AlignQ: Alignment Quantization with ADMM-based Correlation Preservation

Ting-An Chen[1], De-Nian Yang[2,3], Ming-Syan Chen[1,3]
[1]Graduate Institute of Electrical Engineering, National Taiwan University, Taiwan
[2]Institute of Information Science, Academia Sinica, Taiwan
[3]Research Center for Information Technology Innovation, Academia Sinica, Taiwan
tachen@arbor.ee.ntu.edu.tw, dnyang@iis.sinica.edu.tw, mschen@ntu.edu.tw

Abstract

Quantization is an efficient network compression approach to reduce the inference time. However, existing approaches ignored the distribution difference between training and testing data, thereby inducing a large quantization error in inference. To address this issue, we propose a new quantization scheme, Alignment Quantization with ADMM-based Correlation Preservation (AlignQ), which exploits the cumulative distribution function (CDF) to align the data to be i.i.d. (independently and identically distributed) for quantization error minimization. Afterward, our theoretical analysis indicates that the significant changes in data correlations after the quantization induce a large quantization error. Accordingly, we aim to preserve the relationship of data from the original space to the aligned quantization space for retaining the prediction information. We design an optimization process by leveraging the Alternating Direction Method of Multipliers (ADMM) optimization to minimize the differences in data correlations before and after the alignment and quantization. In experiments, we visualize non-i.i.d. in training and testing data in the benchmark. We further adopt domain shift data to compare AlignQ with the state-of-the-art. Experimental results show that AlignQ achieves significant performance improvements especially in low-bit models. Code is available at https://github.com/tinganchen/AlignQ.git.

1. Introduction

Convolutional neural networks (CNNs) have been demonstrated as effective models in computer vision tasks, such as image segmentation [2, 35] and object detection [16, 30, 34]. However, CNNs are suffered from large computation costs and memory storage when deployed on the resource-limited mobile devices [7]. Therefore, various model acceleration methods are proposed, including *pruning* [20, 22, 27, 29], *quantization* [6, 42, 43] and *structure simplification* [10, 45]. Quantization has recently received increasing attentions due to the effectiveness of acceleration on inference by reducing the bit widths of model weights and activations.

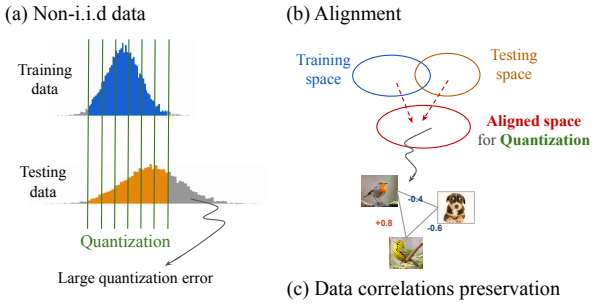


Figure 1. Motivation of *AlignQ*. Figure (a) presents an example of quantization on *non-i.i.d.* training and testing data. A quantization range learned from training data may induce a large quantization error when applied in the testing data with different distributions. Figure (b) and Figure (c) illustrate our motivation to address the issue in Figure (a). In Figure (b), we propose to align the data to the same space for quantization to minimize the quantization error. In addition, as shown in Fig (c), we observe that the significant changes in data correlations induce a large quantization error. Accordingly, we aim to preserve the data correlations after the alignment and quantization to retain the prediction information in the original space for further reducing the quantization error.

In existing quantization research, *quantization-aware training (QAT)* learned quantization parameters, including clipping ranges and scale parameters, from the training data and applied them to the testing data [3, 8, 11, 28, 47, 48]. In contrast, *zero-shot quantization (ZSQ)* adopted the concepts of knowledge distillation and employed the batch normalization means and variances from the full-precision model, to learn a quantized model that can generate similar features to reduce the quantization error [5, 9, 19, 31, 44]. The learned batch statistics are also utilized in testing. However, the previous approaches ignored the difference between the training and testing data. As shown in Fig. 1 (a), the real-world image data are usually collected under inconsistent quali-

ties, such as different colors, brightness, and rotations, leading to *non-i.i.d. (independently and identically distributed)* data [21]. Accordingly, it may induce a large quantization error when using the trained parameters in testing.

To address this issue, we propose *AlignQ* to align the data into the same domain for quantization to minimize the quantization error (illustrated in Fig. 1 (b)). In this paper, our idea is to exploit the cumulative distribution function (CDF) as the alignment function since the CDF of an arbitrary continuous distribution follows the uniform distribution [24] (demonstrated in Sec. 3.1). The uniform space is appropriate for uniform quantization that is hardware-friendly with a few simple operations [3, 8, 28]. In addition, CDF retains the data order, i.e., larger values still exceed small values after the transformation.

Furthermore, our theoretical analysis indicates that notable changes in data correlations after quantization induce a larger quantization error. Therefore, as shown in Fig. 1 (c), we aim to preserve the data correlations after the alignment-quantization process. We leverage the Alternating Direction Method of Multipliers (ADMM) optimization to minimize the differences to reduce the quantization error. To achieve the two goals in this paper, to minimize 1) the prediction loss of the quantized models and 2) the differences of data correlations before and after the alignment-quantization process, ADMM addresses this multi-goal optimization problem by dividing it into sub-problems and solving them [4].

To verify that the proposed AlignQ can reduce the quantization error derived from the *non-i.i.d.* in training and testing data, we compare with the state-of-the-art not only on the benchmark datasets, including CIFAR-10 [25], SVHN [33], ImageNet [37], but also on domain shift benchmarks, including digits [12, 13, 26, 33] and Office-31 [38].

The contributions are summarized as follows:

1. We make the first attempt to design a new quantization scheme, *AlignQ*, that aligns the *non-i.i.d.* data to be *i.i.d.* to minimize the quantization error.

2. We prove that the changes in data correlation after quantization induce a large quantization error and thereby leverage the ADMM optimization procedure to minimize the differences of the data correlations before and after quantization to reduce the error.

3. We compare AlignQ with the state-of-the-art on benchmarks and the domain shift datasets. Experimental results show that AlignQ achieves significant performance improvements, especially at low bit widths.

