Enhancing Post-training Quantization Calibration through Contrastive Learning

Yuzhang Shang[1,2], **Gaowen Liu**[2], **Ramana Rao Kompella**[2], **Yan Yan**[1,†]

[1]Illinois Institute of Technology, [2]Cisco Research

yzshawn@outlook.com, {gaoliu, rkompell}@cisco.com, yyan34@iit.edu

Abstract

Post-training quantization (PTQ) converts a pre-trained full-precision (FP) model into a quantized model in a training-free manner. Determining suitable quantization parameters, such as scaling factors and zero points, is the primary strategy for mitigating the impact of quantization noise (calibration) and restoring the performance of the quantized models. However, the existing activation calibration methods have never considered information degradation between pre- (FP) and post-quantized activations. In this study, we introduce a well-defined distributional metric from information theory, mutual information, into PTQ calibration. We aim to calibrate the quantized activations by maximizing the mutual information between the pre- and post-quantized activations. To realize this goal, we establish a contrastive learning (CL) framework for the calibration, where the quantization parameters are optimized through a self-supervised proxy task. Specifically, by leveraging CL during the PTQ calibration, we can benefit from pulling the positive pairs of quantized and FP activations collected from the same input samples, while pushing negative pairs from different samples. Thanks to the ingeniously designed critic function, we avoid the unwanted but often-encountered collision solution in CL, especially in calibration scenarios where the amount of calibration data is limited. Additionally, we provide a theoretical guarantee that minimizing our designed loss is equivalent to maximizing the desired mutual information. Consequently, the quantized activations retain more information, which ultimately enhances the performance of the quantized network. Experimental results show that our method can effectively serve as an add-on module to existing SoTA PTQ methods.

1. Introduction

To meet the growing demand for equipping deep neural networks in resource-constrained edge devices, researchers have developed network quantization techniques [9], in which high-precision parameters and activations are con-

---

†Corresponding author

verted into low-precision ones. Quantization methods fall into two main categories: post-training quantization (PTQ) [19–23, 33] and quantization-aware training (QAT) [14, 30–33, 43]. QAT involves retraining the model on the labeled training dataset, a process that can be both time-consuming and computationally demanding. In contrast, PTQ only necessitates a small number of unlabeled calibration samples to quantize the pre-trained models, eliminating the need for retraining. This makes PTQ a practical choice for rapid deployment scenarios.

Existing PTQ methods have shown promising results in maintaining good prediction accuracy even when using 4-bit or 2-bit quantization. The majority of these top-performing methods owe their success to meticulous quantization parameter selection. Metrics such as Mean Squared Error (MSE) [5, 19, 40] and cosine distance [41] between the pre- and post-quantization activations in individual layers or modules are commonly used to find the most suitable scaling factors for quantization. These methods are typically referred to as quantized activation calibration techniques. However, existing calibration methods have not taken into account the loss of information during the transition from pre- to post-quantized activation.

In the context of PTQ, we desire the FP and quantized activations (considered as two random variables) to share as much information as possible, as quantized activations are directly derived from their FP counterparts. Mutual information [18] can measure the amount of information that knowing one of these variables provides about the other. Therefore, in this study, our focus is on the mutual information between FP and quantized activations. Our goal is to optimize PTQ calibration process (particularly, learn the suitable quantization parameters), by maximizing the mutual information between these two types of activations. By doing so, we aim to preserve as much of the information from the FP activations as possible in the quantized activations, thereby minimizing the loss in model performance due to quantization. Our proposed method is termed "Enhancing PTQ Calibration through Contrastive Learning", abbreviated as `CL-Calib`. Specifically, we utilize contrastive learning (CT) in the PTQ calibration process. We align positive pairs of quantized and FP activations
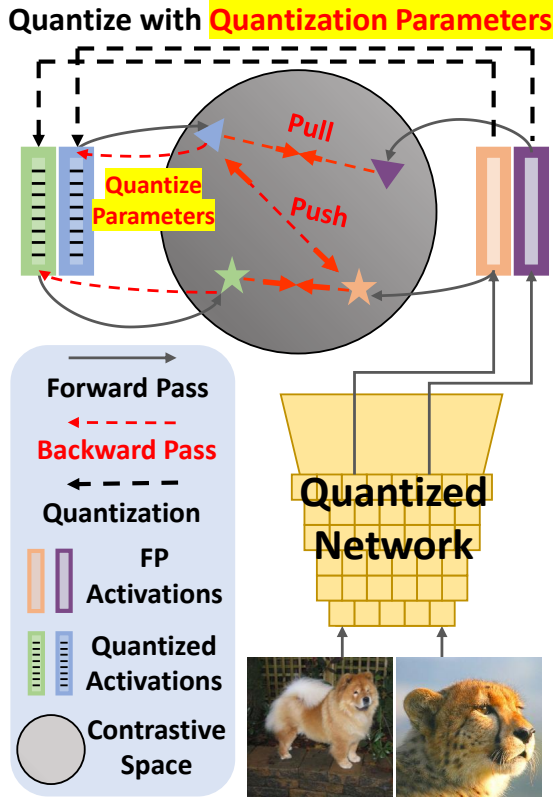
Figure 1. General Idea: calibrating quantized activation distribution is a contrastive learning manner. By embedding the activations into a contrastive space, the quantization parameters can be optimized through the pair correlation within the contrastive learning task. *Note that, although this demonstration involves two images, the actual number of images is large.*

produced from identical input samples, while simultaneously distancing negative pairs from different samples (see Fig. 1). The collision solution is a common occurrence in CL, often resulting from an insufficiency of samples for each class, and exacerbated by the extreme limitations of calibration data within the PTQ setting. To circumvent this collision, we meticulously design a critic function that leverages the pre-trained full-precision (FP) model. Alongside this straightforward explanation, we also provide a theoretical justification for CL-Calib from the perspective of mutual information. In summary, the quantized activations calibrated by CL-Calib can retain more information, ultimately leading to better performance of the quantized network.

