Sharpness-Aware Data Generation for Zero-shot Quantization

Hoang Anh Dung [1] **Cuong Pham** [1] **Trung Le** [1] **Jianfei Cai** [1] **Thanh-Toan Do** [1]

Abstract

Zero-shot quantization aims to learn a quantized model from a pre-trained full-precision model with no access to original real training data. The common idea in zero-shot quantization approaches is to generate synthetic data for quantizing the full-precision model. While it is well-known that deep neural networks with low sharpness have better generalization ability, none of the previous zero-shot quantization works considers the sharpness of the quantized model as a criterion for generating training data. This paper introduces a novel methodology that takes into account quantized model sharpness in synthetic data generation to enhance generalization. Specifically, we first demonstrate that sharpness minimization can be attained by maximizing gradient matching between the reconstruction loss gradients computed on synthetic and real validation data, under certain assumptions. We then circumvent the problem of the gradient matching without real validation set by approximating it with the gradient matching between each generated sample and its neighbors. Experimental evaluations on CIFAR-100 and ImageNet datasets demonstrate the superiority of the proposed method over the state-of-the-art techniques in low-bit quantization settings.

1. Introduction

Due to the impressive performance of deep learning models in various fields and applications, there has been great attention to incorporating deep learning models into resource-constrained devices, such as mobiles. As a result, optimizing the storage and computational expense of state-of-the-art deep neural networks (DNNs) has become increasingly important. Among various network compression techniques such as pruning (Molchanov et al., 2019; He et al., 2017), knowledge distillation (Hinton et al., 2014; Romero et al., 2015; Zhao et al., 2022; Pham et al., 2024), and quantization (Courbariaux et al., 2015; Rastegari et al., 2016), network quantization is considered one of the most effective methods. It aims to acquire smaller models with parameters represented in much smaller bit-width (e.g., 1, 2, or 4 bits), yet still achieving competitive performance compared to full-precision (i.e., 32 bits) models (Chen et al., 2021; Dong et al., 2019; Yang & Jin, 2020; Pham et al., 2023; Wei et al., 2022; Nagel et al., 2020; Courbariaux et al., 2015).

Network quantization approaches can be generally divided into two groups: quantization-aware training (QAT) (Krishnamoorthi, 2018; Esser et al., 2020; Nagel et al., 2021) and post-training quantization (PTQ) (Nahshan et al., 2019; Banner et al., 2018; Li et al., 2021; Nagel et al., 2020). While QAT approaches have shown outstanding performance comparable to full-precision models, they often require access to a large amount of real training data and a significant amount of training time. On the other hand, post-training quantization (PTQ) approaches only require a small amount of original data and focus on distilling knowledge from a full precision model to a quantized one.

As an attempt to combat a more challenging scenario where there is no access to any part of the original data (e.g., due to privacy), zero-shot quantization (ZSQ) setting that does not require any original data has been proposed. The common idea in ZSQ (Choi et al., 2021; Jeon et al., 2023; Li et al., 2023) is to generate a set of synthetic data as a calibration set for the quantization process, by taking advantage of information from a full-precision model. Most of the prominent ZSQ works utilize the batch normalization (BN) statistics information in BN layers of the full-precision model, i.e., they try to generate synthetic samples such that the feature distribution of generated samples matches the BN statistics (Choi et al., 2021; Cai et al., 2020). Some other methods pay attention to the boundary information of the full-precision model and try to generate data that are near the decision boundary of the full-precision model (Choi et al., 2021; Li et al., 2023; Qian et al., 2023a). Although different criteria have been investigated when generating synthetic samples in ZSQ (Choi et al., 2021; Jeon et al., 2023; Li et al., 2023; Qian et al., 2023a), none of them considers connections between the generated data and the

[1]Department of Data Science and AI, Monash University, Melbourne, Australia. Correspondence to: Hoang Anh Dung <hoang.dung@monash.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

model's sharpness, despite that it has been shown reducing the model's sharpness can improve its generalization (Foret et al., 2021; Du et al., 2022; Liu et al., 2022).

To this end, in this paper, we propose a novel method for ZSQ that takes into account the impacts of the generated data on the sharpness of the quantized model during the generation process. Specifically, we aim to generate a set of training images such that using these generated images for learning the quantized model will not only result in a good knowledge transfer from the full-precision model but also minimize the sharpness of the quantized model when evaluating on a validation set of real images. However, the real validation set is not available in the context of ZSQ. To overcome this challenge, we rigorously show that under some assumptions, the sharpness minimization can be achieved by maximizing the matching between the gradients of the reconstruction loss evaluated on the generated data and real validation data, respectively. We then circumvent the problem of the gradient matching without real validation set by approximating it with the gradient matching between each generated sample and its neighbors, which can be done through an SAM-like optimization. The contributions of this work can be summarized:

❶ To our knowledge, this paper is the first one that rigorously leverages the Sharpness-Aware Minimization (SAM) as a criterion for generating the training data in the ZSQ problem.
❷ We link the model sharpness with the gradient matching, i.e., maximization of the matching of the gradients of the reconstruction loss w.r.t. model parameters on the generated and validation data leads to sharpness reduction.
❸ We propose a novel approach to approximate gradient matching without the need for a real validation set.
❹ Experimental results demonstrate that our novel Sharpness-Aware Data Generation (SADAG) method outperforms the state-of-the-art ZSQ methods under low-bit quantization settings.

2. Related work

2.1. Uniform quantisation

Uniform quantization is the most popular quantization technique for quantizing DNNs, due to its simplicity. The integer weight of a uniformly quantized model can be determined by the quantizer Q_b as:

$$\hat{w} = Q_b(w; s) = s \times \text{clip}\left(\left\lfloor \frac{w}{s} \right\rceil, n, p\right), \tag{1}$$

where s represents the scaling factor, $\lfloor . \rceil$ denotes the rounding-to-nearest function, and clip() represents the clipping function. For instance, to represent a quantized model with unsigned b bits, we can have $n = 0$, $p = 2^{b-1}$, while $s = \frac{max(w)-min(w)}{2^b-1}$. The recent state-of-the-art post training quantization (PTQ) approaches (Wei et al., 2022; Jeon et al., 2023) have adopted adaptive rounding (Nagel et al., 2020) to improve the performance of uniform quantization further:

$$\hat{w} = s \times \text{clip}\left(\left\lfloor \frac{w}{s} \right\rfloor + h(v), n, p\right) \tag{2}$$

where $h(v) \in [0, 1]$ is a learnable function that converges towards either 0 or 1. In this work, we also adopt the adaptive rounding (Nagel et al., 2020) for weight quantization.