2. Related works

Quantization-aware training (QAT). QAT is designed to learn clipping ranges or scale parameters for quantization during the training process [3, 11, 28, 46–48]. The trained quantization statistics are then applied to quantize the inference data. DoReFa [48] and LSQ [11] proposed an efficient low-bit forward and backward procedure to estimate the non-differentiable gradients. LLSQ [47] learned the shift and scale factors on batch normalization layers to adjust the quantization level for reducing the quantization error dynamically. ACIQ [3], OCS [46] and APoT [28] learned a clipping function to determine the quantization space. However, the existing QAT approaches ignore the discrepancy between training and testing data. The difference may lead to a larger quantization error and performance degradation.

Zero-shot quantization (ZSQ). Recent research proposes zero-shot quantization (ZSQ) to incorporate the knowledge distillation [40] into quantization [5, 9, 19, 31, 44, 46]. They utilize the knowledge derived from the full-precision model to enhance the performance of the low-bit model. OCS [46] used the KL-divergence to minimize the clipping bounds between the full-precision and the quantized model. In contrast, GDFQ [44], GZNQ [19], ZeroQ [5], Choi *et al.*'s work [9] and ZAQ [31] used different knowledge distillation losses to minimize the prediction results of the full-precision and the quantized model. In particular, GZNQ [19] also distilled the knowledge from other models in different compression approaches, including pruning and low-rank models. ZeroQ [5] further learned the batch statistics (means and variances) close to the full-precision model. In addition, ZAQ [31] focused on learning the features (from the low-bit model) similar to the floating-point model by examining the inter-channel discrepancy. Since ZSQ employs the auxiliary information from the pretrained full-precision model, they obtain larger memory and computation costs in the training process. In addition, ZSQ relies on the prediction results from the training data. Therefore, ZSQ is more sensitive in the discrepancy of the distributions in training and testing data which induces a larger quantization error as illustrated in Fig. 1. Accordingly, in this paper, we proposed to address the issue, *non-i.i.d. (independently and identically distributed)* in training and testing data to minimize the quantization error.

3. AlignQ

In this section, we introduce AlignQ as shown in Fig. 2. Sec. 3.1 presents the *CDF alignment quantization* that individually aligns the batches of the training and testing data to the same domain to minimize the quantization error. We also design a novel approach to update the weights of the quantized model. In Sec 3.2, we focus on preserving the data correlations during the alignment-quantization process. We prove that significant changes in the correlations after quantization induce a large quantization error. Therefore, we propose an optimization process that leverages the Alternating Direction Method of Multipliers (ADMM) [4]

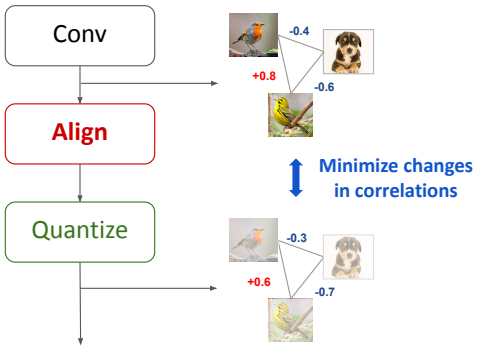


Figure 2. Overview of *AlignQ*. AlignQ is a quantization scheme that considers 1) non-i.i.d. in training and testing data and 2) the changes in data correlations during quantization to minimize the quantization error. AlignQ first aligns the training and testing batch data into the same uniform space (introduced in Sec. 3.1). Afterward, the aligned data is uniformly quantized. In addition, to preserve the data correlations during the alignment-quantization process and minimize the performance degradation, we utilize the Alternating Direction Method of Multipliers (ADMM) to minimize the differences of the data correlations before and after the quantization (detailed in Sec. 3.2).

to minimize the differences of the data correlations before and after the quantization.

3.1. CDF alignment quantization

To diminish the quantization error on non-i.i.d. (independently and identically distributed) data as shown in Fig. 1, we target transforming both training and testing data to be i.i.d., i.e., aligning the data to the same domain, before quantization (see Fig. 2).

3.1.1 CDF alignment

We propose a novel data alignment approach by leveraging *cumulative distribution function (CDF)* [24]. In Theorem 3.1, we demonstrate that the CDF of an arbitrary continuous distribution follows the uniform distribution.

Theorem 3.1. *(*Proved in Appendix A.1*) Let X have the cumulative distribution function (CDF) of the continuous type that is strictly increasing on the support $a \leq x \leq b$. Then the function $Y = F(X)$ has a distribution Uniform(0, 1).*

According to Theorem 3.1, we align the training and testing data into the same uniform space by individual CDFs. In addition, the CDF transformation will not change the order of data, i.e., large values after CDF transformation are still larger than the small values. Correspondingly, the information and property of data after the alignment can still be retained.

The following challenge is which CDF should be adopted for the alignment. Since previous research demonstrates that the CNN weights and activations converge in normal distribution [17, 28, 32, 48], which is also experimentally validated on the benchmark datasets in this paper (see Appendix D), we then adopt the CDF of the normal distribution as the alignment function:

$$F(x) = \Phi(x; \mu, \sigma) = \frac{1}{2}[1 + erf(\frac{x-\mu}{\sigma\sqrt{2}})],$$

where

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt. \tag{1}$$

In Eq. (1), x is the CNN weight or activation (feature) value, and μ and σ are the mean and standard deviation of the normal distribution. For weight quantization, we utilize the batch data's mean and standard deviation to estimate μ and σ in each training iteration. On the other hand, we use the CDF of the standard normal distribution instead for activation quantization, i.e., $\mu = 0$ and $\sigma = 1$. However, the aligned space is $Uniform(0, 1)$, but the wights and activations are not always positive values. Thus, we scale and shift $F(x)$ to $Uniform(-\alpha, \alpha)$ by $(2 \cdot F(x) - 1) \cdot \alpha$.

3.1.2 Uniform quantization

After the alignment, data follows $Uniform(-\alpha, \alpha)$ (i.i.d). As shown in Fig. 2, we can then apply the uniform quantization [15]:

$$Q(z) = \frac{round(2^{k-1} \cdot z)}{2^{k-1}}, \tag{2}$$

where z is the value after the shifted and scaled CDF alignment, i.e., $z = (2 \cdot F(x) - 1) \cdot \alpha$, $round$ denotes the rounding operation, and k is the bit width.

