Efficient and Robust Quantization-aware Training via Adaptive Coreset Selection

Anonymous authors
Paper under double-blind review

Abstract

Quantization-aware training (QAT) is a representative model compression method to reduce redundancy in weights and activations. However, most existing QAT methods require end-to-end training on the entire dataset, which suffers from long training time and high energy costs. In addition, the potential label noise in the training data undermines the robustness of QAT. In this work, we show that we can improve data efficiency and robustness with a proper data selection strategy designed specifically for QAT. We propose two metrics based on analysis of loss and gradient of quantized weights: error vector score and disagreement score, to quantify the importance of each sample during training. Guided by these two metrics, we proposed a quantization-aware **A**daptive **C**oreset **S**election (**ACS**) method to select the data for the current training epoch. We evaluate our method on various networks (ResNet-18, MobileNetV2, RetinaNet), datasets(CIFAR-10, CIFAR-100, ImageNet-1K, COCO), and under different quantization settings. Specifically, our method can achieve an accuracy of 68.39% of 4-bit quantized ResNet-18 on the ImageNet-1K dataset with only a 10% subset, which has an absolute gain of 4.24% compared to the baseline. Our method can also improve the robustness of QAT by removing noisy samples in the training set.

1 Introduction

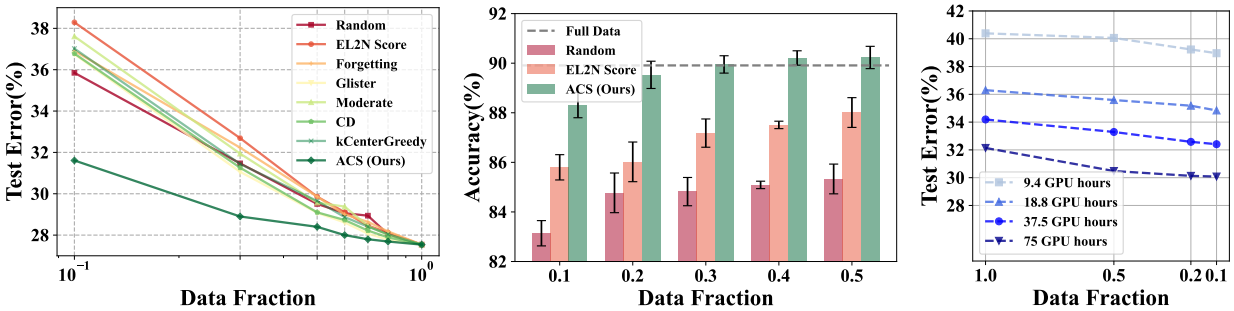


Figure 1: **Left:** Data scaling curve for 4-bit quantized ResNet-18 on the ImageNet-1K dataset. Our ACS significantly reduces test error using the same training data fraction compared to baselines. **Middle:** Accuracy of 2/32-bit quantized ResNet-18 trained on CIFAR-10 with 10% random label noise. Our ACS is the only method to outperform the full data training performance by effectively removing noisy samples. **Right:** Test error for quantized ResNet-18 with different data fraction and the same GPU training hours. Under the same training budget, QAT with smaller coreset selected by ACS outperforms full data training.

Deep learning models have achieved remarkable achievements across various applications, including computer vision (Krizhevsky et al., 2012; He et al., 2016; Tan & Le, 2019; Kirillov et al., 2023) and natural language processing (Kenton & Toutanova, 2019; Yang et al., 2019; Conneau & Lample, 2019; Touvron et al., 2023). The outstanding performance of these models can be attributed to their large number of parameters and the availability of large-scale training data. For example, GPT-3 (Brown et al., 2020) boasts 175 billion

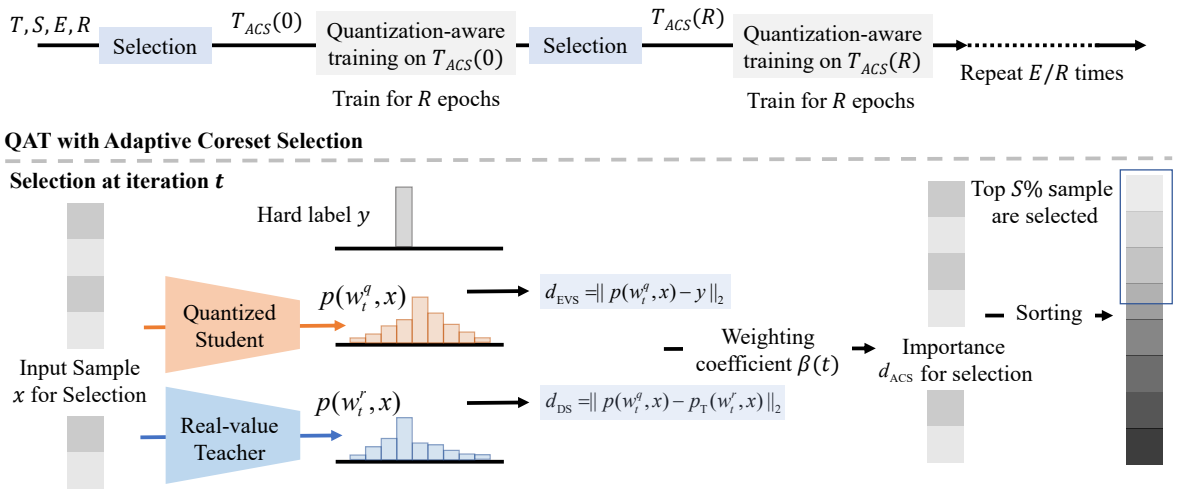


Figure 2: An overview of the Adaptive Coreset Selection (ACS) for efficient and robust QAT.

parameters and the pre-training is performed on the dataset with more than 410 billion tokens. Neural scaling law Kaplan et al. (2020); Alabdulmohsin et al. (2022) is an explanation of this size expansion that empirically demonstrates test loss of a model follows a power law to the model size, training dataset size, and amount of computation. While neural scaling law guides the expanding usage of large models, the large model size and training set scale have become the most significant challenges for the training and deployment of deep learning models, especially on edge devices with computation and storage constraints.

Many model compression methods have been proposed recently to address these challenges and enable the effective deployment of deep learning models. These model compression techniques include quantization (Zhou et al., 2016; Choi et al., 2018; Esser et al., 2020; Bhalgat et al., 2020), pruning (Liu et al., 2017; 2018; Molchanov et al., 2019; Liu et al., 2019), knowledge distillation (Hinton et al., 2015; Park et al., 2019; Shen & Xing, 2021), and compact network design (Howard et al., 2017; Pham et al., 2018). Among the aforementioned methods, quantization methods have been the most widely adopted because they have the advantage of promising hardware affinity across different architectures (Judd et al., 2016; Jouppi et al., 2017; Sharma et al., 2018). To minimize the performance gap between the quantized and full-precision models, quantization-aware training (QAT) is often utilized. Although QAT has improved the inference efficiency of the target model, it is computation-intensive and requires more time than full-precision training. While previous QAT methods assume an ideal case in which computation resources are unlimited, we aim to research a more practical case when the cost of QAT should be considered.

Coreset selection techniques aim to mitigate the high training cost and potential negative influence of label noise to improve data efficiency and robustness for full-precision training. Specifically, coreset selection methods leverage the redundancy in training datasets and select the most informative data to build a coreset for training. One recent research Sorscher et al. (2022) has pointed out that coreset can be a solution to beat the scaling law. Previous methods select data based on feature (Agarwal et al., 2020; Sener & Savarese, 2018), error (Paul et al., 2021), gradient (Killamsetty et al., 2021a), and decision boundary (Margatina et al., 2021). While these methods can achieve notable efficiency improvement for full-precision training, their effectiveness on low-precision QAT has not been explored before. To utilize coreset selection methods to improve the data efficiency and robustness of QAT, the characteristics of quantized weights must be considered in the design of methods. Existing full-precision methods are not designed for quantization, and severe computation overhead during selection may hinder the application of QAT. Therefore, a coreset selection method specifically designed for QAT is required.

In this work, we propose the coreset selection specifically for QAT and find that we can elevate the data efficiency and training robustness. We start by analyzing the impact of removing one specific sample from the training set during QAT and identifying that the error vector score (EVS) is a good theoretical approximation

of the importance of each sample. Based on the common practice of utilizing knowledge distillation during QAT, we also propose the disagreement score (DS) measuring the prediction gap between the quantized model and full-precision models. Based on these metrics, we propose a fully quantization-aware adaptive coreset selection (ACS) method to select training samples that fit the current training objective. An overview of the proposed method is shown in Fig. 2.

We demonstrate the superiority of our ACS in terms of effectiveness and efficiency on different networks (ResNet-18, MobileNet-V2, RetinaNet), datasets (CIFAR-10, CIFAR-100, ImageNet-1K, COCO), and quantization settings. For 2-bit weights-only quantization of MobileNetV2 on the CIFAR-100 dataset, QAT based on our ACS can achieve a mean accuracy of 67.19% with only 50% training data used for training per epoch. For 4-bit quantization of ResNet-18 on the ImageNet-1K dataset, our ACS can achieve top-1 accuracy of 68.39% compared to the 64.15% of random selection when only 10% training data is selected for the training of every epoch. In summary, our contribution can be summarized as follows:

- We are **the first** to investigate the data efficiency and robustness of quantization-aware training and empirically observe that the importance of different data samples varies during the training process.
- We propose two metrics: error vector score and disagreement score, to quantify the importance of each sample based on theoretical analysis of loss gradient.
- We propose a **quantization-aware** **A**daptive **C**oreset **S**election (**ACS**) method, which adaptively selects informative samples that fit current training epochs and prune redundant or noisy samples.
- We verify ACS on different network architectures, datasets, and quantization settings. Compared with previous methods, ACS can significantly improve both the data efficiency and robustness of quantization-aware training shown in Fig. 1.