2.2. Zero shot quantization and data generation

To address the lack of calibration data, Zero-shot quantization (ZSQ) approaches often aim to exploit information from full-precision models and generate synthetic data that matches that information. The most intuitive method for synthesizing the data is to utilize the cross-entropy (CE) loss between the prediction of the full-precision model and the synthetic label. However, ZeroQ (Cai et al., 2020), one of the early ZSQ works, argues that the distribution mismatch between synthetic data and real data may lead to a significant gap in performance. Their core idea to tackle this problem is to reconstruct synthetic data based on batch normalization (BN) statistics from the BN layers of the full-precision model, which has led to significant performance improvement. Since then, built upon this idea, more and more data generation methods for ZSQ have been proposed to incorporate other factors. For instance, KMDFQ (Xu et al., 2023) and GDFQ (Xu et al., 2020) assert that ZeroQ (Cai et al., 2020) disregards the class and distribution attributes inherent in the real dataset. To align the distribution characteristics of between the synthetic and real data, they propose to minimize both the BN loss and the cross-entropy loss of synthetic noise, sampled from predefined distributions. Genie (Jeon et al., 2023) directly constructs samples by optimizing according to BN statistics stored in full-precision models, where both the generator and its inputs initialized from a Gaussian distribution are learned simultaneously. Qimera (Choi et al., 2021) and HAST (Li et al., 2023) both propose to generate boundary-supporting samples that they argue would be important for the quantization process, but with fairly different approaches. While HAST tries to reward samples based on their uncertainty from CE loss, Qimera tries to acquire samples lying within the separating boundary between classes by mixing up class embeddings. Other prominent works include DSG (Qin et al., 2021) and AdaDFQ (Qian et al., 2023a). DSG (Qin et al., 2021) points out the lack of diversity in generated data due to BN statistics optimization. To improve the diversity, they add a margin threshold when minimizing BN statistics mismatch. AdaDFQ (Qian et al., 2023a) proposes to combine both boundary and sample diversity optimization. Unfortunately, while different criteria have been proposed

in previous works for the data generation process, most of them did not investigate how those criteria link to the model's generalization in a rigorous way. In this paper, we propose a novel method for ZSQ in which the sharpness of the quantized model is directly taken into account when synthesizing the data.

2.3. Sharpness aware minimization

Sharpness aware minimization (SAM) (Foret et al., 2021) is an approach proposed to improve the model's generalization. By minimizing the loss value and the loss sharpness during training, SAM can guide the model to a flat local optimum, showing promising results. Departing from this idea, various variants of SAM have been proposed to address the shortcomings of this approach. In (Liu et al., 2022), the authors propose LookSAM as an attempt to improve efficiency while still retaining the benefits of the SAM algorithm. They claimed that their method is the first to successfully scale up the batch size when training Vision Transformers with SAM. Another SAM variant is Efficient SAM (Du et al., 2022), which attempts to reduce the computational cost of the original SAM. In each iteration of the training, Efficient SAM optimizes the model's sharpness on a set of weights that are stochastically selected. In addition, the sharpness loss is only evaluated on a subset of data that is sensitive to the model's sharpness. One of the problems with SAM is the gradient conflicts between the perturbed loss and the model's sharpness. Recently, GSAM (Zhuang et al., 2022) and SAGM (Wang et al., 2023) have been proposed to address this gradient conflict problem by modifying the gradient from SAM and removing conflict components.

2.4. Sharpness aware minimization for model quantization

Although there is limited investigation in model generalization, some previous works have attempted to incorporate SAM into model quantization. One of the early works is SAQ (Liu et al., 2021), which demonstrates the existence of a sharper loss landscape in the low-precision model than in the full-precision model. Their work introduces several ways to incorporate sharpness aware loss on optimizing model weights, and proposes an efficient scheme to incorporate SAM without incurring significant cost. The recent work Bit-shrinking (Lin et al., 2023) proposes to gradually reduce the number of bits from high bit-width to the target bit-width such that the sharpness of intermediate quantized models does not go over a threshold.

Another recent work, ZSAQ (Zhu et al., 2023), jointly learns the quantized model and the generator with an adversarial learning strategy through a minimax optimization. It is worth noting that while both ZSAQ (Zhu et al., 2023) and ours have the same objective, i.e., reducing the sharpness of the quantized model, when generating data, their generator does not consider the impact of generated samples on the sharpness of the model, while ours explicitly considers that. In other words, we aim to find a set of synthetic data that is directly beneficial for the model's sharpness over the hidden real data distribution. Unlike our approach, ZSAQ (Zhu et al., 2023) only evaluates the sharpness of the model on the generated data which may not well generalize to real data.

3. Method

3.1. Problem definition

Suppose we have a large validation set $X^{(V)}$, and given a deep learning model $\mathtt{f}(.)$ with its pretrained weight θ_{FP}, our ultimate goal is the generation of a small synthetic dataset $X^{(T)}$, such that using $X^{(T)}$ for learning the quantized model θ_Q will result in a good transfer of knowledge from the full-precision model θ_{FP} to the quantized model θ_Q on the validation set $X^{(V)}$. In the context of model quantization, the transfer knowledge process of a model θ_{FP} to a quantized model θ_Q can be done with the layer-wise reconstruction loss on a calibrated set X as:

$$\mathcal{L}_R(\theta_Q, \theta_{FP}, X) = \frac{1}{2} \sum_{i=1}^{|X|} \sum_{l=1}^{L} \left\| \mathtt{f}(\theta_{FP}, \mathbf{x}_i, l) - \mathtt{f}(\theta_Q, \mathbf{x}_i, l) \right\|^2, \tag{3}$$

where L is the number of layers of the model $\mathtt{f}$; $\mathtt{f}(\theta_{FP}, \mathbf{x}_i, l)$ and $\mathtt{f}(\theta_Q, \mathbf{x}_i, l)$ are respectively the lth-layer outputs of the full-precision and quantized models w.r.t the input sample $\mathbf{x}_i$; $|.|$ denotes the cardinality of a set.