3.1.3 Gradient approximation for the update of the quantized weights

Since the quantized values are discrete, i.e., the quantization function in Eq. (2) is non-differentiable, it is challenging to access the gradients of weights for an update. Therefore, we propose a gradient approximation approach to address this issue. According to Eq. (1) and Eq. (2), we derive the quantized weight w_q from $w_q = Q((2 \cdot F(w) - 1) \cdot \alpha)$, where w is the original floating-point weight. Let the probability distribution function (pdf) of w be $f(w)$ which is the normal distribution with batch mean and standard deviation (see Sec. 3.1.3). In addition, assume $\mathcal{L}$ is the training loss of the quantized model. Therefore, the gradient of w is obtained by the chain rule in Calculus:

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial w_q} \cdot \frac{\partial w_q}{\partial w} = \frac{\partial \mathcal{L}}{\partial w_q} \cdot \frac{\partial Q}{\partial w} \cdot (2\alpha \cdot f(w)). \tag{3}$$

Eq. (3) shows that the gradient of w can be formulated as the multiplication of three terms. The first term is the gradient of w_q, $\frac{\partial \mathcal{L}}{\partial w_q}$, which can be directly accessed from the back propagation. The following equation $\frac{\partial w_q}{\partial w} = \frac{\partial Q}{\partial w} \cdot (2\alpha \cdot f(w))$ is 2α is derived from $w_q = Q((2 \cdot F(w) - 1) \cdot \alpha)$. However, the term $\frac{\partial Q}{\partial w}$ cannot be directly derived from its first-order derivative since Q is non-differentiable. Nevertheless, since $Sigmoid$ function is distributed as "the stairs" of the step function Q, we then use the gradients of the continuous sigmoid function $\frac{\partial \tilde{Q}}{\partial w}$ to estimate $\frac{\partial Q}{\partial w}$:

$$\frac{\partial Q}{\partial w} \simeq \frac{\partial \tilde{Q}}{\partial w} = s(t(w_q)) \cdot (1 - s(t(w_q))), \tag{4}$$

where $s(\cdot) = \frac{1}{1+e^{-x}}$ is the sigmoid function, and its first-order derivative is $s(x) \cdot (1 - s(x))$. Notice that before calculating the sigmoid gradient, we first transform the quantized weight w_q by the transformation function t with $t(w_q) = \{2 \cdot [(w_q + 0.5) \cdot (2^k - 1)] \ mod \ 1\}$ to shift and scale the quantized space to the space of the sigmoid function. k (also appears in Eq. (2)) that represents the bit width, and mod denotes the operation of taking the remainders. As a consequence, we adopt the approximate gradient $\frac{\partial \mathcal{L}}{\partial w_q} \cdot \frac{\partial \tilde{Q}}{\partial w} \cdot (2\alpha \cdot f(w))$ for the update of the model weights.

3.2. Data correlation preservation by Alternating Direction Method of Multipliers (ADMM)

3.2.1 Quantization error induced by the changes in data correlations

During the alignment-quantization process, we aim to preserve the data correlations as illustrated in Fig. 2 due to the following observation.

Proposition 1. The significant changes in the data correlations after quantization induces a larger quantization error (proved in Theorem 3.2).

Theorem 3.2. (Proved in Appendix A.2) *Let $\boldsymbol{X}_i \in \mathbb{R}^d$ be the CNN representation of the i-th of n input image data. The function Q quantizes the values to the discete Uniform$(-\alpha, \alpha)$, $\alpha \geq 0$. The quantized representation is denoted as $Q(\boldsymbol{X}_i)$, and the total quantization error of n data is $\sum_{i=1}^{n} ||\boldsymbol{X}_i - Q(\boldsymbol{X}_i)||_1$, where $||\cdot||_1$ represents the $l1$-norm. Now let the individual quantization error $\delta_i = \boldsymbol{X}_i - Q(\boldsymbol{X}_i)$, $\forall i = 1, 2, ..., n$, and the tolerated quantization error as ϵ. Then $P(\sum_{i=1}^{n} ||\boldsymbol{X}_i - Q(\boldsymbol{X}_i)||_1 < \epsilon) \geq 1 - \frac{n}{\epsilon^2}\mathbb{E}[||\delta_i||_1^2] - \frac{4\alpha}{\epsilon^2} \sum_{i,j=1;\ i<j}^{n} \mathbb{E}(||\delta_i||_1 + ||\delta_j||_1) - \frac{2}{\epsilon^2} \sum_{i,j=1;\ i<j}^{n} \mathbb{E}(|\boldsymbol{X}_i^T \boldsymbol{X}_j - Q(\boldsymbol{X}_i)^T Q(\boldsymbol{X}_j)|)$.*

Theorem 3.2 demonstrates that the total quantization error $\sum_{i=1}^{n} ||\mathbf{X}_i - Q(\mathbf{X}_i)||_1$ is not only relevant to the individual quantization error $||\delta_i||_1$ from each data i, but also tightly correlated to the discrepancy of the data correlations before and after the quantization $|\mathbf{X}_i^T \mathbf{X}_j - Q(\mathbf{X}_i)^T Q(\mathbf{X}_j)|, \forall i < j$. We first set a tolerated error ϵ. The probability that the total quantization error is smaller than ϵ will be large, i.e., the quantization error is limited in an acceptable (small) range in a high probability, if the changes in the data correlations after quantization are minor.

3.2.2 Minimization the changes in data correlations

According to Theorem 3.2, a small quantization error will be obtained when the discrepancy of the data correlations before and after the quantization, $\mathbb{E}(|\mathbf{X}_i^T \mathbf{X}_j - Q(\mathbf{X}_i)^T Q(\mathbf{X}_j)|)$, $\forall i < j = 1, 2, ..., n$, is minimized. We first define the discrepancy of the data correlations before and after the quantization $\mathbf{D} \in \mathbb{R}^{n \times n}$ as follows:

$$\mathbf{D} := \mathbf{X}^T \mathbf{X} - Q(\mathbf{X})^T Q(\mathbf{X}), \tag{5}$$

where $\mathbf{X} \in \mathbb{R}^{d \times n}$ represents the n d-dimensional image representations obtained after the CDF alignment (introduced in Sec. 3.1), and $Q(\mathbf{X}) \in \mathbb{R}^{d \times n}$ stands for the quantized representations (detailed in Eq. (2)). In order to minimize the discrepancy $\mathbf{D}$ to avoid a large quantization error, we target on the following objective function:

$$\min_{\mathbf{W}} \mathcal{L}_Q(\mathbf{W}) + \mu ||\mathbf{D}||_1, \tag{6}$$

where $\mathcal{L}_Q$ represents the loss function of the quantized network under the CDF quantization (detailed in Sec. 3.1), $\mathbf{W}$ denotes the set of the network weights, and $\mathbf{D}$ defined in Eq. (5) is the discrepancy of the data correlations which is minimized by the l_1- norm regularization (denoted as $||\cdot||_1$) with the penalty $\mu > 0$.