2 Related Work

Quantization Quantization methods are powerful tools for improving the efficiency of model inference. The core insight is replacing full-precision weights and activations with lower-precision representation. The quantization methods can be classified as quantization-aware training (QAT) (Zhou et al., 2016; Esser et al., 2020; Bhalgat et al., 2020; Huang et al., 2022; Liu et al., 2023c) and post-training quantization (PTQ) (Nagel et al., 2020; Fang et al., 2020; Wang et al., 2020; Xiao et al., 2023; Liu et al., 2023b; Lee et al., 2023) based on whether to retrain a model with quantized weights and activations or start with a pre-trained model and directly quantize it without extensive training. Based on the characteristics of quantization intervals, these methods can be categorized into uniform and non-uniform quantization. While uniform quantization (Zhou et al., 2016; Choi et al., 2018; Esser et al., 2020) with uniform interval are more hardware-friendly and efficient, non-uniform quantization (Miyashita et al., 2016; Zhang et al., 2018; Li et al., 2019b; Yvinec et al., 2023), due to the flexibility of representation, can minimize the quantization error and achieve better performance than uniform schemes. While previous QAT methods assume an ideal case for unlimited computation resources, we research a more practical case to conduct QAT with a fixed computation budget in this work.

Learning from Noisy Labels The *noisy labels* are defined as unreliable labels that are corrupted from ground truth. The noise can stem from non-expert labeling or malicious label-flipping attack (Xiao et al., 2012). The estimated ratio of noisy labels in real-world datasets ranges from 8.0%-38.5% according to Song et al. (2022). To improve the robustness of learning for noisy labels, the solutions include designing robust architectures that are attuned to label noise (Yao et al., 2018; Lee et al., 2019), label correction (Tu et al., 2023; Li et al., 2023; 2024), robust regularization (Wei et al., 2021), and robust loss design (Huang et al., 2023). One important category that enables robust noise-tolerant training is sample selection (Paul et al., 2021; Xia et al., 2023a; Park et al., 2024). While the effectiveness of sample selection methods has been verified on full-precision deep learning, their performance on QAT is under-explored.

Coreset Selection Coreset selection targets improving the data efficiency by identifying the informative training samples. Previous works can be mainly classified into the following categories: **Geometry-based Methods:** These methods assume that samples close to each other in the feature space similarly influence the training. These redundant data points should be removed to improve efficiency. Some representative

works include Contextual diversity (CD) (Agarwal et al., 2020), k-Center-Greedy (Sener & Savarese, 2018), Sorscher et al. (2022), Moderate Coreset (Xia et al., 2023b), and Herding (Welling, 2009). **Decision Boundary-based Methods:** These methods select samples distributed near the decision boundary. These samples are difficult for the model to separate. Representative works include Adversarial Deepfool (Ducoffe & Precioso, 2018) and Contrastive Active Learning (CAL) (Margatina et al., 2021). **Gradient/Error-based Methods:** These methods assume the samples are more informative if they contribute more to the error or loss during the training. This category includes EL2N (Paul et al., 2021), CRAIG (Mirzasoleiman et al., 2020), GradMatch (Killamsetty et al., 2021a), Forgetting events (Toneva et al., 2019), and AdaCore (Pooladzandi et al., 2022). **Optimization-based Methods:** These methods formulate the coreset selection as a bilevel optimization problem. The outer level objective is the selection of samples and the inner level objective is the optimization of model parameters. The representative works include Borsos et al. (2020), Glister (Killamsetty et al., 2021b), Zhou et al. (2022), and Retrieve (Killamsetty et al., 2021c).

These methods perform well in some scenarios of full-precision training(specific subset fraction, specific outlier distribution, etc.). However, **none are verified on quantization settings or consider the requirements for QAT**. As shown in Fig. 1, the majority of these methods cannot even outperform random sampling in the QAT settings. Moreover, these methods either require early training of the target model for several epochs (Toneva et al., 2019; Paul et al., 2021) or time-consuming search (Sener & Savarese, 2018) and optimization (Killamsetty et al., 2021b;c), which leads to heavy computation overhead during QAT.

3 Importance of Each Sample in QAT

One underlying assumption of data scaling law is that all samples in the training set are equally important. However, many previous coreset research Sorscher et al. (2022) have proved that information within different samples is different for full-precision training, and the quantization setting has not been verified yet. In this section, we will first introduce quantization-aware training (QAT) and derive the gradient under cross-entropy and SGD in Sec. 3.1. We then analyze the change of gradient when a specific sample is removed from a training batch to investigate the importance of each training sample in Sec. 3.2. We propose an approximation of this gradient change considering the prediction error without introducing any memory overhead. We further prove that less training data is required when knowledge distillation is applied to QAT in Sec. 3.3. Another metric quantifying sample importance based on the prediction gap between the quantized student and full-precision teacher model is also introduced in Sec. 3.3.

3.1 Preliminaries of QAT

During QAT, the real-value data v^r is converted to b-bit quantized representation $v^q = q_b(v^r)$ by the quantizer q_b. Given the scale factor s of the quantizer, the number of positive quantization levels Q_P, and the number of negative quantization levels Q_N, we can have the quantizer q_b as

$$v^q = q_b(v^r) = s \times \lfloor \text{clip}(v^r/s, -Q_N, Q_P) \rceil, \tag{1}$$

where $\lfloor \cdot \rceil$ is the rounding function that rounds the input to the nearest integer, $\text{clip}(v, r_{\text{low}}, r_{\text{high}})$ return v with all value below r_{low} set to be r_{low} and all values above r_{high} set to be r_{high}. For the unsigned quantization, $Q_N = 0, Q_P = 2^b - 1$. While for the quantization of signed data, $Q_N = 2^{b-1}, Q_P = 2^{b-1} - 1$. To solve the problem that the gradient cannot back-propagate in Equation 1 during QAT, the straight-through estimator (STE) (Bengio et al., 2013) is utilized to approximate the gradient. In the back-propagation of QAT with STE, the gradient of the loss $\mathcal{L}$ with respect to the real-value data v^r is set to be

$$\frac{\partial \mathcal{L}}{\partial v^r} = \frac{\partial \mathcal{L}}{\partial v^q} \cdot \mathbf{1}_{-Q_N \leq v^r/s \leq Q_P}, \tag{2}$$

where $\mathbf{1}$ is the indicator function that outputs 1 within the quantization limit and 0 otherwise. Note that as we will apply clipping in the beginning of QAT, the gradient of quantized data can be seen as approximately equal as the gradient of real-value data.

The training set of QAT is denoted as $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^N$, where input $x_i \in \mathbb{R}^d$ are vectors of length d and $y \in \{0, 1\}^M$ are one-hot vectors encoding labels. The neural network to be trained is denoted as $f(w^r, x)$,

where real-value weights are w^r. We use cross-entropy loss $\mathcal{L}(\hat{p}, y) = \sum_{m=1}^{M} y^{(m)} \log p^{(m)}$ in the QAT and $p(w^q, x)$ is a probability vector denoting the output of the quantized neural network $f(w^q, x)$. Stochastic gradient descent (SGD) is used for the optimization. Suppose the real-value weights at iteration t are w_t^r and the batch of input samples is $\mathcal{T}_{t-1} \subseteq \mathcal{T}$, the weights are updated following

$$w_t^r = w_{t-1}^r - \eta \sum_{(x,y)\in\mathcal{T}_{t-1}} g_{t-1}(x, y), \tag{3}$$

where η denotes the learning rate and $g_{t-1}(x, y) = \nabla\mathcal{L}_{t-1}(p(w_{t-1}^r, x), y)$ is the gradient of cross entropy loss.

3.2 Error Vector Score

To look into the importance of each sample $\{(x_i, y_i)\}$, we measure the difference of the expected magnitude of the loss vector on the training set $\mathcal{T}$ and another training set which only removes one specific sample $\mathcal{T}' = \mathcal{T} \setminus \{(x_i, y_i)\}$. For simplicity, we approximate all the training dynamics in continuous time. Based on the chain rule and STE of quantization, we have the change of loss $\mathcal{L}$ at time t on sample (x, y) of batch $\mathcal{T}$ as

$$\frac{d\mathcal{L}}{dt}\bigg|_{(x,y),\mathcal{T}_t} = g_t(x, y)\frac{dw_t^q}{dt} = g_t(x, y)\frac{dw_t^r}{dt}. \tag{4}$$

According to the discrete-time dynamic of real-value weights in Eq. 3, we have $\frac{dw_t^q}{dt} \approx w_t^r - w_{t-1}^r = -\eta\sum_{(x,y)\in\mathcal{T}_{t-1}} g_{t-1}(x, y)$. To measure the contribution of a specific sample (x_i, y_i), we measure the change of loss with and without the sample. For a given data batch $\mathcal{T}$, if the sample $(x_i, y_i) \notin \mathcal{T}$, we can ignore the change in $\frac{d\mathcal{L}}{dt}\big|_{(x,y),\mathcal{T}_t}$. For any sample $(x_j, y_j) \in \mathcal{T}, j \neq i$ in the same batch, the importance $\mathcal{I}(x_i, y_i)$ is measured as

$$\mathcal{I}(x_i, y_i) = \left\| \frac{d\mathcal{L}}{dt}\bigg|_{(x_j,y_j),\mathcal{T}} - \frac{d\mathcal{L}}{dt}\bigg|_{(x_j,y_j),\mathcal{T}'} \right\|. \tag{5}$$

According to the chain rule, we have

$$\frac{d\mathcal{L}}{dt}\bigg|_{(x_j,y_j),\mathcal{T}} = \frac{d\mathcal{L}(p(w_t^q, x_j), y_j)}{dw_t^q}\frac{dw_t^q}{dw_t^r}\frac{dw_t^r}{dt} \tag{6}$$

According to the characteristics of STE in Eq. 2, $\frac{dw_t^q}{dw_t^r} = 1$ holds for all input within the clipping range. Following the updating rule of weights in Eq. 3, the gradient $\frac{dw_t^r}{dt} = -\eta\sum_{(x^*,y^*)\in\mathcal{T}} g_{t-1}(x^*, y^*)$. The only difference of the training set $\mathcal{T}$ and $\mathcal{T}'$ is the existence of sample (x_i, y_i). Thus, we have

$$\begin{aligned}\mathcal{I}(x_i, y_i) &= \left\| \frac{d\mathcal{L}}{dw_t^q}\Big(\frac{dw_t^r}{dt}\bigg|_{(x_j,y_j),\mathcal{T}} - \frac{dw_t^r}{dt}\bigg|_{(x_j,y_j),\mathcal{T}'} \Big) \right\| \\ &= \eta \left\| \frac{d\mathcal{L}}{dw_t^q} \cdot g_{t-1}(x_i, y_i) \right\|.\end{aligned} \tag{7}$$

We use $\frac{d\mathcal{L}}{dw_t^q}$ as a simplification of $\frac{d\mathcal{L}(p(w_t^q,x_j),y_j)}{dw_t^q}$, which is only dependant on the current training sample (x_j, y_j) but not dependant on the sample (x_i, y_i) that is removed from batch $\mathcal{T}$. Since learning rate η and the gradient of loss w.r.t. quantized weights $\frac{d\mathcal{L}}{dw_t^q}$ can be seen as constant given $\mathcal{T}$, the importance of the sample (x_i, y_i) in this batch $\mathcal{T}$ is only related to the gradient norm of cross-entropy loss of this sample $||g_{t-1}(x_i, y_i)||$. The examples with a larger gradient norm expectation have more influence on the supervised training of other data, which means they are important for QAT and should be included in the coreset. We can select data with high importance by sorting by the norm of gradient, which is also covered in previous works Paul et al. (2021). However, storing the loss gradient for comparison requires extra memory and is hard to transfer between network architectures. We approximate the norm gradient with the norm of the error vector, which is defined as follows.