In addition, (ii) in order to enhance the generalization of the quantized model θ_Q, we want that θ_Q locates in a flat (or not too sharp) region of the above $\mathcal{L}_R$ loss landscape. Following (Foret et al., 2021), the flatness of the quantized network θ_Q is defined with the sharpness-aware (SAM) loss:

$$\begin{aligned} &\mathcal{L}_{SAM}(\theta_Q, \theta_{FP}, X^{(V)}) = \mathcal{L}_R(\theta_Q + \epsilon, \theta_{FP}, X^{(V)}) \\ &- \mathcal{L}_R(\theta_Q, \theta_{FP}, X^{(V)}) \\ &\text{s.t: } \epsilon = \arg\max_{||\epsilon|| \leq \rho} \mathcal{L}_R(\theta_Q + \epsilon, \theta_{FP}, X^{(V)}), \end{aligned} \tag{4}$$

where ϵ is a small perpetuation in the neighborhood with radius ρ to the model's weight that increases the loss of the quantized model θ_Q the most. The closed-form solution for ϵ is:

$$\epsilon = \rho \frac{\nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})}{||\nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})||}, \tag{5}$$

where $\nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})$ is the short for $\nabla_{\theta_Q} \mathcal{L}_R(\theta_Q, \theta_{FP}, X^{(V)})$, which is the derivative of the reconstruction loss $\mathcal{L}_R$ between the full-precision model θ_{FP} and the quantized model θ_Q.

In summary, we aim to generate a calibration set $X^{(T)}$ such that a calibration process using $X^{(T)}$ to learn the model θ_Q can minimize the sum of the two losses in Eqs. (3)&(4).

3.2. Sharpness-aware data generation

Suppose that we already initialize a training set $X^{(T)}$. Given the full-precision model θ_{FP}, we start by initializing the quantized model θ_Q without any data, and then calibrate it in the training set by minimizing the reconstruction loss in Eq. (3) (i.e., using $X^{(T)}$ to update θ_Q with one-step gradient descent to get θ_Q^*):

$$\theta_Q^* = \arg\min_{\theta_Q} \mathcal{L}_R(\theta_Q, \theta_{FP}, X^{(T)}). \tag{6}$$

Using Eqs. (3)&(4), we can represent the sum of the SAM loss and the reconstruction loss of the calibrated model θ_Q^*:

$$\begin{aligned}
&\mathcal{L}_{TOTAL}(\theta_Q, \theta_{FP}, X^{(V)}, X^{(T)}) \\
&= \mathcal{L}_{SAM}(\theta_Q^*, \theta_{FP}, X^{(V)}) + \mathcal{L}_R(\theta_Q^*, \theta_{FP}, X^{(V)}) \\
&= \underbrace{\mathcal{L}_R(\theta_Q^* + \epsilon, \theta_{FP}, X^{(V)}) - \mathcal{L}_R(\theta_Q^*, \theta_{FP}, X^{(V)})}_{\text{First term}} \\
&\quad + \underbrace{\mathcal{L}_R(\theta_Q^*, \theta_{FP}, X^{(V)}) - \mathcal{L}_R(\theta_Q, \theta_{FP}, X^{(V)})}_{\text{Second term}} \\
&\quad + \underbrace{\mathcal{L}_R(\theta_Q, \theta_{FP}, X^{(V)})}_{\text{Third term}} \\
&\text{s.t: } \epsilon = \arg\max_{||\epsilon|| \le \rho} \mathcal{L}_R(\theta_Q^* + \epsilon, \theta_{FP}, X^{(V)}).
\end{aligned} \tag{7}$$

The synthetic dataset $X^{(T)}$ that optimizes the quantized model's sharpness and performance can be acquired by minimizing $\mathcal{L}_{TOTAL}$. It is worth noting that the third term $\mathcal{L}_R(\theta_Q, \theta_{FP}, X^{(V)})$ is independent from the training set $X^{(T)}$ that we want to generate, so we can ignore this term.

3.3. Gradient matching

To simplify the representation, for the rest of the paper, we use $\nabla_\theta \mathcal{L}_R(X)$ to stand for $\nabla_\theta \mathcal{L}_R(X, \theta, \theta_{FP})$, which represents the gradient of the reconstruction loss between some model θ with full-precision model θ_{FP} over some data X. Let $\delta_{\theta_Q} = \theta_Q^* - \theta_Q$ be the difference in weights of the quantized model θ_Q before and after calibrated with the training set $X^{(T)}$. In practice, we only use a single-step gradient descent for the model update, so $\delta_{\theta_Q} = -\alpha \nabla_{\theta_Q} \mathcal{L}_R(X^{(T)})$, where α is the learning rate.

Using the first order Taylor expansion for $\mathcal{L}_R(\theta_Q^* + \epsilon, \theta_{FP}, X^{(V)})$ around θ_Q^* in the first term and for $\mathcal{L}(\theta_Q^*, \theta_{FP}, X^{(V)})$ around θ_Q in the *second term* of $\mathcal{L}_{TOTAL}$ in (7), we have:

$$\begin{aligned}
&\mathcal{L}_{TOTAL}(\theta_Q, \theta_{FP}, X^{(V)}, X^{(T)}) \\
&\approx \epsilon^T \nabla_{\theta_Q^*} \mathcal{L}_R(X^{(V)}) - \alpha \nabla_{\theta_Q} \mathcal{L}_R(X^{(T)})^T \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)}),
\end{aligned} \tag{8}$$

where the second term $-\alpha \nabla_{\theta_Q} \mathcal{L}_R(X^{(T)})^T \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})$ is the matching of gradients of the reconstruction loss $\mathcal{L}_R$ w.r.t. the model θ_Q when evaluating the synthetic training set $X^{(T)}$ and the real validation set $X^{(V)}$, respectively. Replace ϵ with its closed-form solution in Eq. (5), the *first term* of $\mathcal{L}_{TOTAL}(.)$ in (8) can be represented as:

$$\begin{aligned}
&\rho \left(\frac{\nabla_{\theta_Q^*} \mathcal{L}_R(X^{(V)})}{\left\| \nabla_{\theta_Q^*} \mathcal{L}_R(X^{(V)}) \right\|} \right)^T \nabla_{\theta_Q^*} \mathcal{L}_R(X^{(V)}) \\
&= \rho \left\| \nabla_{\theta_Q^*} \mathcal{L}_R(X^{(V)}) \right\| \\
&= \rho \left\| \nabla_{\theta_Q^*} \mathcal{L}_R(X^{(V)}) - \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)}) + \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)}) \right\| \\
&\approx \rho || \underbrace{\nabla_{\theta_Q^*} \mathcal{L}_R(X^{(V)}) - \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})}_{\text{First-order Taylor}} + \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)}) || \\
&\approx \rho \left\| \mathcal{H}^{(\theta_Q)} \delta_{\theta_Q} + \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)}) \right\|,
\end{aligned} \tag{9}$$