In Eq. (6), our goals are two-fold: 1) to minimize the prediction loss of the quantized model and 2) to minimize the changes in data correlations. Since Alternating Direction Method of Multipliers (ADMM) optimization has been demonstrated outperforming SGD to solve a multi-goal problem [4], we leverage ADMM to divide a complicated problem into sub-problems and effectively solve them. The ADMM constrained objective function is as follows:

$$\begin{aligned} &\min_{\mathbf{W}, \widetilde{\mathbf{D}}} \mathcal{L}_Q(\mathbf{W}) + \mu ||\widetilde{\mathbf{D}}||_1, \\ &s.t.\ \mathbf{D} - \widetilde{\mathbf{D}} = \mathbf{0}, \end{aligned} \tag{7}$$

which is equivalent to the objective function in Eq. (6). $\widetilde{\mathbf{D}}$ is a set of parameters as a proxy of the target $\mathbf{D}$ to be regularized, and the difference between proxy and target is minimized in the constraint.

ADMM solves the constrained objective Eq. (7) by formulating it as the augmented Lagrangian function:

Algorithm 1: The quantization and optimization process of AlignQ

Input: Training data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, model with initial weights $\mathbf{W}^{\{0\}}$, (k_1, k_2)-bitwidth setting for (weight, activation) quantization, and the parameters ($\mathbf{\Gamma}$, μ, ρ) for optimization.

Output: Quantized model weights $\mathbf{W}^*$.

1 **for** *k=1 to s steps* **do**

`/* CDF alignment quantization */`

2 Forward batch data through domain alignment quantization process (Eq. (1) and Eq. (2)).

3 Approximate the gradients by Eq. (4).

`/* ADMM correlation preservation */`

4 Retrieve the representations of intermediate layers.

5 Compute the discrepancy of the data correlations $\mathbf{D}^{\{k\}}$ defined as Eq. (5).

6 Update $\mathbf{W}^{\{k\}}$ by minimizing Eq. (9) with SGD.

7 Update $\mathbf{D}^{\{k\}}$ with Eq. (11).

8 Update $\mathbf{\Gamma}^{\{k\}}$ with Eq. (12).

return: $\mathbf{W}^{\{s\}}$

$$\begin{aligned}\Psi(\mathbf{W}, \widetilde{\mathbf{D}}, \mathbf{\Gamma}) = \mathcal{L}_Q(\mathbf{W}) + \mu||\widetilde{\mathbf{D}}||_1 \\ + \sum_l trace(\mathbf{\Gamma}_l^T(\widetilde{\mathbf{D}}_l - \mathbf{D}_l)) \\ + \frac{\rho}{2}\sum_l ||\widetilde{\mathbf{D}}_l - \mathbf{D}_l||_F^2,\end{aligned} \tag{8}$$

where $\mathbf{D}_l$ is the discrepancy of the correlations for the data features (extracted before and after the alignment-quantization process as illustrated in Fig. 2) obtained from the l-th network layer[1], and $\mathbf{\Gamma}_l$ denotes the dual variable, i.e., the Lagrange multiplier, which works as an attention mechanism that locally regularizes the change in correlation of each pair of data to different extents. The last term represents the global regularization on the primal residual in Frobenius norm (denoted as $||\cdot||_F$) with the penalty $\rho > 0$.

To efficiently obtain the optimal solution $(\mathbf{W}^*, \widetilde{\mathbf{D}}^*, \mathbf{\Gamma}^*)$, ADMM algorithm solves the decoupled sub-problems from Eq. (8):

1. Optimization for the CDF alignment quantization:

$$\mathbf{W}^{\{k+1\}} = arg\min_{\mathbf{W}} \Psi(\mathbf{W}, \widetilde{\mathbf{D}}^{\{k\}}, \mathbf{\Gamma}^{\{k\}}). \tag{9}$$

The weights are updated with the *stochastic gradient descent (SGD)* method [36, 41], where the gradients are approximated via Eq. (4).

[1] $\mathbf{D}$ is the concatenation of $\mathbf{D}_l$, and $\widetilde{\mathbf{D}}$ is the concatenation of $\widetilde{\mathbf{D}}_l$, $\forall\, l$.

2. Optimization for data correlation preservation:

$$\widetilde{\mathbf{D}}^{\{k+1\}} = arg.\min_{\widetilde{\mathbf{D}}} \mu||\widetilde{\mathbf{D}}||_1 + \frac{\rho}{2}\sum_l ||\widetilde{\mathbf{D}}_l - \mathbf{V}_l^{\{k\}}||_F^2, \tag{10}$$

where $\mathbf{V}_l^{\{k\}} = \mathbf{D}_l + \frac{1}{\rho}\mathbf{\Gamma}_l^{\{k\}}$. The solution is determined by the thresholding operation,

$$\widetilde{\mathbf{D}}_l^{\{k+1\}} = \begin{cases} (1 - \frac{\mu}{\rho||\mathbf{V}_l^{\{k\}}||_F})\mathbf{V}_l, & \text{if } ||\mathbf{V}_l^{\{k\}}||_F > \frac{\mu}{\rho}, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

3. Update of the dual variable:

$$\mathbf{\Gamma}_l^{\{k+1\}} = \mathbf{\Gamma}_l^{\{k\}} + \rho(\mathbf{D}_l^{\{k+1\}} - \widetilde{\mathbf{D}}_l^{\{k+1\}}). \tag{12}$$

The proposed quantization and optimization processes of AlignQ are summarized in Algorithm 1.

