SHARPNESS-AWARE QUANTIZATION FOR DEEP NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

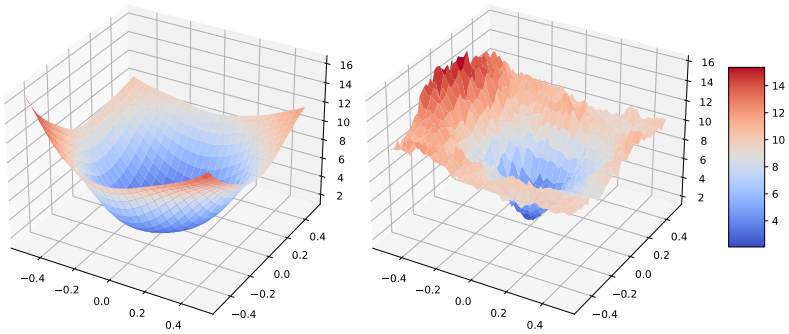
ABSTRACT

Network quantization has gained increasing attention since it can significantly reduce the model size and computational overhead. However, due to the discrete nature of quantization, a small change in full-precision weights might incur large change in quantized weights, which leads to severe loss fluctuations and thus results in sharp loss landscape. The fluctuating loss makes the gradients unstable during training, resulting in considerable performance degradation. Recently, Sharpness-Aware Minimization (SAM) has been proposed to smooth the loss landscape and improve the generalization performance of the models. Nevertheless, how to customize SAM to the quantized models is non-trivial due to the effect of the clipping and discretization in quantization. In this paper, we propose a novel method, dubbed Sharpness-Aware Quantization (SAQ), to smooth the loss landscape and improve the generalization performance of the quantized models, which explores the effect of SAM in model compression, particularly quantization for the first time. Specifically, we first propose a unified view for quantization and SAM, where we consider them as introducing quantization noises and adversarial perturbations to the model weights. According to whether the quantization noises and adversarial perturbations depend on each other, SAQ can be divided into three cases. We then analyze and compare different cases comprehensively. Extensive experiments on both convolutional neural networks and Transformers show that SAQ improves the generalization performance of the quantized models, yielding the SOTA results in uniform quantization. For example, on ImageNet, our SAQ outperforms the model trained with the conventional optimization procedure (*i.e.*, SGD) by 1.1% on the Top-1 accuracy on 4-bit ResNet-50. Our 4-bit ResNet-34 surpasses the previous SOTA quantization method by 1.0% on the Top-1 accuracy.

1 INTRODUCTION

With powerful high-performance computing and massive labeled data, convolutional neural networks (CNNs) and Transformers have dramatically improved the accuracy of many computer vision (CV) and natural language processing (NLP) tasks, such as image classification (He et al., 2016; Dosovitskiy et al., 2021), dense prediction (Ren et al., 2015; Carion et al., 2020), sentence classification (Wang et al., 2019a; Devlin et al., 2019), and machine translation (McCann et al., 2017; Vaswani et al., 2017), to the level of being ready for real-world applications. Despite the remarkable breakthroughs that deep learning has achieved, the considerable computational overhead and model size greatly hampers the development and deployment of deep learning techniques at scale, especially on resource-constrained devices such as mobile phones. To obtain compact models, many network quantization methods (Hubara et al., 2016; Zhou et al., 2016) have been proposed to tackle the efficiency bottlenecks.

Despite the high compression ratio, training a low-precision model is very challenging due to the discrete and non-differentiable nature of network quantization. Compared with the full-precision ones, the low-precision models represent weights, activations, and even gradients with only a small set of values, which limits the representation power of the quantized models. As shown in Figure 1, a slight change in full-precision weights coming from the gradient update or quantization noises might incur large change in quantized weights due to discretization, which leads to drastic loss fluctuations and results in much sharper loss landscape (Liu et al., 2021a). As a result, the enormous



(a) Full-precision ResNet-18 (b) 2-bit ResNet-18

Figure 1: The loss landscapes of the full-precision and 2-bit ResNet-18 models on ImageNet. We plot the loss landscapes using the visualization method in (Li et al., 2018). More visualizations can be found in Section C of the appendix.

loss fluctuations make gradients unreliable during optimization, which misleads weight update and thus incurs a performance drop.