where $\mathcal{H}^{(\theta_Q)}$ denotes the Hessian matrix over the model weights using the validation set $X^{(V)}$. The final row in (9) is the result of using the first-order Taylor expansion for $\nabla_{\theta_Q^*} \mathcal{L}_R(X^{(V)})$ around θ_Q. From Eq. (9), minimizing the first term of $\mathcal{L}_{TOTAL}(.)$ in (8) is equivalent to minimizing the magnitude of $\mathcal{H}^{(\theta_Q)} \delta_{\theta_Q} + \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})$. It is worth noting that $\nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})$ and $\mathcal{H}^{(\theta_Q)}$ are independent from the calibration set $X^{(T)}$ that we want to generate. The necessity condition for Eq. (9) to be minimized is when $\mathcal{H}^{(\theta_Q)} \delta_{\theta_Q}$ and $\nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})$ have opposite directions, which is equivalent to optimize:

$$\begin{aligned}
&\arg\min_{X^{(T)}} cos(\mathcal{H}^{(\theta_Q)} \delta_{\theta_Q}, \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})) \\
&\equiv \arg\max_{X^{(T)}} cos(\mathcal{H}^{(\theta_Q)} \nabla_{\theta_Q} \mathcal{L}_R(X^{(T)}), \nabla_{\theta_Q} \mathcal{L}_R(X^{(V)})).
\end{aligned} \tag{10}$$

Estimating $\mathcal{H}^{(\theta_Q)}$ and $\nabla_{\theta_Q} \mathcal{L}_R(X^{(T)})$ for the whole model can be very expensive, therefore, we turn to utilize the Hessian matrix $\mathcal{H}_l^{(\theta_Q)}$ and the Jacobian vector $\nabla_{\theta_Q} \mathcal{L}_R(X^{(T)})_l$ of one layer l instead. As many quantization methods (Zhang et al., 2018; Dong et al., 2019; Qian et al., 2023a; Li et al., 2021; Jeon et al., 2023) usually keep the bit-width of the first convolutional layer and the last fully-connected layer higher than other layers in low-bit width setting (e.g., in 2/2 setting, they are still kept at 8 bits while other layers are quantized to 2 bits), optimization based on the gradient of these two layers is likely to have more impact to the model's performance. Therefore, we choose to use the fully-connected layer for our estimation, as it usually has a far higher number of parameters (more influence) than the first convolutional layer and

its Jacobian matrix is easy to compute. For instance, for ResNet-18 architecture, the first convolutional layer has $64 \times 7 \times 7 \times 3 = 9,408$ parameters, while the last fully connected layer has $512 \times 1000 = 512,000$ parameters.

Let $\mathcal{A}^{(T)} = \mathtt{f}(\theta_Q, X^{(T)}, L-1)$ and $\mathcal{A}^{(V)} = \mathtt{f}(\theta_Q, X^{(V)}, L-1)$ denote respectively input for the fully-connected layer of the training and validation sets with size $N \times d$ where N is the number of samples, and $\mathcal{W}_{(L)}$ be the weight matrix of the fully-connected layer with size $d \times C$ where C is the number of classes. We have:

$$\begin{aligned} \mathtt{f}(\theta_Q, X^{(T)}, L) &= \mathcal{A}^{(T)}\mathcal{W}_{(L)} \\ \mathtt{f}(\theta_Q, X^{(V)}, L) &= \mathcal{A}^{(V)}\mathcal{W}_{(L)}. \end{aligned} \tag{11}$$

From Eq. (11), the Jacobian matrix w.r.t the fully connected layer can be easily estimated as:

$$\nabla_{\theta_Q}\mathcal{L}_R(X^{(T)})_{(L)} = flatten\left({\mathcal{A}^{(T)}}^T g^{(T)}\right), \tag{12}$$

where $g_i^{(T)} = \mathtt{f}(\theta_Q, \mathbf{x}_i, L) - \mathtt{f}(\theta_{FP}, \mathbf{x}_i, L)$. As the fully-connected layer is just a linear layer, the Hessian matrix $\mathcal{H}_{(L)}^{(\theta_Q)}$ of this layer has the size $dC \times dC$ and can be represented be:

$$\begin{pmatrix} {\mathcal{A}^{(V)}}^T\mathcal{A}^{(V)} & 0 & \dots & 0 \\ 0 & {\mathcal{A}^{(V)}}^T\mathcal{A}^{(V)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & {\mathcal{A}^{(V)}}^T\mathcal{A}^{(V)} \end{pmatrix} \tag{13}$$

We can see that the Hessian matrix in (13) has a positive diagonal, its diagonal elements are periodically overlapped and most of its off-diagonal elements are 0. Note that this matrix is independent of the training set $X^{(T)}$ that we want to generate. Similar to AdaRound (Nagel et al., 2020), we assume that $\mathcal{H}_{(L)}^{(\theta_Q)}$ is a diagonal matrix with the same main diagonal value:

$$\mathcal{H}_{(L)}^{(\theta_Q)} = cI, \tag{14}$$

where c is some constant, and I denotes the identity matrix. Then (10) is equivalent to:

$$\arg\max_{X^{(T)}} cos(\nabla_{\theta_Q}\mathcal{L}_R(X^{(T)}), \nabla_{\theta_Q}\mathcal{L}_R(X^{(V)})). \tag{15}$$

As (15) and the second term of $\mathcal{L}_{TOTAL}(.)$ in Eq. (8) correlate, optimizing (15) will also partly optimize the second term in Eq. (8).

3.4. Gradient matching without a validation set

From (15), intuitively, we can see that the small set $X^{(T)}$ that we want to generate for calibration can decrease the SAM loss of the model over the validation set $X^{(V)}$ when the gradient of the model when evaluating on $X^{(T)}$ matches with that when evaluating on $X^{(V)}$. In summary, there is a strong correlation between the minimization of the SAM loss and the matching of the gradients of the reconstruction loss $\mathcal{L}_R(.)$ w.r.t. θ_Q evaluated on the calibrated set and the validation set. Unfortunately, this objective is still hard to optimize because we do not have a large validation set $X^{(V)}$. Thus, we propose a novel approach to approximate (15). Our idea is that if the training set is diverse enough,i.e., it spans over the data space, then for each sample $\mathbf{x}_i^{(T)}$ that we want to generate, we just need to optimize its gradient to match with the gradients of samples in its neighborhood. In other word, we want to generate a training set $X^{(T)}$ such that samples of $X^{(T)}$ are diverse and each sample of $X^{(T)}$ maximizes the gradient matching with its neighbors. To this end, we introduce an approach similar to SAM. We want to minimize the difference between the gradients of each generated sample and a sample in its neighbor with the highest gradient dissimilarity. Specifically, given a generated training set $X^{(T)} := \{\mathbf{x}_i^{(T)} | \mathbf{x}_i^{(T)} = \mathcal{G}(\mathbf{z}_i)\}$, with $\mathcal{G}(.)$ is a generator and $Z := \{\mathbf{z}_i\}_{i=1}^{|X^{(T)}|}$ is the set of embedding vectors to generate $X^{(T)}$, we want to estimate the perturbation $\epsilon^{(\mathcal{N})} = \{\epsilon_i^{(\mathcal{N})}\}_{i=1}^{|X^{(T)}|}$ such that $\mathbf{x}^{*(T)}_i = \mathcal{G}(\mathbf{z}_i + \epsilon_i^{(\mathcal{N})})$ is the sample with the least similarity in gradient within the

Algorithm 1 SA zero-shot quantization.