3.2.3 Convergence analysis

In this subsection, we analyze the convergence of ADMM optimization. As illustrated Line 5 to Line 8 in Algorithm 1, we compute the data correlations and update the ADMM parameters with the model weights in each training iteration. In other words, we update the parameters once in each iteration for an efficient training process. To ensure the convergence, we examine the decremental of training loss on data correlation preservation, i.e., the second term of Eq. (8), during the quantization process (see Appendix E).

4. Experiments

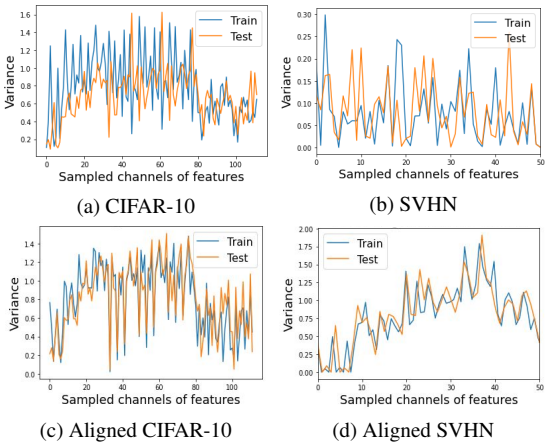
4.1. Experiment settings

Benchmark datasets. We evaluate AlignQ on benchmark datasets: CIFAR-10, SVHN [33] and ImageNet ILSVRC 2012 [37]. CIFAR-10 consists of 60K images for 10 classes. SVHN contains 600K images for 10 classes. ImageNet has more than 1.2M images with 1000 classes.

Domain shift datasets. In addition to the benchmark image classification datasets, we also evaluate AlignQ on domain shift datasets, including Office-31 [38] which contains three domains of data (each for 31 classes) and digit datasets including four domains (MNIST [26], MINIST-M [13], SynDigits [12] and SVHN [33]). The domain shift data is a non-i.i.d. scenario, where the training and testing data are derived from different domains.

Architectures. We evaluate AlignQ for the models ResNets [18], DensNet-40 [23] and MobileNet-V2 [39] on the benchmark datasets. In addition, we also implement the quantized DANN [14] and DSAN [49], the benchmark models in the domain shift tasks.

Training. We implement AlignQ with PyTorch [1] on an NVIDIA Tesla V100 GPU and an NVIDIA GTX 2080Ti.



(a) CIFAR-10 (b) SVHN

(c) Aligned CIFAR-10 (d) Aligned SVHN

Figure 3. Variances of features from training and testing data. Figure (a) presents the variances of CIFAR-10 features extracted from ResNet-20. Figure (b) shows the variances of SVHN features extracted from MobileNet-v2. Figure (c) and Figure (d) show the data variances after CDF alignment corresponding to Figure (a) and Figure (b).

Hyperparameters. The batch size is 512 for ImageNet, 128 for CIFAR-10 and SVHN, 100 for Office-31, and 64 for digits datasets during the training process. Training epochs are 200. The learning rate is in [0.01, 0.1]. The quantized space is $U(-\alpha, \alpha)$, where α is set to 1. The penalties μ and ρ in the ADMM optimization are in [0, 0.3].

4.2. Non-i.i.d. in benchmark datasets and the effectiveness of CDF alignment

Fig. 3 compare the distributions of training and testing features of ResNet-20 (on CIFAR-10) and MobileNet-v2 (on SVHN) before and after the CDF alignment (introduced in Sec. 3.1.1). Fig. 3 (a)-(b) present the variances of the original data features. Therefore, we verify the distributions of training and testing data are non-i.i.d. in the benchmark datasets accordingly. In comparison, Fig. 3 (c)-(d) visualize the variances after adopting the proposed CDF alignment. The results show the discrepancy from the non-i.i.d training and testing data distributions is notably reduced, validating the effectiveness of the proposed CDF alignment.

4.3. Comparison results

In the following, we compare the quantization results of AlignQ with QAT [11, 28, 47, 48] and ZSQ [5, 9, 19, 31, 44][2].

4.3.1 Comparisons on benchmarks

CIFAR-10. Table 1 compares AlignQ with the state-of-the-art under small (ResNet-20) and large architectures (ResNet-56 and DenseNet-40). Compared with QAT [11, 28, 47], AlignQ achieves 5% to 10% accuracy increment at 4-bit quantization and 10% to 30% improvement at 2-bit quantization. ZSQ [5, 9, 19, 31, 44] takes advantage of the training statistics from the full-precision model to improve the accuracy compared to QAT. In contrast, AlignQ without the knowledge distilled from the full-precision model also outperforms the baselines. Especially for 2-bit models, AlignQ can obtain 1% to 3% accuracy improvements since it addresses the issue of non-i.i.d. in training and testing data by aligning the data to the same space for quantization and then preserving the data correlations to effectively minimize the quantization error (detailed in Sec. 3.1 and 3.2).

SVHN. Table 2 indicates that ResNet architecture quantized

[2]In this paper, AlignQ and the compared works are quantized to W/A bits for each convolution layer.

Table 1. Quantization results on CIFAR-10. "W/A bit" means quantization bitwidth for weights and activations.

Model	Method	W/A bit	Acc (%)	W/A bit	Acc (%)
ResNet-20	LLSQ [47]	2 / 2	76.9	4 / 4	81.5
	LSQ [11]	2 / 2	77.7	4 / 4	83.4
	APoT [28]	2 / 2	65.2	4 / 4	81.0
	OCS [46]	2 / 2	-	4 / 4	89.1
	GZNQ [19]	2 / 2	-	4 / 4	89.1
	GDFQ [44]	2 / 2	-	4 / 4	90.3
	ZeroQ [5]	2 / 2	87.9	4 / 4	91.8
	Choi *et al.* [9]	2 / 2	88.1	4 / 4	91.9
	ZAQ [31]	2 / 2	88.9	4 / 4	92.1
	AlignQ (Ours)	2 / 2	**91.2**	4 / 4	**92.8**
ResNet-56	LSQ [11]	2 / 2	79.6	4 / 4	85.5
	APoT [28]	2 / 2	68.3	4 / 4	84.8
	ZeroQ [5]	2 / 2	88.1	4 / 4	92.5
	Choi *et al.* [9]	2 / 2	88.7	4 / 4	92.7
	ZAQ [31]	2 / 2	89.2	4 / 4	92.9
	AlignQ (Ours)	2 / 2	**91.7**	4 / 4	**93.2**
DenseNet-40	LLSQ [47]	2 / 2	81.5	4 / 4	87.2
	LSQ [11]	2 / 2	79.5	4 / 4	85.6
	APoT [28]	2 / 2	59.4	4 / 4	85.6
	ZeroQ [5]	2 / 2	91.3	4 / 4	92.6
	Choi *et al.* [9]	2 / 2	91.5	4 / 4	92.5
	ZAQ [31]	2 / 2	91.4	4 / 4	92.7
	AlignQ (Ours)	2 / 2	**93.0**	4 / 4	**93.1**

Table 2. Quantization results on SVHN. * indicates the quantization approach fails in the classification task.