Overall, the contributions of this paper are three-fold: **(i)** Under the PTQ calibration framework, we design a self-supervised learning proxy task to optimize the quantization parameters (*e.g.*, scaling factors) by introducing contrastive learning; **(ii)** We provide a mutual information maximization perspective to understand the quantization problem, and proof that our method can maximize the mutual information between the quantized and full-precision activations; **(iii)** Experimental results demonstrate that our PTQ

calibration method, CL-Calib, can function as a plug-and-play module for existing state-of-the-art PTQ methods.

2. Preliminary and Background

In this section, we revisit the idea of network quantization and the overarching framework of post-training quantization (PTQ), especially the PTQ activation calibration.

Basic Notations. We first define a K-layer Multi-Layer Perceptron (MLP). For simplification, we discard the bias term of this MLP. The network $f(\mathbf{X})$ can be denoted as:

$$f(\mathbf{W}^1, \cdots, \mathbf{W}^K; \mathbf{X}) = (\mathbf{W}^K \cdot \sigma \cdot \mathbf{W}^{K-1} \cdots \cdot \sigma \cdot \mathbf{W}^1)(\mathbf{X}), \quad (1)$$

where $\mathbf{X}$ is the input sample and $\mathbf{W}^k : \mathbb{R}^{d_{k-1}} \longmapsto \mathbb{R}^{d_k} (k = 1, ..., K)$ stands for the weight matrix connecting the $(k-1)$-th and the k-th activation layer, with d_{k-1} and d_k representing the sizes of the input and output of the k-th network layer, respectively. We denote $f^k(\mathbf{W}^k; \cdot)$ as the k-th layer's mapping function. The $\sigma(\cdot)$ function performs element-wise activation operations on the input feature maps. Utilizing the aforementioned predefined notions, we define a *sliced MLP* $f^{[:k]}(\mathbf{x})$ and $f^{[k:]}(\mathbf{x})$, which consists of the first k layers and the last $(K-k)$ layers of $f(\mathbf{x})$, as follows:

$$\mathbf{A}^k = f^{[:k]}(\mathbf{W}^1, \cdots, \mathbf{W}^k; \mathbf{X}) = (\mathbf{W}^k \cdot \sigma \cdots \sigma \cdot \mathbf{W}^1)(\mathbf{X}),$$
$$f^{[k:]}(\mathbf{W}^k, \cdots, \mathbf{W}^K; \mathbf{A}^k) = (\mathbf{W}^K \cdot \sigma \cdots \sigma \cdot \mathbf{W}^k)(\mathbf{A}^k), \quad (2)$$

where $\mathbf{A}^k$ is the k-th layer's output activation. And the MLP f can be seen as a special case in the sliced function sequences $\{f^{[:k]}\}$ and $\{f^{[k:]}\}$ ($k \in \{1, \cdots, K\}$), *i.e.*, $f = f^{[:K]} = f^{[0:]}$. For ease of reference, we use subscripts F and Q for f and $\mathbf{A}$ to represent the full-precision and quantized versions, respectively.

Post-training quantization (PTQ) takes a well-trained full-precision network f_F as input and selects quantization parameters to quantize the weight tensor and activation tensor in each layer for obtaining quantized network f_Q. *To convert a tensor into a quantized tensor, only two quantization parameters are required, i.e., the scaling factor S and the zero point Z.* Consequently, most PTQ methods primarily focus on selecting appropriate quantization parameters [19–23]. One of the most prevalent approaches for selecting parameters is to minimize the error induced by quantization. The quantization process can be formulated as an optimization problem:

$$\arg\min_{S,Z} \mathcal{L}_{quant}, \text{ where } \mathcal{L}_{quant} = \text{Metric}(\mathbf{X}_Q, \mathbf{X}_F), \quad (3)$$

where Metric is the metric function measuring the distance between $\mathbf{X}_Q$ and the full-precision tensor $\mathbf{X}_F$. MSE, cosine distance, L1 distance, and KL divergence are commonly

used metric functions. Formally, after obtaining appr‌ ate S and Z, a full-precision tensor can be converte‌ follows:

$$\mathbf{X}_Q = S(\text{clamp}(\lfloor \frac{\mathbf{X}_F}{S} \rceil - Z, p_{min}, p_{max}) + Z),$$

where $[p_{min}, p_{max}]$ is the quantization range determine‌ bit-width, for 8 bit integer, the range is $[-128, 127]$. that, in this work We only consider uniform unsigned metric quantization, as it is the most widely used quar‌ tion setup, and $\mathbf{X}_Q$ is not in an integer form, but is de‌ tized float value ($\mathbf{X}_Q \approx \mathbf{X}_F$ is easily for alignment in F while $\frac{\mathbf{X}_Q}{S}$ is in integer form).

Activation Calibration. Although the netw‌ weights can be quantized without data by minimizin‌ quantization error (Eqs. 3 and 4), a similar approach c‌ be simply used for activation quantization, as the activa‌ tion tensors are inaccessible without input. To collect the activation tensors, a set of unlabeled input samples (cali‌ bration dataset) is used as the network input. The size of the calibration dataset (*e.g.*, 128 randomly selected images in ImageNet [6]) is significantly smaller than the training dataset. After obtaining the k-th layer's full-precision ac‌ tivation $\mathbf{A}_F^k$, the quantized activation $\mathbf{A}_Q^k$ can be similarly calibrated based on Eqs. 4 and 3.