Train(θ_{FP},$\mathcal{G}$,T,N_w,N_g,N_q).
θ_{FP}: The full-precision model.
$\mathcal{G}$: The generator.
T: Number of images we want to generate.
N_w: Number of warm-up iterations.
N_g: Number of iterations for data optimization.
N_q: Number of iterations for model quantization.
Initialize θ_Q from θ_{FP}.
Initialize $\mathcal{G}$ and $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Warm up θ_Q, $\mathcal{G}$, z with BN loss (20) for N_w iterations.
for $j = 1$ to N_g **do**
  for $i = 1$ to T **do**
    $\epsilon_i^{(\mathcal{N})} = \nu \frac{\nabla_{\mathbf{z}_i}\mathcal{D}(\nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)), \nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)))}{\left\|\nabla_{\mathbf{z}_i}\mathcal{D}(\nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)), \nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)))\right\|}$
    $\mathbf{z}_i = \mathbf{z}_i - \gamma\partial\frac{\mathcal{L}_{FINAL}}{\mathbf{z}_i}$
    $\mathcal{G} = \mathcal{G} - \gamma\partial\frac{\mathcal{L}_{FINAL}}{\mathcal{G}}$
  end for
end for
Get the dataset $X^{(T)} := \{\mathbf{x}_i^{(T)} | \mathbf{x}_i^{(T)} = \mathcal{G}(\mathbf{z}_i)\}$.
for $t = 1$ to N_q **do**
  $\theta_Q = \theta_Q - \alpha_\theta \frac{\partial\mathcal{L}_R(\theta_Q, \theta_{FP}, X^{(T)})}{\partial\theta_Q}$.
end for
return θ_Q.

neighborhood of the original samples $\mathbf{x}_i^{(T)}$:

$$\epsilon_i^{(\mathcal{N})} = \arg\max_{\epsilon_i^{(\mathcal{N})}} \mathcal{D}(\nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)), \nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i + \epsilon_i^{(\mathcal{N})})))$$
$$\text{s.t: } ||\epsilon_i^{(\mathcal{N})}|| \leq \nu, \tag{16}$$

where $\mathcal{D}(.)$ is a loss that measures the difference in gradients, assumed to be the cosine similarity distance. Following (Foret et al., 2021), the closed-form solution for $\epsilon_i^{(\mathcal{N})}$ is:

$$\epsilon_i^{(\mathcal{N})} = \nu \frac{\nabla_{\mathbf{z}_i}\mathcal{D}(\nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)), \nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)))}{\left\|\nabla_{\mathbf{z}_i}\mathcal{D}(\nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)), \nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)))\right\|}. \tag{17}$$

Given $\epsilon^{(\mathcal{N})}$, we can estimate the sample that has its gradient matches with the gradients of its neighborhoods:

$$\begin{aligned}\mathbf{z}_i, \mathcal{G} &= \arg\min_{\mathbf{z}_i, \mathcal{G}} \mathcal{L}_{GRAD}(\theta_Q, \theta_{FP}, X^{(T)}) \\ &= \arg\min_{\mathbf{z}_i, \mathcal{G}} \mathcal{D}(\nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i)), \nabla_{\theta_Q}\mathcal{L}_R(\mathcal{G}(\mathbf{z}_i + \epsilon_i^{(\mathcal{N})}))) \\ &\forall i = 1, 2, .., |X^{(T)}|,\end{aligned} \tag{18}$$

The above gradient matching loss will encourage (15) if the calibration set $X^{(T)}$ is diverse enough. Therefore, after normalizing $\nabla_{\theta_Q}\mathcal{L}_R(X^{(T)})$ to be unit vectors, we add a term to encourage the diversity of the calibration set:

$$\begin{aligned}&\mathcal{L}_{DIVERSE}(\theta_Q, X^{(T)}) \\ &= \sum_{i,j,i\neq j} \max(0, abs(\nabla_{\theta_Q}\mathcal{L}_R^{(i)}{}^T \nabla_{\theta_Q}\mathcal{L}_R^{(j)}) - \zeta),\end{aligned} \tag{19}$$

where $abs(.)$ denotes the absolute function; $\nabla_{\theta_Q}\mathcal{L}_R^{(j)} = \nabla_{\theta_Q}\mathcal{L}_R(\mathbf{x}_j)$ represents the gradient of the reconstruction loss w.r.t. θ_Q for sample $\mathbf{x}_j$; ζ is a small positive threshold. This loss encourages gradients of samples x_i and x_j $(i \neq j)$ to be orthogonal when ζ is close to 0. Consequently, it encourages the calibration set to be diverse.

Verification of the gradient matching on real data. To verify the effectiveness of the gradient matching for the quantization process, we conduct experiments using the gradient matching on real images. Given a set of real images, we compare the performance of the quantized model calibrated over a randomly extracted subset of data and a subset of data that minimizes the gradient matching loss in (15). The results presented in Table 1 show that the gradient matching loss (15) consistently improves the performance of the quantized model. The improvements are more significant when the number of samples is small.

Table 1. The comparative performance when quantizing the ResNet-18 model using the real data randomly selected and the real data selected by (15).