Definition 1 (Error Vector Score) *The error vector score of a training sample (x, y) at iteration t id defined to be $d_{EVS} = \|p(w_t^q, x) - y\|_2$.*

For any input $x \in \mathbb{R}^d$, gradient norm $||g_t(x, y)||$ is a non-random score. We take its expectation over random minibatch sequence and random initialization to get the expected gradient norm $\mathbb{E}||g_t(x, y)||$, which can also be indicated as

$$\mathbb{E}\,\|g_t(x, y)\| = \sum_{m=1}^{M} \frac{d\mathcal{L}_t(p(w_t^q, x), y)^T}{df_t^{(m)}} \frac{df_t^{(m)}}{dw_t^q}, \tag{8}$$

where $\frac{df_t^{(m)}}{dw_t^q}$ denotes the gradient of m-th logit on weights. Since we are using cross-entropy as the loss function, the gradient of loss $\mathcal{L}$ on the m-th logit output $f_t^{(m)}$ follows $\frac{d\mathcal{L}_t(p(w_t^q,x),y)^T}{df_t^{(m)}} = p(w_t^q, x)^{(m)} - y^{(m)}$. Previous works (Fort et al., 2020; Fort & Ganguli, 2019) empirically observe that $\frac{df_t^{(m)}}{dw_t^q}$ are similar across different logits $m \in M$ and training sample (x, y). Thus, selecting samples based on the error vector score d_{EVS} will be more likely to identify the samples with a larger gradient norm. Different from the previous method (Paul et al., 2021) also leveraging the error metrics, no early training is required, and we only use the current quantized model prediction $p(w_t^q, x)$ during QAT.

3.3 Disagreement Score and Knowledge Distillation

Intrinsically, a quantized classification network should learn an ideal similar mapping f from input sample x to the output logits $f(w, x)$ as a full-precision network, and the gap between the quantized prediction $p(w^q, x)$ of student model and real-value prediction $p_{\mathbf{T}}(w^r, x)$ of teacher model $\mathbf{T}$ needs to be minimized. Based on this insight, knowledge distillation (KD) is widely used during QAT with a full-precision model as the teacher, which can also be seen in previous works (Polino et al.; Huang et al., 2022; Mishra & Marr, 2018; Liu et al., 2023a). The loss function is designed to enforce the similarity between the output of the full-precision teacher and the quantized student model as

$$\mathcal{L}_{KD} = -\frac{1}{N} \sum_{m}^{M} \sum_{i=1}^{N} p_{\mathbf{T}}^{(m)}(w^r, x_i) \log(p^{(m)}(w^q, x_i)) \tag{9}$$

where the KD loss is defined as the cross-entropy between the output distributions $p_{\mathbf{T}}$ of a full-precision teacher and a quantized student on the same input x but different weights representation w^r and w^q. x_i is one of the input samples from the training set. m and N denote the classes and total training sample numbers, respectively.

Note that this process can be regarded as the distribution calibration for the student network and one-hot label is not involved during QAT. Since the loss function used for knowledge distillation is still cross-entropy, and we still assume we use SGD for the optimization, most conclusions in Sec. 3.2 still hold by replacing the one-hot ground truth label y with full-precision teacher's prediction $p_{\mathbf{T}}(w_t^r, x)$. Thus, we propose the disagreement score as follows.

Definition 2 (Disagreement Score) *The disagreement score of a training sample (x, y) at iteration t is defined to be $d_{DS} = \|p(w_t^q, x) - p_{\mathbf{T}}(w_t^r, x)\|_2$.*

The core difference between error vector score d_{EVS} and disagreement score d_{DS} is the target label. While d_{EVS} uses one-hot hard labels, the d_{DS} uses the distilled soft labels. We empirically notice that the data needed for the training is reduced when knowledge distillation is applied, which is helpful for our selection with a small data fraction. We further demonstrate the advantage in terms of training data requirements using the soft label based on Vapnik–Chervonenkis theory (Vapnik, 1999), which decomposes the classification error $R(f_s)$ of a classifier $f_s \in \mathcal{F}_s$ as

$$R(f_s) - R(f) \leq O\left(\frac{|\mathcal{F}_s|_{\text{C}}}{n^{\alpha_s}}\right) + \varepsilon_s, \tag{10}$$

where $O(\cdot)$ denotes the asymptotic approximation and ε_s is the approximation error of $\mathcal{F}_s$ with respect to $\mathcal{F}$. $f \in \mathcal{F}$ denotes the real target function. $|\cdot|_{\text{C}}$ is the VC-Dimension of the function class measuring its capacity.

n is the number of total training data. $\frac{1}{2} \leq \alpha_s \leq 1$ is an indicator measuring the difficulty of the problems. For non-separable and difficult problems, $\alpha_s = \frac{1}{2}$, which means the classifier learns at a slow rate of $O(n^{-\frac{1}{2}})$. For separable and easy problems, $\alpha_s = 1$, indicating the classifier learns at a fast rate. In our setting, if the quantized model $f_q \in \mathcal{F}_q$ directly learns from the hard labels, the difficulty of the problem is high, and we assume $\alpha_q = \frac{1}{2}$, we have

$$R(f_q) - R(f) \leq O\left(\frac{|\mathcal{F}_q|_{\mathrm{C}}}{\sqrt{n}}\right) + \varepsilon_q, \tag{11}$$

where ε_q is the approximation error of the quantized model. However, if we first train the full-precision teacher model $f_r \in \mathcal{F}_r$ and then utilize knowledge distillation to learn the representation from the teacher, the difficulty of the learning becomes easier, assuming $\alpha_r = 1$, we have

$$R(f_r) - R(f) \leq O\left(\frac{|\mathcal{F}_r|_{\mathrm{C}}}{n}\right) + \varepsilon_r, \quad R(f_q) - R(f_r) \leq O\left(\frac{|\mathcal{F}_q|_{\mathrm{C}}}{n^{\alpha_{qr}}}\right) + \varepsilon_{qr}, \tag{12}$$

where the $\varepsilon_r, \varepsilon_{qr}$ denotes the approximation error of the $\mathcal{F}_r$ with respect to $\mathcal{F}$ and approximation error of the $\mathcal{F}_q$ with respect to $\mathcal{F}_r$, respectively. Compared to the direct learning quantized model from the hard label shown in Eq. 11, the knowledge distillation with real-value teacher f_r yields the classification error as follows:

$$R(f_q) - R(f) \leq O\left(\frac{|\mathcal{F}_r|_{\mathrm{C}}}{n}\right) + \varepsilon_r + O\left(\frac{|\mathcal{F}_q|_{\mathrm{C}}}{n^{\alpha_{qr}}}\right) + \varepsilon_{qr} \leq O\left(\frac{|\mathcal{F}_q|_{\mathrm{C}} + |\mathcal{F}_r|_{\mathrm{C}}}{n^{\alpha_{qr}}}\right) + \varepsilon_r + \varepsilon_{qr}. \tag{13}$$

Following previous studies on knowledge distillation (Lopez-Paz et al., 2015; Mirzadeh et al., 2020), the soft labels contain more information than hard labels for each sample. Thus, we have $\varepsilon_r + \varepsilon_{qr} \leq \varepsilon_q$ and $O\left(\frac{|\mathcal{F}_q|_{\mathrm{C}}+|\mathcal{F}_r|_{\mathrm{C}}}{n^{\alpha_{qr}}}\right) \leq O\left(\frac{|\mathcal{F}_q|_{\mathrm{C}}}{\sqrt{n}}\right)$. Combining these two inequalities, we have the inequality

$$O\left(\frac{|\mathcal{F}_q|_{\mathrm{C}} + |\mathcal{F}_r|_{\mathrm{C}}}{n^{\alpha_{qr}}}\right) + \varepsilon_r + \varepsilon_{qr} \leq O\left(\frac{|\mathcal{F}_q|_{\mathrm{C}}}{\sqrt{n}}\right) + \varepsilon_q, \tag{14}$$

which means when the number of training samples n is the same, the upper bound of classification error based on the soft label is lower. When we want to achieve the same upper bound of classification error $R(f_q) - R(f)$ using these two techniques, learning from soft labels requires less data. This is the core reason why we use knowledge distillation and disagreement score d_{DS} to select the coreset.

4 Adaptive Coreset Selection for QAT

In Sec. 3, we propose to use d_{EVS} and d_{DS} to select the coreset for QAT. While d_{DS} could help select those samples that produce large performance gaps between quantized and full-precision models, d_{EVS} targets more at the error of quantized prediction. These two metrics cover different characteristics of training data, and we need both to improve the diversity of our coreset. For different stages of QAT, different metrics should be considered when selecting samples that fit the current training objective. Previous research (Kim et al., 2019) has shown that QAT should start with the hard label to help a better initialization for the quantized model and then use soft labels to guide it to better local minima. In light of this, we propose Adaptive Coreset Selection (ACS) for QAT to select the important samples considering current training epoch t, d_{EVS}, and d_{DS} adaptively.