Model	Method	W/A bit	Acc (%)	W/A bit	Acc (%)
ResNet-20	LLSQ [47]	2 / 2	93.0	4 / 4	93.4
	LSQ [11]	2 / 2	87.5	4 / 4	91.7
	APoT [28]	2 / 2	59.6	4 / 4	86.1
	ZeroQ [5]	2 / 2	94.3	4 / 4	95.6
	Choi *et al.* [9]	2 / 2	94.7	4 / 4	95.6
	ZAQ [31]	2 / 2	94.9	4 / 4	95.2
	AlignQ (Ours)	2 / 2	**95.5**	4 / 4	**95.6**
MobileNet-v2	DoReFa [48]	2 / 2	*	4 / 4	20.2
	LLSQ [47]	2 / 2	*	4 / 4	62.5
	LSQ [11]	2 / 2	*	4 / 4	77.9
	APoT [28]	2 / 2	*	4 / 4	*
	ZeroQ [5]	2 / 2	*	4 / 4	95.8
	Choi *et al.* [9]	2 / 2	*	4 / 4	96.2
	ZAQ [31]	2 / 2	83.6	4 / 4	96.3
	AlignQ (Ours)	2 / 2	**95.7**	4 / 4	**96.3**

Table 3. Quantization results on ImageNet.

Model	Method	W/A bit	Acc (%)	W/A bit	Acc (%)
ResNet-50	DoReFa [48]	2 / 2	-	4 / 4	33.2
	ACIQ [3]	2 / 2	-	4 / 4	59.3
	APoT [28]	2 / 2	-	4 / 4	58.2
	OCS [46]	2 / 2	-	4 / 4	66.2
	GDFQ [44]	2 / 2	65.0	4 / 4	68.7
	Choi *et al.* [9]	2 / 2	63.0	4 / 4	69.1
	ZeroQ [5]	2 / 2	63.1	4 / 4	69.3
	ZAQ [31]	2 / 2	65.5	4 / 4	70.1
	AlignQ (Ours)	2 / 2	**66.1**	4 / 4	**72.7**
ResNet-18	APoT [28]	2 / 2	-	4 / 4	44.3
	ZeroQ [5]	2 / 2	-	4 / 4	26.0
	GDFQ [44]	2 / 2	-	4 / 4	60.6
	GZNQ [19]	2 / 2	-	4 / 4	64.5
	AlignQ (Ours)	2 / 2	**61.1**	4 / 4	**65.7**

by AlignQ also outperforms the state-of-the-art on SVHN. Furthermore, we evaluate AlignQ on an efficient model MobileNet-v2 [39] with lightweight architectures and fewer model parameters. It is thereby more challenging to retain the prediction accuracy during the quantization process. Table 2 shows that most previous approaches fail in quantizing such a lightweight model at low bitwidths, e.g., 2 bits. ZAQ achieves 83.6% accuracy since it considers the inter-channel discrepancy in the quantized model and the full-precision model to enhance the prediction performance. However, the accuracy degradation is significant since ZAQ mainly focuses on distilling knowledge from the pretrained full-precision model but ignores the difference between the training and testing data. AlignQ addresses this issue by aligning data into the same domain and preserving the data (see Sec. 3.1 and Sec. 3.2). Accordingly, AlignQ achieves 95.7% accuracy for 2-bit MobileNet-v2, outperforming the state-of-the-art.

ImageNet. We further evaluate AlignQ and the state-of-the-art on a large-scale dataset, ImageNet. Table 3 shows the quantization results on ImageNet under ResNet-50 and ResNet-18 architectures. ResNet-50 quantized by AlignQ achieves 72.7% accuracy at 4-bit quantization, superior to ZAQ with 70.1%. In addition, ResNet-18 quantized by AlignQ also obtains a higher prediction accuracy than GZNQ. ZAQ utilizes the prediction results of the full-precision model to enhance the performance. GZNQ further employs the results of other lightweight models compressed by pruning and low-rank approaches. However, AlignQ, without knowledge distilled from the full-precision model, can still outperform them. It indicates that despite a minimal discrepancy of the generated features between the quantized model and other teacher models, the quantization error due to the non-i.i.d. training and testing data is not reduced. AlignQ with the idea of the data space alignment for quantization can effectively reduce such errors and obtain performance improvements.

Table 4. Accuracy (%) of quantized DANN (VGG-2) [14] on digits datasets. The header A → B for an example indicates training on the source A dataset and testing on the target B dataset. * indicates the quantization approach fails in the classification task.

W/A bit	Method	MNIST → MNIST-M	MNIST → SVHN	SynDigits → MNIST
32/32	Source only	58.8	30.4	50.6
	DANN [14]	91.3	30.6	58.0
	AlignQ (Ours)	**95.3**	**36.1**	**59.1**
2/2	DoReFa [48]	83.5	36.5	55.4
	LSQ [11]	52.7	24.1	54.5
	LLSQ [47]	57.1	31.1	50.6
	APoT [28]	*	*	*
	Choi *et al.* [9]	*	54.3	48.4
	ZeroQ [5]	*	56.2	47.2
	ZAQ [31]	*	56.5	48.8
	AlignQ (Ours)	**95.5**	**59.5**	**58.2**
3/3	DoReFa [48]	88.5	39.1	55.8
	LSQ [11]	54.6	24.1	53.8
	LLSQ [47]	80.9	38.2	56.8
	APoT [28]	85.2	29.0	*
	Choi *et al.* [9]	77.5	57.4	46.9
	ZeroQ [5]	76.9	57.4	47.4
	ZAQ [31]	66.8	58.1	48.1
	AlignQ (Ours)	**95.8**	**59.5**	**59.0**
4/4	DoReFa [48]	90.6	41.2	58.4
	LSQ [11]	55.5	23.2	53.4
	LLSQ [47]	81.8	34.5	57.8
	APoT [28]	91.6	29.6	55.6
	Choi *et al.* [9]	87.4	58.6	47.2
	ZeroQ [5]	86.6	58.9	48.4
	ZAQ [31]	88.3	59.5	48.5
	AlignQ (Ours)	**96.1**	**59.9**	**61.1**

4.3.2 Comparisons on domain shift data

In addition to the benchmark datasets, we evaluate the effectiveness of AlignQ on domain shift datasets, where the training and testing data are in different domains (non-i.i.d) in transfer learning, including digits [12, 13, 26, 33] and Office-31 [38].