In general, PTQ quantizes a network in three steps: **(i)** Select which operations in the network should be quantized and leave the other operations in full-precision. For ex‌ ample, some special functions such as softmax and GeLU often takes full-precision [35]. **(ii)** Collect the calibration samples. The distribution of the calibration samples should be as close as possible to the distribution of the real data to avoid over-fitting of quantization parameters on calibra‌ tion samples. **(iii)** Use the proper method to select quan‌ tization parameters for weight and activation tensors. *Re‌ cently, state-of-the-art PTQ works [19, 20, 40] reveal that the bottleneck for further improving the PTQ performance is the activation quantization rather than weight quantiza‌ tion.* They focus on **activation calibration**. Specifically, determining the quantization hyper-parameters (*i.e.*, scaling factor S_a^k and zero point Z_a^k) for quantizing the k-th layer's activation is addressed as an optimization problem:

$$\arg\min_{S_a^k, Z_a^k} \mathcal{L}_{calib}, \text{ where } \mathcal{L}_{calib} = \text{Metric}(\mathbf{A}_Q^k, \mathbf{A}_F^k), \quad (5)$$

in which $\mathcal{L}_{calib}$ is the calibration objective for activation quantiztaion. In this paper, we also focus on enhancing the activation calibration (step **(iii)**). Specifically, we intro‌ duce contrastive learning as a proxy task during the calibra‌ tion phase. By leveraging the proxy task, the quantization parameters are optimized based on self-supervised signals (Sec. 3.1), making the quantized activations more similar to their full-precision counterparts in terms of mutual informa‌ tion (Sec. 3.2).
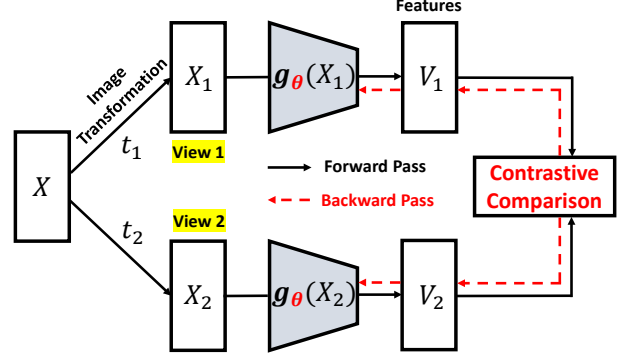

Figure 2. In **contrastive instance learning**, the features produced by different transformations of the same sample are contrasted to each other.

3. Method

In this section, we elucidate the methodology for PTQ calibration via contrastive learning, which we refer to as `CL-Calib`. On the one hand, we straightforwardly demonstrate the construction of a self-supervised learning proxy task within the context of the PTQ calibration frame‌ work, as well as the associated optimization formulation. On the other hand, based on the method, we give a theoret‌ ical explanation for the efficacy of `CL-Calib` through the lens of mutual information maximization.

3.1. Methodology: `CL-Calib`

Instance Recognition (IR) for PTQ Calibration. As pi‌ oneers, [25, 42] propose instance recognition as a self-supervised learning task, categorizing images based on their unique labels. Specifically, for a image $\mathbf{X}$ in the dataset $\mathcal{D}$ following p_{data}, a learnable encoder g_θ mapping the image $\mathbf{X}$ to a feature vector $\mathbf{V} = g(\mathbf{X})$. The resulting vector $\mathbf{V}$ should serve as an accurate representation of $\mathbf{X}$. To accom‌ plish this task, contrastive learning (CL) approaches use a training method that distinguishes a positive from multiple negatives, based on the similarity principle between sam‌ ples as shown in Fig. 2. The InfoNCE [10, 25] loss function, a popular choice for CL, can be expressed as:

$$\mathcal{L}_{contrstive} = \mathbb{E}_{(\mathbf{X}, \mathbf{X}_+) \sim p_{\text{pos}}, \{\mathbf{X}_{-,i}\}_{i=1}^M \overset{\text{i.i.d}}{\sim} p_{\text{data}}}$$
$$\left[-\log \frac{\exp(\mathbf{V} \cdot \mathbf{V}_+/\tau)}{\exp(\mathbf{V} \cdot \mathbf{V}_+/\tau) + \sum_{i=1}^M \exp(\mathbf{V} \cdot \mathbf{V}_{-,i}/\tau)} \right\} \right], \quad (6)$$

where $\mathbf{X}_+$ represents a positive sample for $\mathbf{X}$, and $\mathbf{X}_{-,i}$ represents the i-th negative sample for $\mathbf{X}$. The symbol "$\cdot$" refers to the inner (dot) product, and $\tau > 0$ is a temper‌ ature hyper-parameter. It is worth noting that the embed‌ dings used in the loss function are normalized by L2 nor‌ malization. From a representation learning perspective, we can interpret the aforementioned loss function in an intu‌ itive manner. The term for positive pairs is optimized to accentuate intra-class correlations, while the term for neg‌

ative pairs serves to promote inter-class decorrelation. Because pairs are constructed instance-wisely, the number of negative samples can, theoretically, be as large as the size of the entire training set. Several CL methods [1, 4, 13, 25] have demonstrated the benefits of enhancing the similarity between representations in multiple views as shown in Fig. 2. These methods inspire that by enhancing the correlation between quantized and full-precision (FP) activations, we can improve the representational capacity of quantized networks, thereby improving the performance of the quantized model.

For the calibration of the k-th layer's quantized activation $\mathbf{A}_Q^k$, only two quantization parameters are required: scaling factor S_a^k and zero point Z_a^k are required. After obtaining suitable quantization parameters, $\mathbf{A}_Q^k$ can be quantized as follows:

$$\mathbf{A}_Q^k = S_a^k(\text{clamp}(\lfloor \frac{\mathbf{A}_F^k}{S_a^k} \rceil - Z_a^k, p_{min}, p_{max}) + Z_a^k). \quad (7)$$

The essence of contrastive learning (CL) is to compare different views of the data (typically under various data augmentations) for calculating similarity scores. This CL approach is effective in *defining the view and negative samples* in a manner that can be numerically accessed [1, 4, 11, 13, 25]. Therefore, to learn S_a^k and Z_a^k via CL, we must first define the views, and corresponding positive and negative pairs for comparison. Generally, we collect positive pairs by the quantized and FP activations from the same input sample, while negative pairs are collected from different samples. By optimizing the CL task, we are able to learn the quantization parameters that make the quantized activations contain more information, and hence, more similar to the FP ones in terms of information. This can result in a more effective PTQ calibration.