Num. Images	32	64	128	256
Random	32.67	43.15	49.45	53.64
SADAG (Ours)	35.46	44.05	50.30	54.01

3.5. Final algorithm

Besides the sharpness of the model, we also want the generated samples to follow the distribution of original data. Particularly, we encourage $X^{(T)}$ to have similar BN statistics stored in the BN layers of the full-precision model θ_{FP}, by introducing the BN loss L_{BN}:

$$\mathcal{L}_{BN}(\theta_{FP}, X^{(T)}) = \sum_{j=1}^{L}(||\mu_j^{(s)} - \mu_j||^2 + ||\sigma_j^{(s)} - \sigma_j||^2), \tag{20}$$

where $\mu_j^{(s)}$ and μ_j are respectively the mean output values of the synthetic dataset $X^{(T)}$ from the full-precision model at the jth layer and the BatchNorm statistic of the full-precision model from the same layer, while $\sigma_j^{(s)}$ and σ_j are the corresponding standard deviations. Initially, we need to warm up the calibrated set $X^{(T)}$ using a data generation method. After the warm-up stage, we acquire the final calibration set $X^{(T)}$ by minimizing the loss in Eq. (21) over the warm-up data. Finally, this newly generated training set will be used to calibrate the current quantized model.

$$\begin{aligned}\mathcal{L}_{FINAL}(\theta_{FP}, X^{(T)}) &= \mathcal{L}_{BN}(\theta_{FP}, X^{(T)}) + \\ &\lambda_1\mathcal{L}_{DIVERSE}(\theta_Q, X^{(T)}) + \lambda_2\mathcal{L}_{GRAD}(\theta_Q, \theta_{FP}, X^{(T)}).\end{aligned} \tag{21}$$

Algorithm 1 gives the overall algorithm of our proposed method.

4. Experiments

4.1. Experimental setup

Dataset and network architecture. We evaluate our approach on CIFAR-100 (Krizhevsky et al., 2009) and ImageNet (Russakovsky et al., 2015) datasets, which are commonly utilized for zero-shot quantization. Following the settings in (Qian et al., 2023a; Li et al., 2023; Choi et al., 2021), we evaluate our proposed method using ResNet-20 (He et al., 2016) model for CIFAR-100 dataset. For the ImageNet dataset, we validate our proposed approach using ResNet-18 (He et al., 2016), ResNet-50 (He et al., 2016), and MobileNetV2 (Sandler et al., 2018) architectures.

Quantization setting. Following Genie (Jeon et al., 2023), we adopt the uniform quantization scheme with adaptive rounding approach (Nagel et al., 2020), as elaborated in Section 2.1. In our experiments, the bit-widths of the first layer and the last layer are fixed at 8 bits, which is similar to recent SOTA methods for PTQ (Jeon et al., 2023; Li et al., 2021). Following BRECQ (Li et al., 2021) and Genie (Jeon et al., 2023), we also set the activation bit-widths of the second layer and the last layer to 8 bits. The remaining weight and activation bit-widths of other layers follow the specifications in the corresponding experimental setups.

Implementation details. For the initial warm up of the synthetic dataset, we adopt a generator and a set of embedding vectors with 256 dimensions for each mini-batch of images that we generate, similar to the Genie model (Jeon et al., 2023). We use the same quantization setting as BRECQ (Li et al., 2021). The learning rates of the generator and embedding are initially set at 0.1 and 0.01, respectively. We adopt the Adam optimizer (Kingma & Ba, 2014) for both generator and data embedding, but utilize different schedulers for them, i.e., the ExponentialLR scheduler and ReduceLRonPlateau scheduler are used for scheduling the learning rates of the generator and the embeddings, respectively. Across all experiments, the batch size for the data generation process is set to 128, while in the quantization step, we keep the batch size at 32. The threshold ζ in Eq. (19) is set to 0 or 0.1. The radius ν in Eq. (16) for the embedding perturbation is set to 2. To demonstrate the effectiveness of our method, we compare our model's performance with the recent state-of-the-art models on different low-bit ZSQ settings. Following previous works (Jeon et al., 2023; Qian et al., 2023a) and combined with an additional extreme low-bit setting (2/2), we use a total of 4 different quantization settings, including 2/2, 3/3, 2/4, and 4/4 bit-width for ImageNet experiments. On the other hand, for CIFAR-100, we report the results of 3/3 and 4/4 bit-width settings. Our model is then compared with recent prominent zero-shot quantization models, i.e., Qimera (Choi et al., 2021), AdaSG (Qian et al., 2023b), IntraQ (Zhong et al., 2022), AdaDFQ (Qian et al., 2023a) and Genie (Jeon et al., 2023). Regarding the number of generated images, we generate a total of 1,024 images for a fair comparison with Genie (Jeon et al., 2023).

4.2. Comparison with the state of art

Table 2 represents the comparative results of our method SADAG and other state-of-the-art (SOTA) methods when evaluated on the CIFAR-100 dataset. The results of (Choi et al., 2021; Qian et al., 2023b; Zhong et al., 2022; Qian et al., 2023a) are cited from (Qian et al., 2023a), while for Genie (Jeon et al., 2023), we use their official released code and adapt it for the ResNet-20 model. The results confirm the superior performance of the proposed SADAG over the state of the art. Comparing to Genie (Jeon et al., 2023), the current SOTA model in ZSQ, our method improves over Genie 0.69% and 0.76% for the 3/3 and 4/4 settings, respectively.

Table 3 presents the comparative results of our method SADAG and other state-of-the art methods when evaluated on the ImageNet dataset. It is worth noting that Genie (Jeon et al., 2023) – the SOTA method for ZSQ does not report results for the 2/2 and 3/3 settings. We use their official released code to produce Genie's results for those settings. Regarding other results of the competitors, except for the 2/4 setting of Genie with the MobileNetV2 architecture which we produce it result by using the Genie's released code[1], they are cited from (Qian et al., 2023a; Jeon et al., 2023).

Table 2. Comparisons of Top-1 classification accuracy (%) with the state of the art on CIFAR-100 dataset.

Method	Bit-width (W/A)	ResNet-20 (FP: 70.33)
Qimera (Choi et al., 2021)	3/3	46.13
AdaSG (Qian et al., 2023b)		52.76
IntraQ (Zhong et al., 2022)		-
AdaDFQ (Qian et al., 2023a)		52.74
Genie (Jeon et al., 2023)		65.25
SADAG (Ours)		**65.94**
Qimera (Choi et al., 2021)	4/4	65.10
AdaSG (Qian et al., 2023b)		66.42
IntraQ (Zhong et al., 2022)		64.98
AdaDFQ (Qian et al., 2023a)		66.81
Genie (Jeon et al., 2023)		68.35
SADAG (Ours)		**69.11**

The results in Table 3 show that our method SADAG consistently outperforms previous approaches, including the current best method Genie (Jeon et al., 2023) on all bit-width settings and all considered model architectures, which confirms the effectiveness of the proposed sharpness-aware data generation approach. The improvements of our method over Genie are more clear in the 2/2 setting, i.e., the improvements are 0.77%, 0.74%, and 1.08% for the ResNet-18, ResNet-50, and MobileNetV2, respectively.