Digits dataset. Table 4 presents the quantization results of DANN [14] (benchmark model in transfer learning). The quantized DANN model under AlignQ at 2-bit precision achieves 10% to 40% accuracy improvements on MNIST → MNIST-M, 3% to 40% accuracy increments on MNIST → SVHN and 4% to 10% improvements on SynDigits → MNIST, because the CDF alignment can effectively align the non-i.i.d. data to i.i.d. (uniform space) to reduce the quantization error (detailed in Sec 3.1). In addition, ADMM optimization regularizes on the changes in data correlations to further lower the error (illustrated Sec. 3.2). Table 4 also manifests that ZSQ approaches are not always superior to QAT (e.g., MNIST → MNIST-M and SynDigits → MNIST), since ZSQ relies on the knowledge from the full-precision model on the training data and thereby tends to generate a larger prediction error in testing data.

Office-31 dataset. Table 5 presents the performances of the quantized DANN [14] on Office-31. It shows that AlignQ in the six domain shift classification tasks outperforms the state-of-the-art, especially in low bit widths. The 5-bit DANN model by AlignQ achieves 71.2% accuracy in

Table 5. Accuracy (%) of quantized DANN (ResNet-50) [14] on Office-31. Three data domains in Office-31 include Amazon (A), Webcam (W), and DSLR (D), thereby indicating six combinations of domain shift classification tasks. The average performance is denoted as "Avg.".

W/A bit	Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg.
32/32	Source only	78.4	94.7	99.1	82.1	58.9	61.0	79.0
	DANN [14]	78.9	95.3	98.2	82.1	59.1	61.8	79.2
	AlignQ (Ours)	**78.9**	**97.1**	**99.1**	**85.7**	**60.6**	**62.9**	**80.6**
4/4	DoReFa [48]	59.6	82.5	90.2	62.5	38.2	45.5	63.1
	APoT [28]	58.5	88.3	85.7	51.8	44.4	46.7	62.6
	Choi *et al.* [9]	12.3	11.7	15.2	10.7	9.5	8.7	11.4
	ZeroQ [5]	11.7	12.3	13.4	9.8	8.0	9.9	10.9
	ZAQ [31]	12.1	12.4	14.2	10.3	7.8	8.9	11.0
	AlignQ (Ours)	**64.9**	**94.2**	**97.3**	**65.2**	**45.6**	**49.7**	**69.5**
5/5	DoReFa [48]	60.2	87.1	92.0	57.1	37.2	45.3	63.2
	APoT [28]	63.7	94.2	92.9	60.7	46.7	48.3	67.8
	Choi *et al.* [9]	67.8	86.6	90.2	67.9	45.5	50.8	68.1
	ZeroQ [5]	67.2	86.6	88.4	67.2	41.6	50.1	66.9
	ZAQ [31]	67.4	87.8	89.5	68.1	43.2	50.2	67.7
	AlignQ (Ours)	**67.8**	**94.7**	**98.2**	**68.8**	**47.7**	**50.2**	**71.2**
8/8	DoReFa [48]	64.9	91.2	93.8	57.1	40.2	47.2	65.7
	Choi *et al.* [9]	67.2	94.2	95.5	72.8	46.2	58.5	72.4
	ZeroQ [5]	67.2	94.2	95.5	72.4	43.1	58.4	71.8
	ZAQ [31]	67.7	94.7	99.1	72.7	45.8	62.9	73.8
	AlignQ (Ours)	**68.4**	**95.3**	**99.1**	**73.2**	**47.7**	**63.0**	**74.5**

overall tasks (vs. Choi *et al.*'s work in 68.1% accuracy), while the 4-bit model by AlignQ obtains 69.5% accuracy, 6% accuracy increment than DoReFa with 63.1% accuracy. In particular, ZSQ (Choi *et al.*'s work, ZeroQ, and ZAQ) at the 4-bit quantization has a significant performance decrement. Accordingly, the results show the ZSQ is more sensitive than QAT (DeReFa and APoT) in the domain shift in training and testing data since ZSQ learns the quantized model depending on the pretrained full-precision model in training data. Moreover, AlignQ has a significant improvement at 4-bit quantization since AlignQ can effectively reduce the quantization error from the discrepancy of the non-i.i.d. training and testing data. In addition to DANN, we also implement the quantized DSAN [49] (the state-of-the-art transfer learning model) on Office-31 in Appendix B.

5. Ablation study

This section evaluates the effectiveness of the proposed CDF alignment and the ADMM-based correlation preservation in AlignQ. Table 6 presents the results of the quantized ResNet models by AlignQ and the baseline uniform quantization (see Eq. (2)). It shows that AlignQ only considers the ADMM correlation preservation (noted as **ADMM only**) can improve the performance of uniform quantization, since we minimize the changes in data correlations that reduce the quantization error (also proved in Proposition 1 in Sec. 3.2). Furthermore, AlignQ with only the CDF alignment in quantization (noted as **CDF only**) obtains a significant improvement particularly at 2-bit quantization, since the alignment process enables the training and testing data to be i.i.d. (see Sec 3.1) to avoid a large quantization error as illustrated in Fig. 1. After we adopt ADMM correlation preservation with the CDF alignment during the quantization process (noted as **CDF + ADMM**), the results validate that the quantization error is further reduced.

Table 6. Effectiveness of AlignQ components. Accuracy (%) of quantized ResNets on CIFAR-10.