Definitions: Views and Negative Samples for `CL-Calib`. Notably, during the PTQ calibration phase, we naturally have two views of the data: the quantized activations A_Q and the FP activations A_F. We leverage these two views, together with additional contrastive pairs, to improve the correlation between the quantized and full-precision activations (see Eq. 7 and Fig. 3). Specifically, For a calibration batch with $M + 1$ samples, the samples can be denoted as: $\{\mathbf{X}_i\}(i \in \{1, \cdots, M + 1\})$. We feed a batch of samples to the quantized network $f_Q^{[:k]}$ and obtain $(M + 1)^2$ pairs of activations $(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$, which augments the data for the auxiliary task. We define a pair containing two activations from the same sample as positive pair, *i.e.*, if $i = j$, $(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,+,j}^k)$ and *vice versa* $(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,-,j}^k)$. The core idea of contrastive learning is to discriminate whether a given pair of activation $(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$ is positive or negative with a learnable neural network $d(\cdot, \cdot)$, *i.e.*, estimating the distribution $P(D \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$ with d, in which D is the latent variable determining whether $i = j$ or $i \neq j$. The

d has a form as $d(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) \triangleq \frac{\exp(g(\mathbf{A}_{Q,i}^k) \cdot g(\mathbf{A}_{F,j}^k)/\tau)}{\exp(g(\mathbf{A}_{Q,i}^k) \cdot g(\mathbf{A}_{F,j}^k)/\tau)+1}$. Upon setting up all the prerequisites for CL during the calibration phase, we can define the `CL-Calib` loss as follows:

$$\begin{aligned}\mathcal{L}_{\texttt{CL-Calib}} =& \mathbb{E}_{(\mathbf{A}_Q^k, \mathbf{A}_{F,+}^k)} \left[-\log d(\mathbf{A}_Q^k, \mathbf{A}_{F,+}^k) \right] \\ &+ \mathbb{E}_{\{\mathbf{A}_{F,-,i}^k\}_{i=1}^M} \left[-\log(1 - d(\mathbf{A}_Q^k, \mathbf{A}_{F,-,i}^k)) \right], \end{aligned} \quad (8)$$

in which $\mathbf{A}_Q^k$ is the k-th layer's quantized output, *i.e.*, $\mathbf{A}_Q^k = f_Q^{[:k]}(\mathbf{W}_Q^1, \cdots, \mathbf{W}_Q^k; \mathbf{X})$. Theoretically, minimizing $\mathcal{L}_{\texttt{CL-Calib}}$ is able to produce a binary discriminator $d^\star$, which can classify a given pair $(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$ into positive or negative.

Combining the designed `CL-Calib` loss, the overall calibration objective $\mathcal{L}_{calib}$ can be defined as:

$$\mathcal{L}_{calib} = \mathcal{L}_{quant} + \lambda \mathcal{L}_{\texttt{CL-Calib}}, \quad (9)$$

where $\mathcal{L}_{quant}$ is the activation reconstruction loss (*e.g.*, state-of-the-arts QDrop [40] and PD-Quant [20] in practice), λ is used to control the degree of NCE loss. Straightforwardly, the quantized and FP activations from identical samples can be pulled close, and activations from different samples can be pushed away, which corresponds to the core idea of contrastive learning. Hence, the representation ability of the quantized activations calibrated by our method is enhanced, as demonstrated in Fig. 5.

Network Architecture of mapping function g. Because there is only one instance per class in contrastive instance discrimination, training stability is significantly compromised, resulting in considerable optimization fluctuations. This instability often leads CL algorithms towards collision solutions [1, 4, 13, 25]. In the context of PTQ calibration, the limited availability of calibration data further exacerbates the optimization challenge. We carefully design the mapping network g by leveraging the frozen FP network $f_F^{[k:]}$ as shown in Fig. 3. Specifically, instead of training a fully-learnable network from scratch as most CL methods do, we utilize the existing pre-trained model $f_F^{[k:]}$ as the mapping network. The benefits of using $f_F^{[k:]}$ are three-fold: **(i)** $f_F^{[k:]}$ represents a well-trained mapping from $\mathbf{A}_{F,j}^k$ to the prediction, corresponding to the task itself. Therefore, the need to design a contrastive space is eliminated. Empirically, we have also attempted to use a fully-learnable network as g. However, its optimization fails, mainly due to the limited training data available for traditional CL. **(ii)** The primary goal is to optimize the scaling factor and zero point for the quantization of the k-th layer's activation (involving only two parameters). As such, learning g proves to be a thankless task. **(iii)** Foregoing the learning of g is in line with the application scenarios of PTQ, that is, rapid network deployment.
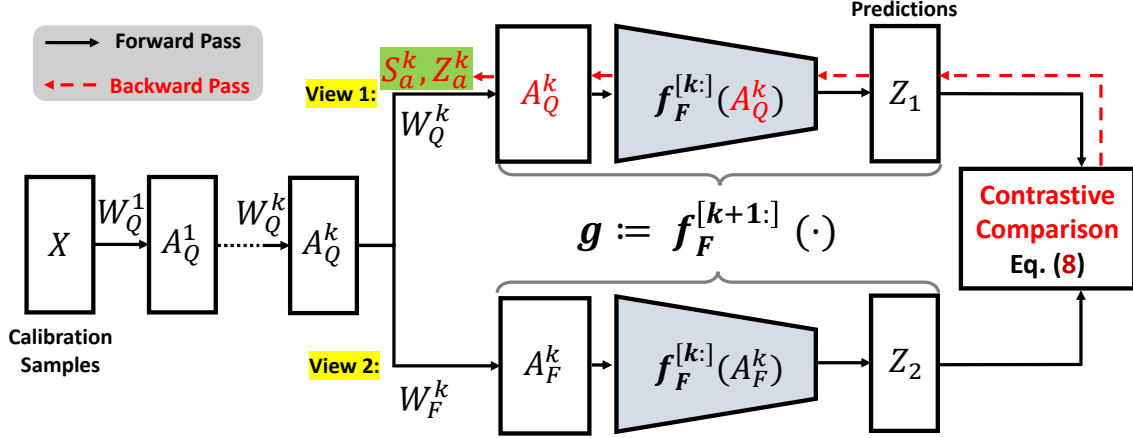
Figure 3. The goal is to optimize the quantization parameters (*i.e.*, scaling factor S_a^k and zero point Z_a^k) for quantizing the activation of k-th layer. Feeding a set of calibration images into a quantized network, we can obtain the full-precision and quantized activations (*i.e.*, A_F^k and A_Q^k). Then, we embed the activations into a prediction space via $g = f_F^{[k:]}$ (with frozen FP parameters). By optimizing the pair correlation within the contrastive learning task in Eq. 8, we can refine the scaling factor for calibrating the activation of the k-th layer.