For the given current training epoch t and the total training epoch E, we propose a cosine annealing weights coefficient $\beta(t) = \cos(\frac{t}{2E}\pi)$ to consider two metrics simultaneously and balance between them. The final selection metric is a linear combination of d_{EVS} and d_{DS} as follows:

$$d_{\mathrm{ACS}}(t) = \beta(t) d_{\mathrm{EVS}}(t) + (1 - \beta(t)) d_{\mathrm{DS}}(t). \tag{15}$$

As we have $\beta(0) = 1$ and $\beta(E) = 0$ in the early stage, the selection is mainly based on d_{EVS}. When the quantized model converges, we focus more on the d_{DS} in later epochs. We perform coreset selection every R epoch, where R is determined before training. The pseudo-code for our ACS algorithm is shown in Alg. 1.

There are two major advantages of our quantization-aware ACS method. The first lies in the adaptation to the training phase when knowledge distillation is applied. As soft labels retain more information about the

Algorithm 1 Adaptive Coreset Selection for QAT

Input: Training dataset $T = \{(x_i, y_i)\}_{i=1}^{n}$, Real-value network with weights $\mathbf{W^r}$, Coreset data fraction per epoch S, Total training epochs E, Selection interval R, Initial coreset $T_{\mathrm{ACS}}(t) = \emptyset$
Output: Quantized network with weights $\mathbf{W^q}$
Initialize quantized weights $\mathbf{W^q}$ from $\mathbf{W^r}$ following Eq. 1
for $t \in [0, ..., E-1]$ **do**
  if $t\%R == 0$ **then**
    $\beta(t) = \cos(\frac{t}{2E}\pi)$
    for $(x_i, y_i) \in T$ **do**
      $d_{\mathrm{EVS}}(x_i, t) = \|p(\mathbf{W_t^q}, \mathbf{x_i}) - \mathbf{y_i}\|_\mathbf{2}$
      $d_{\mathrm{DS}}(x_i, t) = \|p(\mathbf{W_t^q}, \mathbf{x_i}) - \mathbf{p_T}(\mathbf{W_t^r}, \mathbf{x_i})\|_\mathbf{2}$
      $d_{\mathrm{ACS}}(x_i, t) = \beta(t) d_{\mathrm{EVS}}(x_i, t) + (1 - \beta(t)) d_{\mathrm{DS}}(x_i, t)$
    end for
    Sort $d_{\mathrm{ACS}}(x_i, t)$, select top $S\%$ samples to replace $T_{\mathrm{ACS}}(t)$
  else
    $T_{\mathrm{ACS}}(t) \leftarrow T_{\mathrm{ACS}}(t-1)$
  end if
  Train $\mathbf{W^q}$ on $T_{\mathrm{ACS}}(t)$ following Eq. 9
end for

target than hard labels, we should encourage the quantized student model to learn sequentially on hard labels first and soft labels then. This implicit learning hierarchy is observed in QKD (Kim et al., 2019) and is named "self-learning" and "tutoring". With the proposed ACS fully aware of this hierarchy, the selected coreset helps stabilize the training and guarantee faster convergence. The second advantage is the diversity of training data. More samples could be covered in the coreset of different epochs, and the coverage of the original full dataset contributes to the convergence of a more robust model. Note that only when the **optimal data sequence** and **high training sample diversity** are achieved simultaneously, the performance of QAT will be significantly better. We demonstrate in the Appendix. E that even when all data are covered but the order is random, the accuracy of our quantized model will be negatively influenced.

5 Experiments

Datasets and networks The efficiency experiments are conducted on CIFAR-100 (Krizhevsky et al., 2009) and ImageNet-1K dataset (Deng et al., 2009). We evaluate MobileNetV2 (Howard et al., 2017) on CIFAR-100 and evaluate ResNet-18 (He et al., 2016) on the ImageNet-1K dataset. The *width multiplier* is set to be 0.5 for MobileNetV2. We further provide experimental results of quantized RetinaNet Lin et al. (2017) on MS COCO object detection benchmark (Lin et al., 2014). The robustness experiments are conducted following the setting of Paul et al. (2021).

Baselines We choose multiple coreset selection methods from different categories as our baseline. The selected methods include: Random Sampling, EL2N-Score (Paul et al., 2021)[1], Forgetting (Toneva et al., 2019), Glister (Killamsetty et al., 2021b), kCenterGreedy (Sener & Savarese, 2018), Contextual Diversity (CD) (Agarwal et al., 2020), Moderate Coreset (Xia et al., 2023b). For methods involving early training, we set training epochs as 5. Note that comparing adaptive-based and non-adaptive-based methods is fair and the common practice of previous research (Pooladzandi et al., 2022; Killamsetty et al., 2021c).

Training and selection details For MobileNetV2, we train the network for 200 epochs using a learning rate of 0.01, weight decay of 5e-4, batch size of 512, $R = 20$, and SGD optimizer. For ResNet-18 on ImageNet-1K, we train the network for 120 epochs using a learning rate of 1.25e-3, no weight decay, batch

[1]Note that we only verify the EL2N-score instead of Gradient norm (GradN) score in Paul et al. (2021) based on the TMLR reproduction from Kirsch (2023) observing that the inconsistency of GraNd at initialization will results in a sub-optimal performance even compared with random selection.

size of 512, $R = 10$, and Adam optimizer. We use the quantization method and full-precision model following LSQ+ (Bhalgat et al., 2020). All the experiments were carried out on 2 NVIDIA RTX 3090 GPUs. As we notice that the results on the ImageNet-1K dataset will not vary significantly across different runs, experiments of each settings are only perform once. For CIFAR-100 experiments, each experiment of different settings is repeated 5 times. For a fair comparison, we use knowledge distillation with the corresponding full-precision model as the teacher in **all** experiments, regardless of the method and dataset fraction.

Table 1: Comparison of applying different coreset selection methods to 2/32-bit quantized MobileNetV2 on CIFAR-100 dataset with various subset fractions. When full data are selected ($\mathcal{S} = 100\%$), the mean accuracy and standard deviation is 68.1±0.9%. The first and second best accuracy is shown in **bold** and underlined, and the performance improvement over the previous SOTA is marked in blue.

Method/Fraction	$\mathcal{S} = 10\%$	$\mathcal{S} = 20\%$	$\mathcal{S} = 30\%$	$\mathcal{S} = 40\%$	$\mathcal{S} = 50\%$
Random	62.3±0.9	64.1±0.7	65.2±0.3	65.6±0.6	66.2±0.9
EL2N Score (Paul et al., 2021)	63.0±0.6	64.0±1.6	65.6±0.8	64.6±0.8	65.3±0.4
Forgetting (Toneva et al., 2019)	60.7±0.3	63.1±0.6	65.0±0.8	65.3±0.8	66.0±1.2
Glister (Killamsetty et al., 2021b)	56.5±1.2	60.4±1.0	61.3±0.6	63.0±1.0	64.8±0.9
kCenterGreedy (Sener & Savarese, 2018)	60.2±0.8	62.7±0.5	63.8±0.7	64.8±0.6	66.3±1.1
CD (Agarwal et al., 2020)	60.3±0.8	62.6±0.9	64.0±1.0	64.8±0.4	65.3±0.4
Moderate (Xia et al., 2023b)	57.9±0.3	60.4±0.7	62.8±0.6	63.6±0.8	64.5±1.6
Ours	**63.7±0.8** (↑0.7)	**65.9±0.7** (↑1.8)	**66.4±0.8** (↑0.8)	**66.9±0.5** (↑1.3)	**67.2±0.5**(↑0.9)

Table 2: Comparison of different methods of 4/4-bit quantized ResNet-18 on ImageNet-1K. When full data are selected ($\mathcal{S} = 100\%$), the accuracy is 72.46%. The first and second best accuracy is shown in **bold** and underlined, and the performance improvement over the previous SOTA is marked in blue.

Method/Fraction (%)	$\mathcal{S} = 10\%$	$\mathcal{S} = 30\%$	$\mathcal{S} = 50\%$	$\mathcal{S} = 60\%$	$\mathcal{S} = 70\%$	$\mathcal{S} = 80\%$
Random	64.15	68.53	70.49	70.94	71.06	71.96
EL2N Score (Paul et al., 2021)	61.71	67.31	70.14	70.89	71.54	71.88
Forgetting (Toneva et al., 2019)	63.09	67.77	70.14	71.00	71.36	71.82
Glister (Killamsetty et al., 2021b)	63.25	68.94	70.92	71.39	71.93	72.22
kCenterGreedy (Sener & Savarese, 2018)	62.98	68.56	70.35	71.13	71.59	71.96
CD (Agarwal et al., 2020)	63.22	68.74	70.90	71.27	71.78	72.11
Moderate (Xia et al., 2023b)	62.39	68.06	70.43	70.62	71.56	71.99
Ours	**68.39** (↑4.24)	**71.09** (↑2.15)	**71.59** (↑0.67)	**72.00** (↑0.61)	**72.19** (↑0.26)	**72.31** (↑0.09)

5.1 Benchmarking Previous Coreset Method

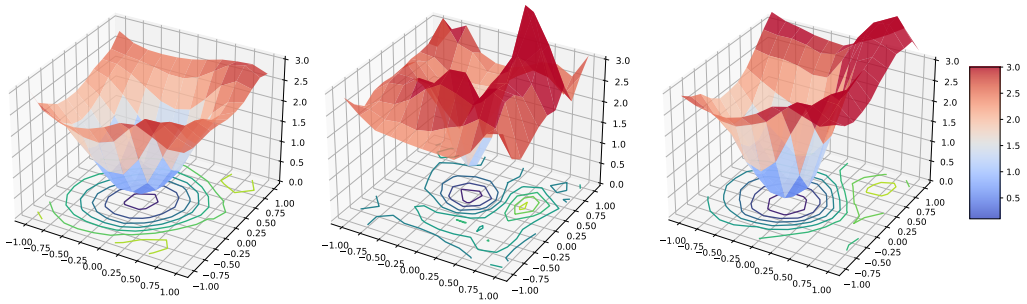
The comparison of the QAT Top-1 accuracy of MobileNetV2 on CIFAR-100 and ResNet-18 on ImageNet-1K is shown in Tab. 1 and Tab. 2. We note that most previous methods cannot exceed random selection in our QAT setting, and the few surpass the baseline only on specific data fractions. This trend is also discussed for full-precision training by Kirsch (2023). For example, kCenterGreedy (Sener & Savarese, 2018) shows satisfying performance when the subset size is large (50% for CIFAR-10, 70%/80% for ImageNet-1K) but fails to demonstrate effectiveness on small coreset size. Our method outperforms state-of-the-art methods on all subset fractions $\mathcal{S}$ by a great margin. Specifically, the ResNet-18 accuracy of 10% subset on ImageNet-1K using our method is 68.39%, achieving an absolute gain of 4.24% compared to the baseline method.