4.3. Visualization and ablation studies

4.3.1. Visualization

In Figure 1, we present several examples of the warm-up images and the corresponding synthetic images generated by our proposed method. The images for visualization are taken from the setting 3/3 with CIFAR-100 dataset with ResNet-20 model. In each corner of the figure, we show three different images that respectively represent the images after the warm-up stage (i.e., warm-up images), the final synthetic images, and the heat map demonstrating the pixel value differences between them. Because the BatchNorm loss converges very fast, the semantics of the images do not change much after the warm-up stage. However, we can observe that our method appears to make the images smoother compared to those in the warm-up stage, i.e., there is less variability in color in the generated images compared to the warm-up images.

[1]While we are able to reproduce other results of Genie, we are unable to reproduce its result for the 2/4 setting with MobileNetV2. There is a large gap between the reported number in their paper and the reproduced result, i.e., 53.38 vs. 51.47.

Table 3. Comparisons of Top-1 classification accuracy (%) with the state of the art on ImageNet dataset. The result denoted with (*) is reproduced using the official released code of the corresponding paper.

Method	Bit-width (W/A)	ResNet-18	ResNet-50	MobileNetV2
	Full precision	71.01	76.63	72.20
Genie (Jeon et al., 2023)	2/2	53.74	56.81	11.93
SADAG (Ours)		**54.51**	**57.55**	**13.01**
Genie (Jeon et al., 2023)	2/4	65.10	69.99	51.47*
SADAG (Ours)		**65.25**	**70.52**	**51.89**
Qimera (Choi et al., 2021)	3/3	1.17	-	-
AdaSG (Qian et al., 2023b)		37.04	16.98	26.90
IntraQ (Zhong et al., 2022)		-	-	-
AdaDFQ (Qian et al., 2023a)		38.10	17.63	28.99
Genie (Jeon et al., 2023)		66.89	72.54	55.31
SADAG (Ours)		**67.10**	**72.62**	**56.02**
Qimera (Choi et al., 2021)	4/4	63.84	66.25	61.62
AdaSG (Qian et al., 2023b)		66.50	68.58	65.15
IntraQ (Zhong et al., 2022)		66.47	-	65.10
AdaDFQ (Qian et al., 2023a)		66.53	68.38	65.41
Genie (Jeon et al., 2023)		69.66	75.59	68.38
SADAG (Ours)		**69.72**	**75.7**	**68.54**

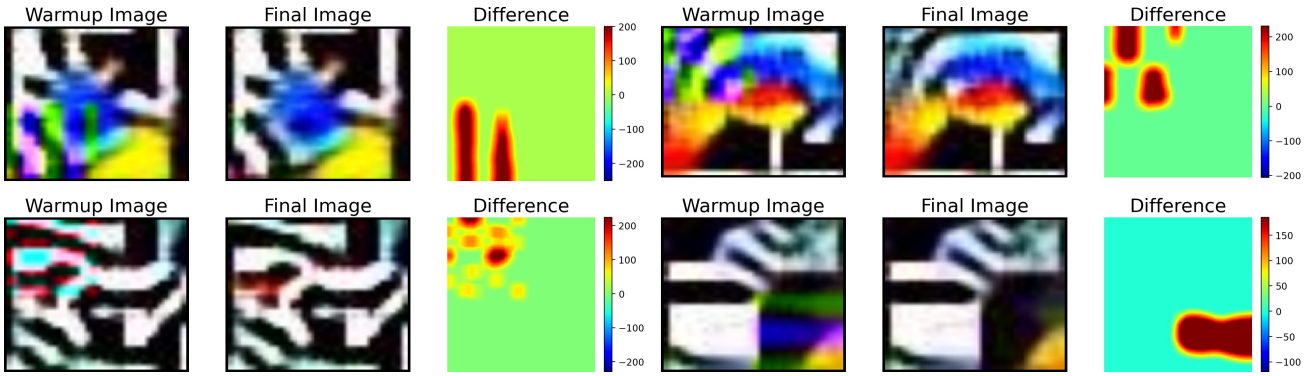

Figure 1. The warm-up images and the corresponding images generated by our proposed method SADAG, and the corresponding heat maps of their differences.

4.3.2. Impact of the number of generated images

We conduct an ablation study about the model's performance gain with different numbers of generated images, ranging from 128 to 1024. We use ResNet-18 architecture and the ImageNet dataset for evaluation, with the 2/2 bit-width for this experiment. The results are presented in Table 4. The results show that increasing the number generated images improves the model's performance. However, the performance gain is smaller when the number of images increase, e.g., for the proposed method, the performance gains are 3.71% and 1.94% when increasing the number of images from 128 to 256, and from 256 to 512, respectively. In addition, the results show that the proposed method consistently outperforms over Genie (Jeon et al., 2023).

Table 4. Comparative results between our method and Genie (Jeon et al., 2023) with different numbers of generated images.

Num. Images	128	256	512	1024
Genie	47.17	50.46	52.79	53.74
SADAG (Ours)	47.54	51.25	53.19	54.51

4.3.3. Impact of the loss factors λ_1 and λ_2

We conduct an ablation study to assess the impact of the hyperparameters λ_1 and λ_2 in Eq. (21). Specifically, we respectively keep one of the two variables λ_1 and λ_2 fixed at 1, while varying the other with five different values $0, 0.5, 1, 2, 5$. The experiments are conducted with ResNet-18 with the 2/2 setting and the number of generated images is 1024 for each experiment. The results are presented in Tables 5 and 6. As we can see, the model's performance degrades with larger or smaller λ_1 and λ_2. Therefore, we simply keep both of them at value 1.

Table 5. Change in performance w.r.t. λ_1 in Eq. (21).

λ_1	0	0.5	1	2	5
SADAG	54.14	54.15	54.51	53.99	53.85

Table 6. Change in performance w.r.t. λ_2 in Eq. (21).

λ_2	0	0.5	1	2	5
SADAG	53.74	54.22	54.51	54.03	53.77

4.3.4. Computation expense

Although the second-order objective in Eq. (7) is computationally intensive, we have successfully reduced the computational expense by approximating it with another first-order optimization in Eq. (15). Our proposed method operates at a speed that is approximately 1.5 times slower than Genie, which solely utilizes BatchNorm loss for optimization and is currently one of the fastest zero-shot quantization methods.