3.2. Theoretical Guarantee: Mutual Information Maximization

In this section, we provide a theoretical explanation as to why CL-Calib is capable of maximizing the mutual information during the quantization process, *i.e.*, maximizing the mutual information between full-precision and quantized activations for calibration.

Mutual Information and Contrastive Learning. For two discrete variables $\mathbf{X}$ and $\mathbf{Y}$, their mutual information (MI) can be defined as [18]:

$$I(\mathbf{X}, \mathbf{Y}) = \sum_{x,y} P_{\mathbf{XY}}(x,y) \log \frac{P_{\mathbf{XY}}(x,y)}{P_{\mathbf{X}}(x)P_{\mathbf{Y}}(y)}, \quad (10)$$

where $P_{\mathbf{XY}}(x,y)$ is the joint distribution, $P_{\mathbf{X}}(x) = \sum_y P_{\mathbf{XY}}(x,y)$ and $P_{\mathbf{Y}}(y) = \sum_x P_{\mathbf{XY}}(x,y)$ are the marginals of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Mutual information quantifies the amount of information gained about one random variable through observing another random variable. It is a dimensionless quantity, typically measured in bits, and can be regarded as the reduction in uncertainty about one random variable given knowledge of the other. High mutual information signifies a substantial reduction in uncertainty, and *vice versa* [18].

Back to the content of calibrating the k-th layer's quantizing activation via CL, activations $\mathbf{A}_F^k$ and $\mathbf{A}_Q^k$ can be considered as two variables. The corresponding variables should share more information, *i.e.*, the mutual information of the activations $I(\mathbf{A}_F^k, \mathbf{A}_Q^k)$ should be maximized to enforce them mutually dependent. The core idea of CL is to discriminate whether a given pair of activation $(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$ is positive or negative, *i.e.*, inferring the distribution $P(D \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$, in which D is the variable decides whether $i = j$ or $i \neq j$. However, accessing

$P(D \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$ directly is not feasible [10], so we introduce its variational approximation instead

$$q(D \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k), \quad (11)$$

which can be estimated by our models $d(\cdot, \cdot)$. Intuitively, $q(D \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$ can be treated as a binary classifier, which can classify a given pair $(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$ into positive or negative. From Bayes' theorem, we can formalize the posterior probability of two activations coming from a positive pair as follows:

$$q(D=1 \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) = \frac{q(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k \mid D=1)\frac{1}{M+1}}{q(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k \mid D=1)\frac{1}{M+1} + q(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k \mid D=0)\frac{M}{M+1}}.$$

The probability of activations from negative pair is $q(D = 0 \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) = 1 - q(D = 1 \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$. To simplify the NCE derivative, several works [10, 32, 34, 37, 42] build assumptions about the dependence of the variables, we also use the assumption that the activations from positive pairs are dependent and the ones from negative pairs are independent, *i.e.* $q(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k \mid D = 1) = P(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$ and $q(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k \mid D = 0) = P(\mathbf{A}_{Q,i}^k)P(\mathbf{A}_{F,j}^k)$. Hence, the above equation can be simplified as:

$$q(D = 1 \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) = \frac{P(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)}{P(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) + P(\mathbf{A}_{Q,i}^k)P(\mathbf{A}_{F,j}^k)M}. \quad (12)$$

By applying the logarithm to Eq. 12 and rearranging the terms, we obtain the following equation:

$$\log q(D = 1 \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) \leq \log \frac{P(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)}{P(\mathbf{A}_{Q,i}^k)P(\mathbf{A}_{F,j}^k)} - \log(M). \quad (13)$$

By taking expectation on both sides with respect to $P(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k)$, and combining the definition of mutual information in Eq. 10, we can derive the form of mutual in-

Table 1. Comparison of `CL-Calib` with various post-training quantization algorithms on ImageNet.

Methods	Bits (W/A)	ResNet-18	ResNet-50	MobileNetV2	RegNetX-600MF	RegNetX-3.2GF	MNasx2
Full Prec.	32/32	71.01	76.63	72.62	73.52	78.46	76.52
ACIQ-Mix [2]		67.00	73.80	-	-	-	-
LAPQ [24]		60.30	70.00	49.70	57.71	55.89	65.32
Bit-Split [39]		67.00	73.80	-	-	-	-
AdaRound [21]	4/4	67.96	73.88	61.52	68.20	73.85	68.86
QDrop [40]		69.17	75.15	68.07	70.91	76.40	72.81
PD-Quant [20]		69.30	75.09	68.33	71.04	**76.57**	73.30
CL-Calib		**69.41** (+0.11)	**75.38** (+0.23)	**68.56** (+0.23)	**71.38** (+0.34)	76.40 (-0.17)	**73.60** (+0.30)
LAPQ [24]		0.18	0.14	0.13	0.17	0.12	0.18
Adaround [21]		0.11	0.12	0.15	-	-	-
QDrop [40]	2/4	64.57	70.09	53.37	63.18	71.96	63.23
PD-Quant [20]		65.07	**70.92**	55.27	64.00	72.43	63.33
CL-Calib		**65.14** (+0.07)	**70.92** (±0.00)	**55.63** (+0.36)	**64.50** (+0.50)	**72.82** (+0.39)	**63.46** (+0.13)
QDrop [40]		57.56	63.26	17.30	49.73	62.00	34.12
PD-Quant [20]	4/2	58.65	64.18	20.40	51.29	62.76	38.89
CL-Calib		**59.03** (+0.38)	**65.12** (+0.96)	**22.77** (+2.37)	**52.35** (+1.06)	**63.53** (+0.77)	**40.80** (+1.91)
BRECQ [19]		42.54	29.01	0.24	3.58	3.62	0.61
AdaQuant [15]		0.11	0.12	0.15	-	-	-
QDrop [40]	2/2	51.42	55.45	10.28	39.01	54.38	23.59
PD-Quant [20]		53.08	56.98	14.17	40.92	55.13	28.03
CL-Calib		**54.45** (+1.37)	**58.30** (+1.32)	**17.70** (+3.53)	**42.19** (+1.27)	**56.39** (+1.26)	**30.34** (+2.31)