Efficiency analysis The detailed QAT training time of ResNet-18 on ImageNet-1K coreset and 2 NVIDIA RTX 3090 GPUs with different methods is listed in Tab. 3. When full data are selected ($\mathcal{S} = 100\%$), the training time is 62.3h. The only method with comparable efficiency to ours is the Moderate Coreset (Xia et al., 2023b), which only needs to perform forwarding on samples and can eliminate optimization or greedy searching. The results prove that our method can effectively reduce training time without incurring computation overhead by the selection algorithm.

Effect of adaptive epochs The choice of selection interval R is vital in our algorithm, as too large R will fail to help adaption and improve data diversity, and too small R will introduce too much computation

Table 3: Training time (hours) of 4/4-bit quantized ResNet-18 on ImageNet-1K with various subset fraction.

Method/Fraction (%)	10%	30%	50%	60%	70%	80%
EL2N Score	12.7	23.8	39.1	44.1	49.9	56.0
Forgetting	12.7	23.9	39.5	44.7	50.2	56.6
Glister	13.8	23.0	39.9	46.6	58.0	66.8
kCenterGreedy	13.0	25.2	38.4	43.8	50.5	56.9
CD	11.9	24.5	37.1	42.5	48.9	55.0
Moderate	11.3	22.3	36.2	42.5	48.2	54.1
Ours (R=10)	**11.2**	**20.7**	**36.1**	**42.2**	**48.0**	**53.7**



(a) Full Dataset (b) Random $\mathcal{S} = 10\%$ (c) Ours $\mathcal{S} = 10\%$

Figure 3: Visualization of the loss landscape (Li et al., 2018) of 2-bit quantized MobileNetV2 trained on the CIFAR-100 (a) full dataset, (b) 10% random subset, and (c) 10% ACS coreset.

overhead and undermine the efficiency. We apply grid search on R and empirically prove that $R = 10$ is optimal for our ImageNet-1K training. The accuracy and training time results are shown in Tab. 4. We can observe from the results that $R = 10$ achieves a similar performance as $R = 5$ with a shorter training time. As no back-propagation of the gradient is involved during the computation of d_{EVS} and d_{DS}, the computation overhead is acceptable under most settings with $R > 5$ compared with previous methods involving training.

Table 4: Analysis on adaptive epochs R.

Fraction	$\mathcal{S}$=10%		$\mathcal{S}$=30%		$\mathcal{S}$=50%		$\mathcal{S}$=70%	
	Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time
$R = 5$	68.8	12.7h	71.05	21.5h	71.47	37.0h	72.15	49.1h
$R = 10$	68.39	11.3h	71.09	20.7h	71.59	36.1h	72.19	48.0h
$R = 20$	67.58	11.0h	70.35	20.2h	71.42	35.5h	71.97	47.6h
$R = 40$	66.10	10.7h	69.86	19.8h	71.25	34.9h	72.00	47.2h
$R = 60$	64.96	10.5h	69.27	19.5h	71.05	34.4h	71.93	46.9h
$R > 120$	62.82	10.3h	67.62	19.2h	69.98	33.9h	71.37	46.6h

Table 5: Analysis on strategy $\beta(t)$.

Fraction (%)	10%	30%	50%	70%
fixed	67.95	70.83	71.21	72.01
linear	68.35	71.07	71.39	72.11
sqrt	68.15	71.03	71.44	72.15
quadratic	68.11	70.95	71.35	72.10
d_{EVS} only	68.06	70.63	71.41	71.96
d_{DS} only	67.07	70.24	71.43	72.08
cosine	**68.39**	**71.09**	**71.59**	**72.19**

Ablation study and effect of annealing strategy As we propose two metrics for coreset selection: d_{EVS} and d_{DS}, it is important to analyze how we should balance between them and the contribution of both metrics. We try the following different annealing strategies and settings: (1) fixed ($\beta(t) = 0.5$); (2) linear ($\beta(t) = 1 - \frac{t}{E}$); (3) square root ($\beta(t) = 1 - \sqrt{\frac{t}{E}}$); (4) quadratic ($\beta(t) = 1 - (\frac{t}{E})^2$); (5) cosine ($\beta(t) = \cos(\frac{t}{2E}\pi)$); (6) d_{EVS} only ($\beta(t) = 1$); (7) d_{DS} only ($\beta(t) = 0$); The results of coreset selection of 4-bit quantized ResNet-18 on ImageNet-1K are listed in Tab. 5. The performance gap with different annealing strategies is close as they all follow the trends to use d_{EVS} in the early epochs of training and d_{DS} in later epochs. Among all these annealing strategies, cosine annealing is slightly better. When only one metric is used for the selection, the

performance will drop. We also notice that d_{EVS} and d_{DS} are complementary as d_{EVS} works well with small data fraction and d_{DS} has better performance with large data fraction.

5.2 Training with Label Noise

We would like to highlight another application of our method: identifying noisy labels in training data and improving the robustness of the QAT. The nature of advantage of our method is to select important training examples for QAT and remove those low-quality redundant examples. This is especially useful when there is label noise in the training set (some examples are not correctly labeled). This application is proposed in previous coreset research Mirzasoleiman et al. (2020), but not verified on QAT. We follow the setting of Mirzasoleiman et al. (2020) to experiment on the QAT of ResNet-18 on CIFAR-10 with 10% randomized labels (labels of 10% of training samples are randomly re-generated). The ResNet-18 is quantized to 2/32 for weights and activations. When the full dataset is selected, the accuracy is 89.91±0.51%. The comparison of the QAT Top-1 accuracy (%) of 2/32-bit quantized ResNet-18 on CIFAR-10 with 10% randomized labels is shown as follows in Tab. 6. Note that we select E2LN Score and GraNd Score for comparison as they are the few coreset selection solutions among the methods we investigate in Tab. 2 claimed with the capacity to remove noise in the training set and improve training robustness.

Table 6: Top-1 Accuracy of 2/32-bit quantized ResNet-18 on CIFAR-10 with 10% random label noise

Method/Fraction (%)	10%	20%	30%	40%	50%
Random	83.14±0.51	84.77±0.80	84.82±0.57	85.09±0.15	85.33±0.60
EL2N Score	85.80±0.51	86.02±0.55	87.18±0.35	87.51±0.29	88.01±0.45
GraNd Score	85.71±0.30	85.96±0.15	87.10±0.55	87.44±0.20	87.94±0.29
Ours	**88.31±0.27**	**89.53±0.62**	**89.95±0.39**	**90.21±0.39**	**90.23±0.18**

As can be seen from the results in Tab. 6, our method outperforms all other selection baselines and even performs better than full-data training when coreset size $\mathcal{S} \geq 30\%$. To quantitatively demonstrate the noisy sample pruning effectiveness, we report the average recall for this noisy-sample detection problem. Noticeably, while the other pruned samples are in the correct label, they are still mostly redundant and can be pruned without significant performance degradation in QAT. From the results of recall in Tab 7, our method states most of the noisy samples. These results show that our method can successfully prune those samples with incorrect labels. In this case, we actually achieve a **"lossless acceleration"** as both the accuracy and efficiency are better than full-data training.

Table 7: The average recall of noisy sample identification on CIFAR-10 with 10% random label noise.

Method/Fraction (%)	10%	20%	30%	40%	50%
EL2N Score	90.7%	80.5%	71.0%	60.2%	47.4%
GraNd Score	92.3%	81.7%	65.4%	41.0%	26.5%
Ours	97.9%	90.2%	84.1%	78.9%	73.5%

5.3 Object Detection Tasks

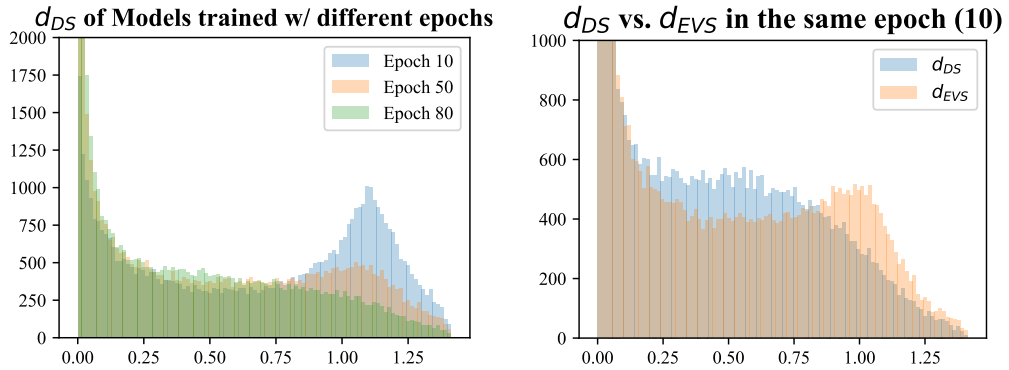
We further conduct experiments of 4/4-bit QAT of RetinaNet (Lin et al., 2017) with ResNet-18 backbone on the MS COCO object detection dataset (Lin et al., 2014). We use the same selection metric for the RetinaNet training and use the output for the classification head as the probability vector p. When there are multiple objects in one image, we use the mean selection metric d_{ACS} of the probability output of all objects. The QAT method used is FQN (Li et al., 2019a) and we perform training on `coco-2017-train` for 100 epochs. We investigate two coreset fractions: 10% and 50% and use R=10. The results of mAP are listed in Tab. 8. Our method outperforms the random selection baseline and state-of-the-art selection method Moderate (Xia et al., 2023b) significantly, which proves that our method is effective on object detection tasks.