5. Conclusion

In this paper, we propose SADAG, a novel approach for data-free quantization that takes into consideration the sharpness of the model calibrated on the synthetic dataset. We have elucidated the relationship between gradient matching between the training and validation sets and its influence on the calibrated model's sharpness on the validation set. Our findings illustrate that enhancing the state-of-the-art generative data-free quantization can be accomplished without significant additional computational overhead or the necessity for any original data; simply by attending to the gradient of the neighborhood of each generated sample, indirect sharpness optimization over the hidden validation set is feasible. Extensive experimentation on two benchmark datasets underscores SADAG's achievement of state-of-the-art performance. However, there still exist several limitations in our approach. A noticeable weakness of the framework is that it requires the relaxation for the Hessian matrix. As future work, we will tackle this problem by integrating the model with some techniques to approximate the Hessian matrix. Another limitation is that our current method only takes into account the gradient of the final fully-connected layer instead of the whole model. Although accurate gradient matching approximation for the whole network is very computationally expensive, as potential improvement in the future, we can try to extend the method to match the gradient of several blocks instead of a single layer.

Acknowledgements

Trung Le was partly supported by ARC DP23 grant DP230101176 and by the Air Force Office of Scientific Research under award number FA2386-23-1-4044.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Banner, R., Nahshan, Y., Hoffer, E., and Soudry, D. Aciq: Analytical clipping for integer quantization of neural networks. *ArXiv*, 2018.

Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. Zeroq: A novel zero shot quantization framework. *CVPR*, 2020.

Chen, W., Wang, P., and Cheng, J. Towards mixed-precision quantization of neural networks via constrained optimization. *ICCV*, 2021.

Choi, K., Hong, D., Park, N., Kim, Y., and Lee, J. Qimera: Data-free quantization with synthetic boundary supporting samples. *NeurIPS*, 2021.

Courbariaux, M., Bengio, Y., and David, J.-P. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NeurIPS*, 2015.

Dong, Z., Yao, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. Hawq: Hessian aware quantization of neural networks with mixed-precision. *ICCV*, 2019.

Du, J., Yan, H., Feng, J., Zhou, J. T., Zhen, L., Goh, R. S. M., and Tan, V. Efficient sharpness-aware minimization for improved training of neural networks. In *ICLR*, 2022.

Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. In *ICLR*, 2020.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

He, Y., Zhang, X., and Sun, J. Channel Pruning for Accelerating Very Deep Neural Networks. In *ICCV*, 2017.

Hinton, G., Vinyals, O., and Dean, J. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*, 2014.

Jeon, Y., Lee, C., and Kim, H.-y. Genie: Show me the data for quantization. In *CVPR*, 2023.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper. *ArXiv*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, H., Wu, X., Lv, F., Liao, D., Li, T. H., Zhang, Y., Han, B., and Tan, M. Hard sample matters a lot in zero-shot quantization. *CVPR*, 2023.

Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., and Gu, S. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *ICLR*, 2021.

Lin, C., Peng, B., Li, Z., Tan, W., Ren, Y., Xiao, J., and Pu, S. Bit-shrinking: Limiting instantaneous sharpness for improving post-training quantization. In *CVPR*, 2023.

Liu, J., Cai, J., and Zhuang, B. Sharpness-aware quantization for deep neural networks. *ArXiv*, 2021.

Liu, Y., Mai, S., Chen, X., Hsieh, C.-J., and You, Y. Towards efficient and scalable sharpness-aware minimization. *CVPR*, 2022.

Molchanov, P., Mallya, A., Tyree, S., Frosio, I., and Kautz, J. Importance estimation for neural network pruning. In *CVPR*, 2019.

Nagel, M., Amjad, R. A., Van Baalen, M., Louizos, C., and Blankevoort, T. Up or down? adaptive rounding for post-training quantization. In *ICML*, 2020.

Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., van Baalen, M., and Blankevoort, T. A white paper on neural network quantization. *ArXiv*, 2021.

Nahshan, Y., Chmiel, B., Baskin, C., Zheltonozhskii, E., Banner, R., Bronstein, A. M., and Mendelson, A. Loss aware post-training quantization. *Machine Learning*, 110: 3245 – 3262, 2019.

Pham, C., Hoang, T., and Do, T.-T. Collaborative multi-teacher knowledge distillation for learning low bit-width deep neural networks. In *WACV*, 2023.

Pham, C., Nguyen, V.-A., Le, T., Phung, D., Carneiro, G., and Do, T.-T. Frequency attention for knowledge distillation. In *WACV*, 2024.

Qian, B., Wang, Y., Hong, R., and Wang, M. Adaptive data-free quantization. *CVPR*, 2023a.

Qian, B., Wang, Y., Hong, R., and Wang, M. Rethinking data-free quantization as a zero-sum game. *AAAI*, 2023b.

Qin, H., Ding, Y., Zhang, X., Wang, J., Liu, X., and Lu, J. Diverse sample generation: Pushing the limit of generative data-free quantization. *TPAMI*, 2021.

Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In *ECCV*, 2016.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Wang, P., Zhang, Z., Lei, Z., and Zhang, L. Sharpness-aware gradient matching for domain generalization. *CVPR*, 2023.

Wei, X., Gong, R., Li, Y., Liu, X., and Yu, F. QDrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *ICLR*, 2022.

Xu, S., Li, H., Zhuang, B., Liu, J., Cao, J., Liang, C., and Tan, M. Generative low-bitwidth data free quantization. *ArXiv*, abs/2003.03603, 2020. URL https://api.semanticscholar.org/CorpusID:212633494.

Xu, S., Zhang, S., Liu, J., Zhuang, B., Wang, Y., and Tan, M. Generative data free model quantization with knowledge matching for classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12):7296–7309, 2023. doi: 10.1109/TCSVT.2023.3279281.

Yang, L. and Jin, Q. Fracbits: Mixed precision quantization via fractional bit-widths. In *AAAI*, 2020.

Zhang, D., Yang, J., Ye, D., and Hua, G. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *ECCV*, 2018.

Zhao, B., Cui, Q., Song, R., Qiu, Y., and Liang, J. Decoupled knowledge distillation. In *CVPR*, 2022.

Zhong, Y., Lin, M., Nan, G., Liu, J., Zhang, B., Tian, Y., and Ji, R. Intraq: Learning synthetic images with intra-class heterogeneity for zero-shot network quantization. In *CVPR*, 2022.

Zhu, M., Zhong, Q., Shen, L., Ding, L., Liu, J., Du, B., and Tao, D. Zero-shot sharpness-aware quantization for pre-trained language models. In *EMNLP*, 2023.

Zhuang, J., Gong, B., Yuan, L., Cui, Y., Adam, H., Dvornek, N. C., sekhar tatikonda, s Duncan, J., and Liu, T. Surrogate gap minimization improves sharpness-aware training. In *ICLR*, 2022.

A. Appendix