formation as:

$$\overbrace{I(\mathbf{A}_Q^k, \mathbf{A}_F^k)}^{\text{targeted MI}} \geq \overbrace{\mathbb{E}_{P(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k | D=1)} \left[\log q(D=1 \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) \right]}^{\text{lower bound}} + \log(M),$$

(14)

where $I(\mathbf{A}_Q^k, \mathbf{A}_F^k)$ is the mutual information between the quantized and full-precision distributions of our targeted object. Instead of directly maximizing the mutual information, maximizing the lower bound in the Eq. 14 is a practical solution. We can further loosen the lower bound, and get the optimization objective corresponding to `CL-Calib` loss in Eq. 8 as follows:

$$I(\mathbf{A}_Q^k, \mathbf{A}_F^k) - \log(M)$$
$$\geq \mathbb{E}_{P(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k | D=1)} \left[\log q(D=1 \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) \right] + M \mathbb{E}_{P(\mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k | D=0)} \left[\log q(D=0 \mid \mathbf{A}_{Q,i}^k, \mathbf{A}_{F,j}^k) \right]$$
$$= \mathbb{E}_{\mathbf{A}_Q^k} \{ \overbrace{\mathbb{E}_{(\mathbf{A}_Q^k, \mathbf{A}_{F,+}^k)} \left[\log d(\mathbf{A}_Q^k, \mathbf{A}_{F,+}^k) \right] + \mathbb{E}_{\{\mathbf{A}_{F,-,i}^k\}_{i=1}^M} \left[-\log(1 - d(\mathbf{A}_Q^k, \mathbf{A}_{F,-,i}^k)) \right]}^{\text{negative CL-Calib loss in Eq. 8}} \}.$$

(15)

By minimizing the overall calibration objective $\mathcal{L}_{calib}$ in Eq. 9, we can obtain the function $d^\star$. This function serves as a variational approximation of the distribution $P(D = 1 \mid \mathbf{A}_Q^k, \mathbf{A}_F^k)$. In this way, we can optimize the lower bound of targeted mutual information $I(\mathbf{A}_Q^k, \mathbf{A}_F^k)$. This validates our assertion that employing `CL-Calib` to learn activation quantization parameters effectively maximizes the mutual information between pre- and post-quantized activations.

4. Experiments

In this section, we perform experiments on image classification with ImageNet dataset [6] to validate the effectiveness of `CL-Calib`. Furthermore, we designate a series of ablative and analytical studies to verify the effectiveness and investigate various properties of `CL-Calib` including the

regularization, and reliability. All experiments are implemented using PyTorch [26] on 8 Nvidia RTX A6000 and **codes** are in the supplementary.

4.1. Experiment Settings

To evaluate our proposed method, we quantize various Convolutional Neural Network (CNN) architectures, including ResNet [12], MobileNetV2 [29], RegNet [28], and Mnas-Net [36]. The full-precision (FP) pre-trained models used in our experiments are sourced from [19]. We evaluate `CL-Calib` on the ImageNet dataset, using a batch size of 128. For calibration, we randomly sample 128 images from the ImageNet training dataset. Unless specified otherwise, we set the first and last layer quantization to 8-bit for all PTQ experiments. Furthermore, we maintain the same quantization settings and hyper-parameters as used in the QDrop [40] and PD-quant [20] (previous state-of-the-art methods) implementations. The learning rate for the activation quantization scaling factor is set to 4e-5, and for weight quantization rounding, the learning rate is set to 3e-3. The choice of hyper-parameters τ and λ in Eq. 8 and Eq. 9 will be discussed in Sec. 4.4, respectively. The fine-tuning of quantization parameters is performed over 20,000 iterations.

4.2. Comparison to State-of-the-arts

In a thorough comparison of our method `CL-Calib` with numerous PTQ algorithms across different bit settings, we observe that `CL-Calib` consistently surpasses state-of-the-art PTQ methods, notably in extremely low-bit scenarios. Notably, merely optimizing the activation scaling factors does not provide satisfactory results at low bits;

Table 2. Serving as an add-on module on ImageNet with 2W2A setting.

Methods	ResNet-18	ResNet-50	MobileNetV2	RegNetX-600MF	RegNetX-3.2GF	MNasx2
Full Prec.	71.01	76.63	72.62	73.52	78.46	76.52
PD-Quant	53.08	56.98	14.17	40.92	55.13	28.03
CL-Calib	**54.45** (+1.37)	**58.30** (+1.32)	**17.70** (+3.53)	**42.19** (+1.27)	**56.39** (+1.26)	**30.34** (+2.31)
QDrop	51.42	55.45	10.28	39.01	54.38	23.59
CL-Calib	**53.63** (+3.21)	**56.78** (+1.33)	**15.48** (+4.20)	**42.00** (+0.99)	**56.65** (+2.27)	**28.89** (+5.30)

(a) Effect of λ in Eq. 9 with W2A2 MNasx2.

(b) Effect of λ in Eq. 9 with W2A2 MobileNetV2.