Table 8: Comparision of performance with different methods of RetinaNet on COCO benchmarks.

Method	Fraction	AP	$AP^{0.5}$	$AP^{0.75}$	AP^{S}	AP^{M}	AP^{L}
Full	100%	28.6	46.9	29.9	14.9	31.2	38.7
Random	10%	21.4	39.8	22.4	7.5	24.3	27.4
EL2N Score	10%	20.7	37.0	19.9	8.4	25.0	28.1
Moderate	10%	22.0	37.8	20.4	8.4	25.0	28.1
Ours	10%	24.4	40.9	25.1	9.9	27.1	31.5
Random	50%	25.4	42.5	26.7	10.7	27.7	32.1
EL2N Score	50%	25.9	42.8	27.0	11.4	28.6	33.8
Moderate	50%	25.0	42.7	25.9	11.2	28.5	33.0
Ours	50%	26.7	44.0	27.8	12.1	30.0	35.1

5.4 Visualization and Analysis

We visualize the loss landscape (Li et al., 2018) of MobileNetV2 training on the full CIFAR-100 dataset, 10% random subset of CIFAR-100, and 10% coreset of CIFAR-100 based on our methods shown in Fig. 3. We can see from the results that QAT on coreset with our method has a more centralized and smoother loss compared to the baseline methods, which reflects that our method helps improve the training stability of QAT. We also visualize the distribution of disagreement scores d_{DS} and error vector score d_{EVS} in Fig. 4a and Fig. 4b. The setting is the same as the MobileNetV2 experiment listed in Tab. 1. We can see from the results that the mean of d_{DS} shifts to zero during the QAT, which proves that d_{DS} is a useful metric to quantify the importance of each sample. The distribution discrepancy between d_{EVS} and d_{DS} proves the necessity of considering both metrics to select diverse data into our coreset.



(a) d_{DS} distribution

(b) d_{DS} vs. d_{EVS}

Figure 4: (a) Distribution of disagreement scores d_{DS} on MobileNetV2 for different epochs. (b) Distribution of disagreement scores d_{DS} and error vector score d_{EVS} on MobileNetV2 in the same epoch.

6 Conclusion

This is the first work focusing on the data efficiency and robustness of quantization-aware training (QAT). By removing samples from the training batches and analyzing the loss gradient, we theoretically prove that the importance of each sample varies significantly for QAT. The error-vector score and disagreement score are proposed to quantify this importance. Considering the training characteristics of QAT, we propose a fully quantization-aware Adaptive Coreset Selection (ACS) method to better adapt to different training phases and improve training data diversity. Extensive experiments on various datasets, networks, and quantization settings further demonstrate the effectiveness of our method. We also verified the proposed ACS on the noisy label setting and proved that our method can improve the robustness of QAT.

References

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI*, pp. 137–153, 2020.

Ibrahim M Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35:22300–22312, 2022.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 696–697, 2020.

Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems*, 33:14879–14890, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. In *International Conference on Learning Representations*, 2020.

Jun Fang, Ali Shafiee, Hamzah Abdel-Aziz, David Thorsley, Georgios Georgiadis, and Joseph H Hassoun. Post-training piecewise linear quantization for deep neural networks. In *European Conference on Computer Vision*, pp. 69–86. Springer, 2020.

Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes. *arXiv preprint arXiv:1910.05929*, 2019.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:5850–5861, 2020.

Satoru Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

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*, pp. 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

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

Zhizhong Huang, Junping Zhang, and Hongming Shan. Twin contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11661–11670, 2023.

Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. *arXiv preprint arXiv:2006.15412*, 2020.

Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pp. 1–12, 2017.

Patrick Judd, Jorge Albericio, Tayler Hetherington, Tor M Aamodt, and Andreas Moshovos. Stripes: Bit-serial deep neural network computing. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1–12. IEEE, 2016.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021a.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118, 2021b.

Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:14488–14501, 2021c.

Jangho Kim, Yash Bhalgat, Jinwon Lee, Chirag Patel, and Nojun Kwak. Qkd: Quantization-aware knowledge distillation. *arXiv preprint arXiv:1911.12491*, 2019.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Andreas Kirsch. Does 'deep learning on a data diet'reproduce? overall yes, but grand at initialization does not. *Transactions on Machine Learning Research*, 2023.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Jung Hyun Lee, Jeonghoon Kim, Se Jung Kwon, and Dongsoo Lee. Flexround: Learnable rounding based on element-wise division for post-training quantization. In *International Conference on Machine Learning*, pp. 18913–18939. PMLR, 2023.

Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International conference on machine learning*, pp. 3763–3772. PMLR, 2019.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6391–6401, 2018.

Lin Li, Jun Xiao, Hanrong Shi, Hanwang Zhang, Yi Yang, Wei Liu, and Long Chen. Nicest: Noisy label correction and training for robust scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2810–2819, 2019a.

Yifan Li, Hu Han, Shiguang Shan, and Xilin Chen. Disc: Learning from noisy labels via dynamic instance-specific selection and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24070–24079, 2023.

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*, 2019b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

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

Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers. *arXiv preprint arXiv:2302.02210*, 2023a.

Shih-yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. Llm-fp4: 4-bit floating-point quantized transformers. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3296–3305, 2019.

Zechun Liu, Kwang-Ting Cheng, Dong Huang, Eric P Xing, and Zhiqiang Shen. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4942–4952, 2022.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023c.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2018.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 650–663, 2021.

Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization techniques*, pp. 234–243. Springer, 1978.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems*, pp. 2049–2057, 2013.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *International Conference on Learning Representations*, 2018.

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

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, 2019.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pp. 7197–7206. PMLR, 2020.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Dongmin Park, Seola Choi, Doyoung Kim, Hwanjun Song, and Jae-Gil Lee. Robust data pruning under label noise via maximizing re-labeling accuracy. *Advances in Neural Information Processing Systems*, 36, 2024.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*, pp. 4095–4104. PMLR, 2018.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations*.

Omead Pooladzandi, David Davini, and Baharan Mirzasoleiman. Adaptive second order coresets for data-efficient machine learning. In *International Conference on Machine Learning*, pp. 17848–17869. PMLR, 2022.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Hardik Sharma, Jongse Park, Naveen Suda, Liangzhen Lai, Benson Chau, Vikas Chandra, and Hadi Esmaeilzadeh. Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural network. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 764–775. IEEE, 2018.

Zhiqiang Shen and Eric Xing. A fast knowledge distillation framework for visual recognition. *arXiv preprint arXiv:2112.01528*, 2021.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35: 19523–19536, 2022.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Yuanpeng Tu, Boshen Zhang, Yuxi Li, Liang Liu, Jian Li, Yabiao Wang, Chengjie Wang, and Cai Rong Zhao. Learning from noisy labels with decoupled meta label purifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19934–19943, 2023.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5): 988–999, 1999.

Peisong Wang, Qiang Chen, Xiangyu He, and Jian Cheng. Towards accurate post-training network quantization via bit-split and stitching. In *International Conference on Machine Learning*, pp. 9847–9856. PMLR, 2020.

Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. Open-set label noise can improve robustness against inherent label noise. *Advances in Neural Information Processing Systems*, 34:7978–7992, 2021.

Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1121–1128, 2009.

Gert W Wolf. Facility location: concepts, models, algorithms and case studies. series: Contributions to management science, 2011.

Xiaobo Xia, Bo Han, Yibing Zhan, Jun Yu, Mingming Gong, Chen Gong, and Tongliang Liu. Combating noisy labels with sample selection by mining high-discrepancy examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1833–1843, 2023a.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2023b.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In *ECAI 2012*, pp. 870–875. IOS Press, 2012.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Jiangchao Yao, Jiajie Wang, Ivor W Tsang, Ya Zhang, Jun Sun, Chengqi Zhang, and Rui Zhang. Deep learning from noisy image labels with quality embedding. *IEEE Transactions on Image Processing*, 28(4): 1909–1922, 2018.

Edouard Yvinec, Arnaud Dapogny, Matthieu Cord, and Kevin Bailly. Powerquant: Automorphism search for non-uniform quantization. In *The Eleventh International Conference on Learning Representations*, 2023.

Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 365–382, 2018.

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.

Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, Zonghao Chen, and Tong Zhang. Probabilistic bilevel coreset selection. In *International Conference on Machine Learning*, pp. 27287–27302. PMLR, 2022.

Bohan Zhuang, Lingqiao Liu, Mingkui Tan, Chunhua Shen, and Ian Reid. Training quantized neural networks with a full-precision auxiliary module. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1488–1497, 2020.

Appendix

This appendix includes an additional introduction to related works, training dynamics, efficiency analysis, extended experimental analysis on data coverage, and a discussion of transferability and generalizability that is not included in the main text due to space limitations. These contents are organized in separate sections as follows:

- Sec. A elaborates the detail of previous representative coreset selection methods and analyzes the reason for failure in quantization-aware training (QAT) scenario.
- Sec. B demonstrate the training dynamics of our ACS with different selection intervals and show how our methods help to stabilize the training and accelerate the convergence.
- Sec. C includes additional experiments of vision transformer model Swin-T on ImageNet-1K with different bitwidth, showing the applicability and scalability of our method.
- Sec. D includes the real training time composition as a supplement of QAT training time comparison in Tab. 3, showing that our coreset selection only incurs a minimal efficiency overhead both with and without knowledge distillation.
- Sec. E provides additional experiments to prove that the improvement of performance with our method does not exclusively come from covering more data from the full training set.
- Sec. F provides detailed experimental results with and without knowledge distillation (KD). Both the accuracy and real training time are shown.
- Sec. G analyzes the transferability and generalizability of our coreset from different models.

