 7

[19] Xijie Huang, Zhiqiang Shen, Shichao Li, Zechun Liu, Hu Xianghong, Jeffry Wicaksana, Eric Xing, and Kwang-Ting Cheng. Sdq: Stochastic differentiable quantization with mixed precision. In *International Conference on Machine Learning*, pages 9295–9309. PMLR, 2022. 2

[20] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In *International Conference on Machine Learning*, pages 4466–4475. PMLR, 2021. 2

[21] Divyansh Jhunjhunwala, Advait Gadhikar, Gauri Joshi, and Yonina C Eldar. Adaptive quantization of model updates for communication-efficient federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3110–3114. IEEE, 2021. 2, 7

[22] Krizhevsky, Alex, Sutskever, Ilya, Hinton, and E. Geoffrey. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 2017. 1

[23] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *IEEE CVPR*, 2019. 1

[24] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021. 2, 5, 7

[25] Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-vit: Accurate and fully quantized low-bit vision transformer. *Advances in Neural Information Processing Systems*, 35:34451–34463, 2022. 2

[26] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17227–17236, 2023. 2, 3

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 7

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[30] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Fully quantized vision transformer without retraining. *arXiv preprint arXiv:2111.13824*, 2021. 1

[31] Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Pd-quant: Post-training quantization based on prediction difference metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24427–24437, 2023. 7

[32] Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20321–20330, 2023. 3

[33] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2

[34] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

[35] Sachin Mehta, Marjan Ghazvininejad, Srinivasan Iyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Delight: Very deep and light-weight transformer, 2020. 1

[36] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization, 2020. 2, 7

[37] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, pages 16318–16330. PMLR, 2022. 2

[38] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, 110(11-12):3245–3262, 2021. 2

[39] Lin Niu, Jiawei Liu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Improving post-training quantization on object detection with task loss-guided lp metric, 2023. 3

[40] Sangyun Oh, Hyeonuk Sim, Jounghyun Kim, and Jongeun Lee. Non-uniform step size quantization for accurate post-training quantization. In *European Conference on Computer Vision*, pages 658–673. Springer, 2022. 2

[41] Sangyun Oh, Hyeonuk Sim, Jounghyun Kim, and Jongeun Lee. Non-uniform step size quantization for accurate post-training quantization. In *European Conference on Computer Vision*, pages 658–673. Springer, 2022. 7

[42] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *IEEE CVPR*, 2019. 1

[43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2, 3

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. 2015. 2, 7

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 1

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE CVPR*, 2015. 1

[48] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2

[49] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023. 2

[50] Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world, 2021. 1

[51] Mingze Wang, Huixin Sun, Jun Shi, Xuhui Liu, Baochang Zhang, and Xianbin Cao. Q-yolo: Efficient inference for real-time object detection, 2023. 3

[52] Yiru Wang, Weihao Gan, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *IEEE ICCV*, 2019. 1

[53] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *ICCV*, 2019. 1

[54] Ziwei Wang, Ziyi Wu, Jiwen Lu, and Jie Zhou. Bidet: An efficient binarized object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2049–2058, 2020. 2

[55] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *International Conference on Learning Representations*, 2021. 2, 4, 7

[56] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *International Conference on Learning Representations*, 2022. 3, 7

[57] Yanlu Wei, Renshuai Tao, Zhangjie Wu, Yuqing Ma, Libo Zhang, and Xianglong Liu. Occluded prohibited items de-

tection: An x-ray security inspection benchmark and de-occlusion attention module. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 138–146, 2020. 1

[58] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. Mobiledets: Searching for object detection architectures for mobile accelerators. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3825–3834, 2021. 1, 2

[59] Sheng Xu, Yanjing Li, Mingbao Lin, Peng Gao, Guodong Guo, Jinhu Lu, and Baochang Zhang. Q-detr: An efficient low-bit quantized detection transformer, 2023. 3

[60] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization framework for vision transformers. *arXiv preprint arXiv:2111.12293*, 2021. 1, 2, 4, 5

[61] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019. 1