(c) Comparison to [40].
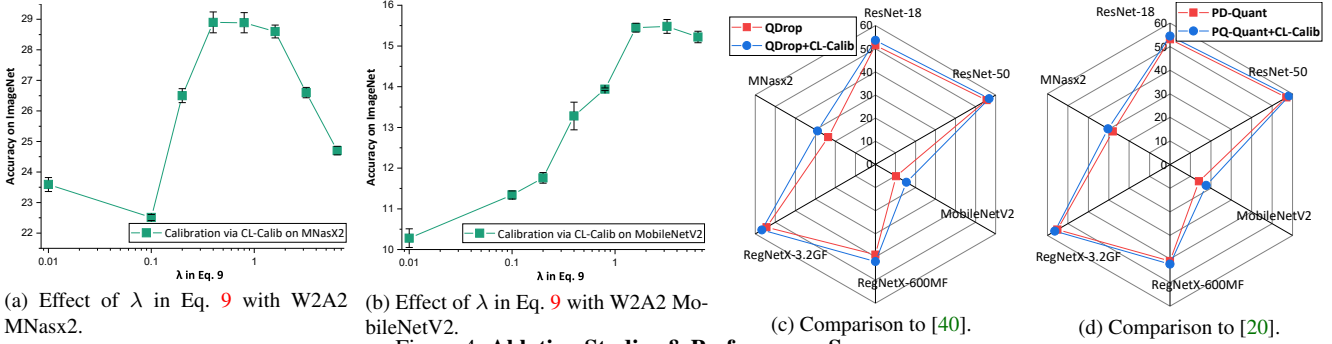
(d) Comparison to [20].

Figure 4. **Ablation Studies & Performance Summary.**

therefore, unless otherwise stated, all our subsequent experiments optimize both the rounding values and activation scaling factors, following the precedent set by previous SoTAs such as QDrop and PD-Quant [20, 40]. The outcomes are summarized in Tab. 1, demonstrating significant improvements achieved by CL-Calib compared to the previously SoTA PTQ baselines. For instance, when quantizing the network to W4A4, experiments indicate that CL-Calib slightly surpasses PD-Quant. However, as the bit-width decreases, the advantages of CL-Calib become increasingly apparent. At the most challenging W2A2 bit setting, CL-Calib exceeds the baseline across all network architectures, in particular, CL-Calib boosts the accuracy of post-quantized MobileNetV2 by 3.43% and MNasx2 by 2.31%.

4.3. Experiments on ViT and Object Detetction

There are more extensive experiments in the Appendix with different architectures, including the **Vision Transformer** [7] (Tab. 3 in Appendix), and on various tasks, such as **object detection** (Tab. 5 in Appendix). We can witness the generalization ability of CL-Calib from those experiments.

4.4. Further Analysis

Ablative Studies. We conduct a series of ablation studies of CL-Calib on the most challenging W2A2 settings with MobileNetV2 and MNasx2 architectures. By adjusting the coefficient λ in the loss function $\mathcal{L}_{calib}$ (Eq.9), we examine the influence of CL-Calib loss in calibration. The results, illustrated in Figs.4c and 4d, show a trend where increasing λ improves the performance, thus validating the effective-

ness of our proposed method. In all our experiments, we use the PQ-Quant [20] codebase. Setting $\lambda = 0$ is our baseline, which corresponds to PD-Quant.

Serving as an Add-on Module. We implement CL-Calib on the SoTA PTQ methods, such as QDrop [40], and PD-Quant [20] with extreme W2A2 PTQ setting. The results are presented in Fig. 4a,4b, and Tab. 2. We can witness that CL-Calib can improve the performances of all the methods. This is a piece of evidence that our method can serve as an add-on module.

Hyper-parameters and Relative Module Selection. In addition to ablation studies, we conduct experiments to select the important hyper-parameters and modules. We investigate two hyper-parameter, the co-efficient τ to adjust the temperature in the CL-Calib loss in Eq. 8, the number of negative samples M, the architecture of embedding network q in critic function, and the similarity metric for contrastive
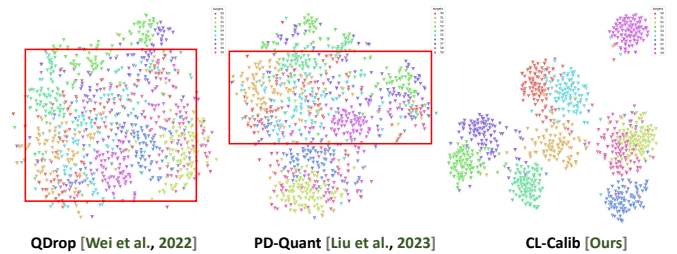

Figure 5. t-SNE visualization for activations calibrated by different calibration methods.

Representation Ability. We examined the representation capability of the quantized activations through t-SNE visualization [38]. The results, shown in Fig. 5, indicate that the
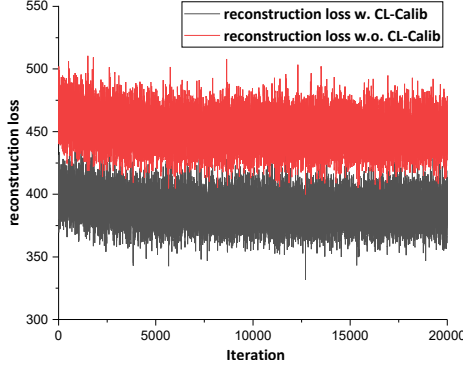
Figure 6. Training curves of quantization loss $\mathcal{L}_{quant}$.

activations calibrated by `CL-Calib` exhibit superior representation capacity.

Regularization: Mitigating Overfitting. We scrutinize the evolution of the quantization loss, $\mathcal{L}_{quant}$, as given in Eq.8 and Eq.9 during training. The term $\mathcal{L}_{quant}$ in Eq.9 can represent any quantization loss, which *is easily overfitted due to the limited calibration data*. By manipulating the coefficient λ in Eq.9, we can examine $\mathcal{L}_{quant}$ in isolation. The training curves are depicted in Fig.6. Upon enabling the `CL-Calib` loss $\mathcal{L}_{\texttt{CL-Calib}}$ (by setting $\lambda = 3.2$), we notice a reduction in $\mathcal{L}_{quant}$ compared to when our designed `CL-Calib` is not used (also see Tab. 7 in Appendix). This decrease is accompanied by an improvement in the corresponding test performance (see Tab. 2). Hence, we can deduce that `CL-Calib` module functions as a regularization term.