A Detailed Introduction of Selected Baselines

In this section, we will introduce the selected baseline coreset selection methods in detail and show the defects of these methods when applied to QAT.

EL2N-Score/GraNd-Score (Paul et al., 2021) The GraNd-score for a given training sample $\{x, y\}$ is defined as $\chi_t(x, y) = E_{w_t} \left\| g_t(x, y) \right\|_2$, which is the expected magnitude of the loss vector with respect to the weights. The importance of each sample is measured by the expected loss gradient norm, which has a similar intuition to our error-vector score d_{EVS}. However, another assumption from GraNd score is that this approximation only holds when the model has been trained for a few epochs. We must perform early training on the current model to have statistics to compute the score. In addition, storing all the gradients incurs significant memory overheads during QAT. The efficiency will be lower than full dataset training with a high subset fraction. The performance with these metrics is sub-optimal as the converged quantized model in the later training epochs of QAT is not considered. In addition, Based on the reproduction from Kirsch (2023) observing that the inconsistency of GraNd at initialization, the performance of GraNd-score-based coreset selection cannot outperform random selection across various datasets and tasks.

Forgetting (Toneva et al., 2019) The core contribution of Forgetting is that forgetting and learning events are defined. In the classification setting with a given dataset $\mathcal{D} = (\mathbf{x}_i, y_i)_i$. For training example $\mathbf{x}_i$ obtained after t steps using SGD, the predicted label is denoted as $\hat{y}_i^t = \arg\max_k p(y_{ik}|\mathbf{x}_i; \theta^t)$. $\mathrm{acc}_i^t = 1_{\hat{y}_i^t = y_i}$ is defined to be a binary indicator encoding whether the classification of the specific example is correct at time step t. Example $\mathbf{x}_i$ undergoes a *forgetting event* when acc_i^t decreases between two consecutive updates: $\mathrm{acc}_i^t > \mathrm{acc}_i^{t+1}$. The example $\mathbf{x}_i$ is misclassified at step $t+1$ after having been correctly classified at step t. Corresponding to the forgetting event, a *learning event* is defined to be $\mathrm{acc}_i^t < \mathrm{acc}_i^{t+1}$. With the statistics of forgetting events in early training, we can select those samples that incur more forgetting events and are more difficult to learn. However, the intuition to select "difficult" samples that incur more misclassification is not always correct. For large-scale datasets (ImageNet-1K, etc.) and datasets with a relatively smaller size (such as MNIST, CIFAR-10, etc.). the difficulties of classification vary significantly. Selecting samples

that are difficult to learn at the very beginning is not always reasonable. In the quantization-aware training setting, the quantized model will first calibrate the quantization parameters and recover the weight at the early stages. We should not select "difficult samples" for calibration and recovery.

Glister (Killamsetty et al., 2021b) The Glister performs data selection based on the assumption that the inner discrete data selection is an instance of (weakly) submodular optimization. Let $V = \{1, 2, \cdots, n\}$ denote a ground set of items (the set of training samples in the setting of coreset selection). Set functions are defined as functions $f : 2^V \to \mathbf{R}$ that operate on subsets of V. A set function f is defined to be a submodular function (Fujishige, 2005) if it satisfies the diminishing returns property that for subsets $S \subseteq T \subseteq V, f(j|S) \triangleq f(S \cup j) - f(S) \geq f(j|T)$. Some natural combinatorial functions (facility location, set cover, concave over modularity, etc.) are submodular functions (Iyer et al., 2020). Submodularity is also very appealing because a simple greedy algorithm achieves a $1 - 1/e$ constant factor approximation guarantee (Nemhauser et al., 1978) for the problem of maximizing a submodular function subject to a cardinality constraint (which most data selection approaches involve). Moreover, several variants of the greedy algorithm have been proposed, further scaling up submodular maximization to almost linear time complexity (Minoux, 1978; Mirzasoleiman et al., 2013). However, the greedy algorithm still consumes tremendous time compared to other metric-based selection methods, which only need to perform sorting on the proposed metrics. We also observed that the greedy algorithm fails on QAT for low-bit settings.

kCenterGreedy (Sener & Savarese, 2018) As one of the most straight-forward geometry-based coreset selection methods, the intuition is simple: we can measure the similarity by the distance of data points and select the center point of data cluster can make use of the redundancy in the dataset. This method aims to solve the *minimax facility location* problem (Wolf, 2011), which is defined to be selecting k samples as subset S from the full dataset T such that the longest distance between a data point in $T \backslash S$ and its closest data point in S is minimized:

$$\min_{S \subset T} \max_{x_i \in T \backslash S} \min_{x_j \in S} \mathcal{D}(x_i, x_j), \tag{16}$$

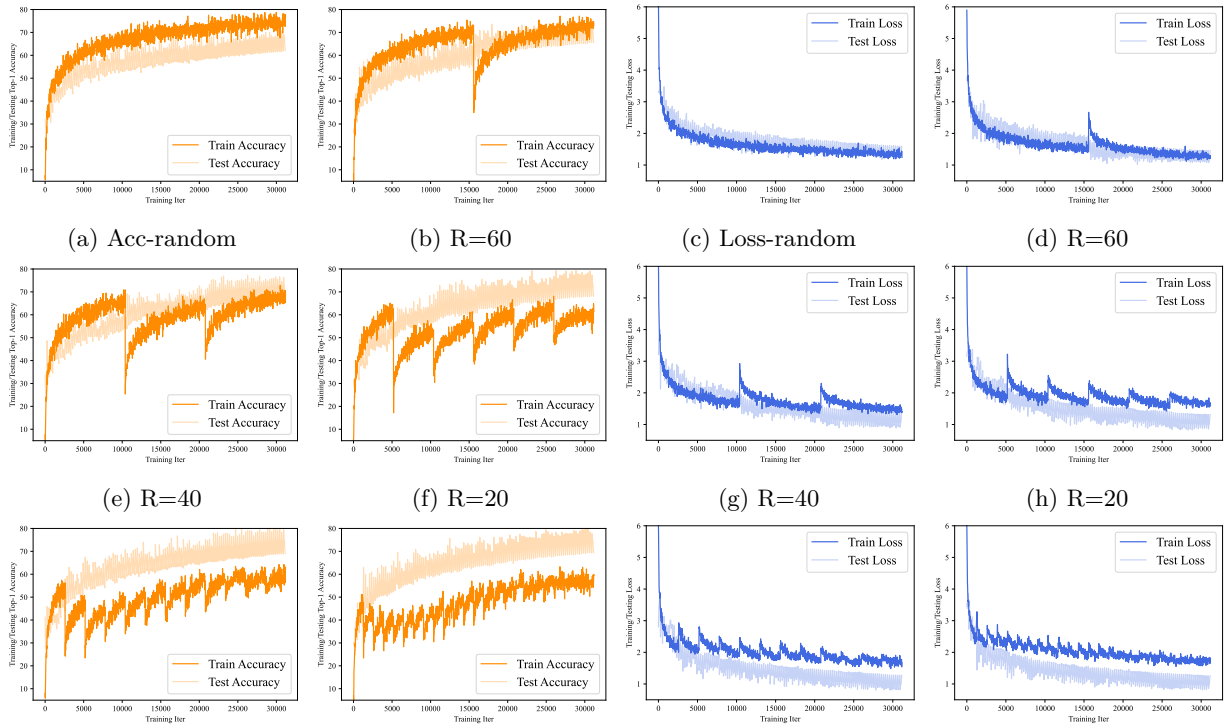
where $\mathcal{D}(\cdot, \cdot)$ is the distance measurement function. The problem is NP-hard, and a greedy approximation known as k-Center Greedy has been proposed in Sener & Savarese (2018). Similarly, as the greedy algorithm is involved, the efficiency is negatively influenced, which is not affordable for our QAT setting.

Contextual Diversity (CD) (Agarwal et al., 2020) Specifically designed for coreset selection for deep convolutional neural networks (CNNs), Contextual Diversity (CD) fully leverages the ambiguity in feature representations. The ambiguity is quantified as class-specific confusion as the selection metric. Assume $C = \{1, \ldots, n_C\}$ is the set of classes predicted by a CNN-based model. For a region r within an input image I, let $P_r = P_r(\widehat{y} \mid I; \theta)$ be the softmax probability vector as predicted by the model θ. The pseudo-label for the region $r \subseteq I$ is defined as $\widehat{y}_r = \arg\max_{j \in C} P_r[j]$, where the notation $P_r[j]$ denotes the jth element of the vector. For a given model θ over the unlabeled I, the class-specific confusion for class c is defined as $P_I^c = \frac{1}{|I^c|} \sum_{I \in I^c} \left[\frac{\sum_{r \in R_I^c} w_r P_r(\widehat{y}|I;\theta)}{\sum_{r \in R_I^c} w_r} \right]$ with $w_r \geq 0$ as the mixing weights. The pairwise contextual diversity, which is the KL-divergence of the metric between two samples I_1 and I_2 could be used as a distance metric to replace the Euclidean distance in the previous kCenterGreedy (Sener & Savarese, 2018). As this work basically follows kCenterGreedy (Sener & Savarese, 2018) to perform coreset selection, the defects also lie in the efficiency of the greedy algorithm and failure with the low-bit setting.

Moderate Coreset (Xia et al., 2023b) As the most recent work of coreset selection, Moderate has optimal efficiency as no greedy searching or back-propagation is involved during selection. Previous methods rely on score criterion that only applies to a specific scenario. This work proposes to utilize the score median as a proxy of the statistical score distribution and select the data points with scores close to the score median into the coreset. The main drawback of this method applied to quantization-aware training is that quantized distribution is not considered.

B QAT Training Dynamics with ACS

In this section, we report the training dynamics, including the training loss and training accuracy of QAT ResNet-18 on ImageNet-1K coreset with a 10% subset fraction. As can be seen from the results in Fig. 5, when the coreset changes adaptively to the current iterations, the accuracy will drop, and loss will increase significantly at the specific iteration of updating the subset. However, the quantized model will converge fast on the new subset. The training with an adaptive coreset can effectively help avoid overfitting and improve the final performance.