Model	Method	W/A bit	Acc (%)	W/A bit	Acc (%)
ResNet-20	Uniform	2 / 2	86.9	4 / 4	91.5
	Ours (ADMM only)	2 / 2	87.3	4 / 4	91.8
	Ours (CDF only)	2 / 2	**90.8**	4 / 4	**92.2**
	Ours (CDF + ADMM)	2 / 2	**91.2**	4 / 4	**92.8**
ResNet-56	Uniform	2 / 2	88.5	4 / 4	89.5
	Ours (ADMM only)	2 / 2	89.5	4 / 4	90.7
	Ours (CDF only)	2 / 2	**91.2**	4 / 4	92.7
	Ours (CDF + ADMM)	2 / 2	**91.7**	4 / 4	**93.2**

Table 7. Effectiveness of AlignQ components. Accuracy (%) of quantized DANN (ResNet-50) [14] on Office-31.

W/A bit	Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg.
5/5	Uniform	54.4	81.5	85.7	48.2	32.1	45.5	57.9
	Ours (ADMM only)	55.5	82.2	86.4	50.1	32.2	46.1	58.8
	Ours (CDF only)	64.9	**94.7**	97.3	67.9	47.4	50.2	70.5
	Ours (CDF + ADMM)	**67.8**	93.6	**98.2**	**68.8**	45.5	**50.2**	70.7
	Ours (Best of all)	**67.8**	**94.7**	**98.2**	**68.8**	**47.4**	**50.2**	**71.2**

Table 7 presents the effectiveness of AlignQ components on the data shift tasks, DANN model on Office-31. The uniform quantization considered with the correlation preservation by ADMM (see Sec. 3.2) outperforms the baseline uniform quantization in each domain shift recognition task. Moreover, when regarding the CDF alignment, the overall performance increases by 12% to 17%, i.e., from 57.9% (uniform) to 70.5% (**CDF only**) and from 58.8% (**ADMM only**) to 70.7% (**CDF + ADMM**). Appendix C the effectiveness of AlignQ components for the quantized DSAN on Office-31.

6. Conclusion

In this paper, we propose *AlignQ* to address non-i.i.d. in training and testing data. We propose CDF alignment to align the data to the same domain (i.i.d) for quantization to minimize the quantization error. Moreover, we prove that the significant changes in data correlations after quantization also induce a larger quantization error. Thus, we design an ADMM optimization process to minimize the discrepancy of data correlations before and after the alignment-quantization process to further lower the quantization error. Experimental results manifest that AlignQ outperforms the state-of-the-art on benchmarks and domain shift datasets especially at low bitwidths.

References

[1] Paszke Adam, Gross Sam, Chintala Soumith, Chanan Gregory, Yang Edward, D Zachary, Lin Zeming, Desmaison Alban, Antiga Luca, and Lerer Adam. Automatic differentia-

tion in pytorch. In *Proceedings of Neural Information Processing Systems*, 2017. 5

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 1

[3] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. Aciq: Analytical clipping for integer quantization of neural networks. 2018. 1, 2, 7

[4] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011. 2, 4

[5] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020. 1, 2, 6, 7, 8

[6] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International conference on machine learning*, pages 2285–2294, 2015. 1

[7] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017. 1

[8] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 1, 2

[9] Yoojin Choi, Jihwan Choi, Mostafa El-Khamy, and Jungwon Lee. Data-free network quantization with adversarial knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 710–711, 2020. 1, 2, 6, 7, 8

[10] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 1

[11] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019. 1, 2, 6, 7

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 2, 5, 7

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 2, 5, 7

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 5, 7, 8

[15] 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. 3

[16] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[17] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[19] Xiangyu He, Jiahao Lu, Weixiang Xu, Qinghao Hu, Peisong Wang, and Jian Cheng. Generative zero-shot network quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3000–3011, 2021. 1, 2, 6, 7

[20] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019. 1

[21] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110:107383, 2021. 2

[22] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017. 1

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[24] James G Kalbfleisch. *Probability and statistical inference*. Springer Science & Business Media, 2012. 2, 3

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2, 5, 7

[27] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016. 1

[28] Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *International Conference on Learning Representations*, 2020. 1, 2, 3, 6, 7, 8

[29] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David Doermann. Towards optimal structured cnn pruning via generative adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2019. 1

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[31] Yuang Liu, Wei Zhang, and Jun Wang. Zero-shot adversarial quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1512–1521, 2021. 1, 2, 6, 7, 8

[32] Daisuke Miyashita, Edward H Lee, and Boris Murmann. Convolutional neural networks using logarithmic data representation. *arXiv preprint arXiv:1603.01025*, 2016. 3

[33] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 2, 5, 7

[34] 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. 1

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[36] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 5

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 5

[38] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 2, 5, 7

[39] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5, 7

[40] Sungho Shin, Yoonho Boo, and Wonyong Sung. Knowledge distillation for optimization of quantized deep neural networks. In *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–6. IEEE, 2020. 2

[41] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 5

[42] Yaman Umuroglu, Nicholas J Fraser, Giulio Gambardella, Michaela Blott, Philip Leong, Magnus Jahre, and Kees Vissers. Finn: A framework for fast, scalable binarized neural network inference. In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 65–74, 2017. 1

[43] Junsong Wang, Qiuwen Lou, Xiaofan Zhang, Chao Zhu, Yonghua Lin, and Deming Chen. Design flow of accelerating hybrid extremely low bit-width neural network in embedded fpga. In *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, pages 163–1636. IEEE, 2018. 1

[44] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 1, 2, 6, 7

[45] Xiangyu Zhang, Jianhua Zou, Xiang Ming, Kaiming He, and Jian Sun. Efficient and accurate approximations of nonlinear convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition*, pages 1984–1992, 2015. 1

[46] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. In *International conference on machine learning*, pages 7543–7552. PMLR, 2019. 2, 6, 7

[47] Xiandong Zhao, Ying Wang, Xuyi Cai, Cheng Liu, and Lei Zhang. Linear symmetric quantization of neural networks for low-precision integer hardware. In *International Conference on Learning Representations*, 2020. 1, 2, 6, 7

[48] 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, 2, 3, 6, 7, 8

[49] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE transactions on neural networks and learning systems*, 2020. 5, 8