5. Related Work

Quantization stands as one of the most potent techniques for compressing neural networks. It can be broadly categorized into two primary methods: Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). QAT [8, 17] considers the quantization in the network training phase, while PTQ [23] quantizes the network after training. Due to its lower time consumption and computational resource requirements, PTQ is extensively employed in network deployment. Most of the work of PTQ involves learning the quantization parameters for weights and activations in each layer. In order to calculate the activations in the network, a small number of calibration samples should be used as input in PTQ. The selected quantization parameters are dependent with the selection of these calibration samples. [16, 19, 24] demonstrate the effect of the number of calibration samples. *Activation quantization is essentially a compression problem with a strong emphasis on maximizing the preservation of information contained in the activation.* Admittedly, heuristically designed distance metrics have achieved promising performance. However, we argue that previous works neglect the well-defined distributional metric in information theory, which is necessary for success

measurement.

Contrastive Learning and Mutual Information Maximization. Recently, contrastive learning is proven to be an effective approach to MI maximization, and many methods based on contrastive loss for self-supervised learning are proposed, such as Deep InfoMax [13], Contrastive Predictive Coding [25], MemoryBank [42], Augmented Multiscale DIM [1], MoCo [11] and SimSaim [4]. These methods are generally rooted in NCE [10] and InfoNCE [13] which can serve as optimizing the lower bound of mutual information [27]. In the meantime, [37] and [3] generalize the contrastive idea into the content of knowledge distillation (KD) to pull-and-push the representations of teacher and student. Intuitively, the core principle of contrastive learning revolves around drawing representations from positive pairs closer, while pushing representations from negative pairs further apart in a contrastive space. One of the main challenges in employing contrastive loss is defining these negative and positive pairs.

Our approach for PTQ calibration harnesses the core concept of existing contrastive learning methods, notably contrastive-based network compression methods such as CRD [37], WCoRD [3], and MIM-BNN [32]. However, our methodology differs from these methods in several key ways: **(i)** The targeted mutual information (MI) we focus on and the numerical problem we formulate are entirely distinct; **(ii)** Our approach can naturally circumvent the cost of the MemoryBank [42] associated with the exponential number of negative pairs in related works, thanks to the limited calibration data size in our task; **(iii)** The discriminator functions we use are distinct. By leveraging the inherent characteristics of layer-wise quantization, where pre-trained full-precision network is available, we design a discriminator network specifically for PTQ, as discussed in Sec.3.1.

6. Conclusion

We argue that previous PTQ calibration works neglect the well-defined distributional metric in information theory, which is necessary for success measurement. To address this, we introduce a contrastive learning (CL) framework for PTQ calibration, focusing on maximizing the mutual information between pre- and post-quantization activations. This approach, which utilizes a self-supervised proxy task for optimizing quantization parameters, effectively retains more information in the quantized activations, enhancing the performance of the quantized model. Importantly, our method avoids common collusion, thanks to an ingeniously constructed critic function, ensuring effectiveness even under conditions of extremely limited calibration data.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019. 4, 8

[2] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. Aciq: analytical clipping for integer quantization of neural networks. In *ICLR*, 2018. 6

[3] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein contrastive representation distillation. In *CVPR*, 2021. 8

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 4, 8

[5] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *ICCV Workshops*, 2019. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3, 6

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[8] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *ICLR*, 2020. 8

[9] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021. 1

[10] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010. 3, 5, 8

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 4, 8

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[13] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019. 4, 8

[14] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *NeurIPS*, 2016. 1

[15] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Improving post training neural quantization: Layer-wise calibration and integer programming. *arXiv preprint arXiv:2006.10518*, 2020. 6

[16] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Improving post training neural quantization: Layer-wise calibration and integer programming. *arXiv preprint arXiv:2006.10518*, 2020. 8

[17] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*, 2018. 8

[18] Solomon Kullback. *Information theory and statistics.* Courier Corporation, 1997. 1, 5

[19] 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. In *ICLR*, 2021. 1, 2, 3, 6, 8

[20] Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Pd-quant: Post-training quantization based on prediction difference metric. In *CVPR*, 2023. 3, 4, 6, 7

[21] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *ICML*, 2020. 6

[22] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *ICCV*, 2019.

[23] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *CoRR*, 2021. 1, 2, 8

[24] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M. Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, 2021. 6, 8

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3, 4, 8

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[27] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *ICML*, 2019. 8

[28] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020. 6

[29] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 6

[30] Yuzhang Shang, Bin Duan, Ziliang Zong, Liqiang Nie, and Yan Yan. Lipschitz continuity guided knowledge distillation. In *ICCV*, 2021. 1

[31] Yuzhang Shang, Dan Xu, Bin Duan, Ziliang Zong, Liqiang Nie, and Yan Yan. Lipschitz continuity retained binary neural network. In *European conference on computer vision*, pages 603–619. Springer, 2022.

[32] Yuzhang Shang, Dan Xu, Ziliang Zong, Liqiang Nie, and Yan Yan. Network binarization via contrastive learning. In *ECCV*, 2022. 5, 8

[33] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1972–1981, 2023. 1

[34] Yuzhang Shang, Zhihang Yuan, and Yan Yan. Mim4dd: Mutual information maximization for dataset distillation. *Advances in Neural Information Processing Systems*, 2023. 5

[35] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao,

Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. In *AAAI*, 2020. 3

[36] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019. 6

[37] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020. 5, 8

[38] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 7

[39] Peisong Wang, Qiang Chen, Xiangyu He, and Jian Cheng. Towards accurate post-training network quantization via bit-split and stitching. In *ICML*, 2020. 6

[40] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: randomly dropping quantization for extremely low-bit post-training quantization. In *ICLR*, 2022. 1, 3, 4, 6, 7

[41] Di Wu, Qi Tang, Yongle Zhao, Ming Zhang, Ying Fu, and Debing Zhang. Easyquant: Post-training quantization via scale optimization. *arXiv preprint arXiv:2006.16669*, 2020. 1

[42] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 3, 5, 8

[43] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. 1