(a) Acc-random (b) R=60 (c) Loss-random (d) R=60

(e) R=40 (f) R=20 (g) R=40 (h) R=20

(i) R=10 (j) R=5 (k) R=10 (l) R=5

Figure 5: Training/Testing loss/accuracy comparison of 4/4-bit quantized ResNet-18 on 10% coreset of ImageNet-1K with different selection interval R.

C Additional Experiments of Vision Transformers

In addition to the three experimental settings (MobileNetV2-CIFAR100 at W2/A32 bits, ResNet18-ImageNet1k at W4A4 bits, RetinaNet-COCO at W4A4 bits) in the main text, we prove two additional experiment settings: Swin-T Liu et al. (2021)-ImageNet1k at W4A4 and W2A2 bits to better demonstrate the applicability and scalability of our method. We train the Swin-T for 300 epochs with a batch size of 512 and learning rate of 5e-4. The selection interval R is set to be 30. The results are listed in the Tab. 9, where our method outperforms various baseline methods by a great margin.

D Training Time Composition

As a supplement of the detailed QAT training time comparison in Tab. 3, we also provide the detailed training time composition of training and selection in Tab. 10. Since KD will change the loss function of SGD training and is difficult to break down these two parts, we provide two sets of results with and without KD. The experiments are also conducted on quantized ResNet-18 on ImageNet-1K coreset with 2 NVIDIA RTX 3090 GPUs.

Table 9: Comparison of applying different coreset selection methods to 4/4 and 2/2-bit quantized Swin-T on ImageNet-1K dataset with various subset fractions. When full data are selected ($\mathcal{S} = 100\%$), the accuracy is 80.6% and 70.2% for W4A4 and W2A2.

Method/Fraction (%)	$\mathcal{S}=10\%$		$\mathcal{S}=30\%$		$\mathcal{S}=50\%$		$\mathcal{S}=60\%$		$\mathcal{S}=70\%$		$\mathcal{S}=80\%$	
Bitwidth (W/A)	**4/4**	**2/2**	**4/4**	**2/2**	**4/4**	**2/2**	**4/4**	**2/2**	**4/4**	**2/2**	**4/4**	**2/2**
Random	69.3	57.8	73.3	62.9	76.3	65.2	77.4	66.0	78.5	67.1	79.1	67.8
EL2N Paul et al. (2021)	66.8	56.0	70.8	60.5	75.9	64.1	77.2	65.2	78.3	66.4	78.9	67.9
Glister Killamsetty et al. (2021b)	65.0	55.1	69.9	59.0	73.5	63.5	75.8	64.9	77.0	65.8	78.3	67.0
Moderate Xia et al. (2023b)	67.9	58.2	73.3	62.8	76.9	65.2	77.8	66.5	79.0	67.3	79.6	68.5
Ours	76.9	66.4	78.0	67.9	78.7	68.4	79.1	68.9	79.5	69.3	80.4	70.0

The results show that coreset selection only incurs a minimal efficiency overhead both with and without KD. The selection time is a constant and is only related to the full dataset size and the selection intervals R. The training with KD and without KD only influence the backpropagation time of the training and there is no backward in the selection phase, which makes the time of selection the same (3.2h) across all settings.

Table 10: Composition of training time on QAT of quantized ResNet-18 on ImageNet-1K with different subset fractions and KD settings. The bitwidth for quantized ResNet-18 is 4/4 for weights/activations.

Stage/Fraction (%)	Apply KD?	10%	30%	50%	60%	70%	80%
Ours-all	✓	11.3h	20.7h	36.1h	42.2h	48.0h	53.8h
Ours-selection	✓	3.2h	3.2h	3.2h	3.2h	3.2h	3.2h
Ours-training	✓	8.1h	17.5h	32.9h	39.0h	44.8h	50.6h
Ours-all		9.7h	16.0h	29.5h	34.4h	40.2h	45.4h
Ours-selection		3.2h	3.2h	3.2h	3.2h	3.2h	3.2h
Ours-training		6.5h	15.8h	26.3h	31.2h	37.0h	42.2h

E Coreset Coverage Analysis

The advantages of the proposed quantization-aware Adaptive Coreset Selection (ACS) algorithms are two-fold: adaption and diversity. In this section, we further demonstrate that improving diversity by covering all the training data in different subsets is not optimal compared to our adaptive coreset. We use the "Full Coverage Split" of the CIFAR-100 dataset as our baseline, which is to select non-overlapped samples randomly into the subset of fraction S in the first $\lceil 1/S \rceil$ epochs. It is guaranteed that all the training samples are included, but the sequence is random in this setting. We apply QAT to a 2/32-bit quantized MobileNet-V2 on the random subset, "Full Coverage Split" subset, and coreset using our methods. The selection interval R is the same across all settings. When the subset fraction S is large, the difference in coverage rate with various methods is minor. We then only evaluate on the $S \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$ The results are shown in Tab. 11, where we can see that "Full Coverage Split" has limited superiority on performance compared to random baseline, especially when the subset fraction R is large. Our method outperforms the other two settings across all fractions, proving that both adaption and diversity help to improve the performance.

Table 11: Comparision of Top-1 Accuracy with Random vs. Full coverage split vs. Ours

Method/Fraction (%)	10%	20%	30%	40%	50%
Random	62.25±0.71	64.07±0.47	65.22±0.41	65.55±0.80	66.24±0.55
Full Coverage Split	62.55±0.65	64.22±0.81	65.34±1.17	65.69±0.69	66.20±0.89
Ours	**63.37±0.55**	**65.91±0.40**	**66.41±0.31**	**66.86±0.72**	**67.19±0.65**

F Detailed Experimental Results without Knowledge Distillation

Knowledge Distillation (KD) is a normal and "default" setting for all previous quantization work (Mishra & Marr, 2018; Zhuang et al., 2020; Liu et al., 2022; Bhalgat et al., 2020; Huang et al., 2022), including the LSQ quantization (Bhalgat et al., 2020) we use in this paper. For a fair comparison with previous work, we equally utilize KD for all the experiments in this work regardless of the precision and dataset fraction. The full-data training baseline also involves training with knowledge distillation.

To verify the effectiveness of our method without KD, we remove the knowledge distillation in our method. Since the DS is built on the soft label of the teacher model, we do not use it in our selection w/o KD. Only EVS is applied as the selection metric. We follow the same settings for quantized ResNet-18 on ImageNet-1K as shown in our paper. The training time and accuracy are shown as follows in Tab. 12. When full data are selected ($S = 100\%$), the Top-1 accuracy is 70.21% and training time is 3.1h without KD. From the results shown in Tab. 12, we can see that our method still outperforms previous SOTA without KD and the training efficiency is still optimal.

Table 12: Comparison of Top-1 Accuracy and the training time of QAT of quantized ResNet-18 on ImageNet-1K with different subset fractions without KD settings. The bitwidth for quantized ResNet-18 is 4/4 for weights/activations.

Method/Fraction (%)	10%		30%		50%		60%		70%		80%	
	Acc	Time	Acc	Time	Acc	Time	Acc	Time	Acc	Time	Acc	Time
EL2N w/o KD	59.71	10.6h	63.50	17.1h	65.19	31.3h	66.38	35.0h	67.90	42.9h	69.01	48.0h
Glister w/o KD	62.41	11.9h	65.18	22.7h	66.47	34.5h	67.05	41.7h	68.81	50.3h	69.45	56.9h
Ours w/o KD	66.91	9.7h	68.77	16.0h	69.25	29.5h	69.66	34.4h	69.85	40.2h	70.03	45.4h

G Coreset Transferability and Generalizability

The theoretical analysis of the importance of each sample in QAT is model-specific, which means that data selected using our method is the optimal coreset for the current model. However, if the coreset discovered by a specific pre-trained model is applicable to other models, our method can be a potential solution for the model-agnostic coreset method. In addition, if there is a significant coreset overlap across different models, our method can even solve dataset distillation.

We first design experiments to verify the transferability of the coreset using our method. We use 2/32-bit ResNet-18 to generate a coreset on CIFAR-100 and apply it to MobileNet-V2. The results are shown in Tab. 13. As the ResNet-18 coreset performs better than the random subset on MobileNet-V2, our coreset has some extent of generalization ability to unseen network structures, but the effectiveness is still worse than the MobileNetV2 coreset.

Table 13: Comparision of Top-1 Accuracy with different coreset of MobileNet-V2 on CIFAR-100.

Method	Coreset Generation Model	10%	20%	30%	40%	50%
Random	-	62.25±0.71	64.07±0.47	65.22±0.41	65.55±0.80	66.24±0.55
Ours	MobileNetV2	63.37±0.55	65.91±0.40	66.41±0.31	66.86±0.72	67.19±0.65
Ours	ResNet-18	62.94±0.45	64.18±0.73	65.70±0.40	65.90±0.51	66.86±0.42

We then design experiments to verify the overlap between the coreset of different models. We analyze the 10% fraction coreset selected using our method in the final epochs of 2/32-bit ResNet-18, ResNet-50, and MobileNet-V2. The percentage of the overlap data is shown in Tab. 14.

We empirically find out that there is some overlap between the coreset of different models, and the overlap is more significant when these two models have a similar structure (ResNet-18 and ResNet-50). We then further apply majority voting from the CIFAR-100 coreset of five models (ResNet-18, ResNet-34, ResNet-50,

Table 14: Coreset overlap of different model pairs on CIFAR-100.

Model Pair	Coreset Overlap Percentage
{ResNet-18, MobileNetV2}	37.1%
{ResNet-18, ResNet-50}	**77.3%**
{ResNet-50, MobileNetV2}	29.0%

MobileNet-V2, MobileNet-V3) to generate a "general coreset". This "general coreset" of 10% data fraction is applied to a 2/32-bit quantized Vision Transformer ViT-B/16 (Dosovitskiy et al., 2020), where we only got a Top-1 accuracy of 72.3%, which is lower than 10% coreset using our method of 78.9% and even lower than 10% random coreset of 74.6%. We conclude that our method is model-specific, which has some extent of generalization ability to unseen network structures but cannot be a general data distillation approach.