There have been some studies showing that flat minima of the loss function found by stochastic gradient-based methods result in good generalization (Hochreiter & Schmidhuber, 1995; Keskar et al., 2017; Dziugaite & Roy, 2017; Jiang et al., 2020). Recently, Sharpness-Aware Minimization (SAM) (Foret et al., 2021) and its variants (Kwon et al., 2021; Zhuang et al., 2022; Kim et al., 2022) have been proposed to smooth the loss landscape and significantly improve model generalization ability. Specifically, SAM first introduces perturbations to model weights and then minimizes a perturbed loss to seek parameters that lie in neighborhoods with uniformly low training loss. However, all the existing methods are based on full-precision over-parameterized models. *How to perform sharpness-aware minimization on the compressed models has rarely been explored, especially on the quantized ones, which is a new and important problem.* A simple solution is to directly apply SAM to train the quantized models, which may lead to sub-optimal performance since the clipping and discretization operation in quantization might hamper the introduced perturbations.

In this paper, we propose a novel method, called Sharpness-Aware Quantization (SAQ), to find minima with both low loss value and low loss curvature and thus improve the generalization performance of the quantized models. To our knowledge, this is a pioneering work to study the effect of SAM in model compression, especially in network quantization. To this end, we first provide a unified view for quantization and SAM, where we formulate them as introducing quantization noises $\boldsymbol{\epsilon}_q$ and adversarial perturbations $\hat{\boldsymbol{\epsilon}}_s$ to the model weights. According to whether $\boldsymbol{\epsilon}_q$ and $\hat{\boldsymbol{\epsilon}}_s$ are dependent on each other, our SAQ can be split into three cases. We then study and compare these cases comprehensively. Extensive experiments on both CNNs and Transformers show the promising performance of our proposed method.

Our main contributions are summarized as follows:

- We propose a Sharpness-Aware Quantization (SAQ) method to seek flatter minima for the quantized models in order to materially improve the generalization performance. To our knowledge, this is a pioneering work that jointly performs the model compression (*i.e.*, quantization) and the loss landscape smoothing.
- We provide a unified view for the landscape smoothing of the quantized models, where we consider quantization and SAM as introducing quantization noises and adversarial perturbations to the model weights, respectively. Relying on this, we present three cases of SAQ according to whether quantization noises and adversarial perturbations depend on each other. We then analyze and make comprehensive comparisons among different cases.
- Extensive experiments on CNNs and Transformers show that our SAQ improves quantized models' generalization performance and performs favorably against SOTA uniform quantization methods. For example, on ImageNet, our 4-bit ResNet-50 surpasses the previous standard optimization scheme (*i.e.*, SGD) by 1.1% on the Top-1 accuracy. Moreover, our 4-bit ResNet-34 exceeds the SOTA method by 1.0% on the Top-1 accuracy.

2 Related Work

Network quantization. Network quantization seeks to reduce the model size and computational cost by mapping weights, activations, and even gradients of a CNN or ViT to low-precision ones. Existing quantization methods can be roughly divided into two categories according to the quantization bitwidth, namely, fixed-point quantization (Zhou et al., 2016; Cai et al., 2017; Hou & Kwok, 2018; Choi et al., 2018; Zhuang et al., 2018; Zhang et al., 2018; Jung et al., 2019; Esser et al., 2020; Chen et al., 2021; Kim et al., 2021b; Liu et al., 2021b; Han et al., 2021) and binary quantization (Hubara et al., 2016; Rastegari et al., 2016; Liu et al., 2018; Lin et al., 2017; Liu et al., 2021a; Bai et al., 2021; Qin et al., 2022). To reduce the quantization error, existing methods (Choi et al., 2018; Zhang et al., 2018; Jung et al., 2019; Esser et al., 2020; Bhalgat et al., 2020; Yamamoto, 2021) explicitly parameterize the quantizer and train it jointly with network parameters. To reduce the optimization difficulty incurred by the non-differentiable discretization, extensive methods (Ding et al., 2019; Yang et al., 2019; Gong et al., 2019; Lee et al., 2021; Kim et al., 2021a) have been proposed to approximate the gradients. To encourage more information to be maintained by the quantized weights, several weight regularization methods (Han et al., 2021; Liu et al., 2022b) have been proposed to alleviate the discrepancy between the full-precision and low-precision weights. Compared with these methods, our SAQ focuses on improving the generalization performance of the quantized models from a new perspective by smoothing the loss landscape.

Loss geometry and generalization. Hochreiter *et al.* (Hochreiter & Schmidhuber, 1995) pioneered the proposition that flat local minima may generalize better in neural networks. Following that, several studies have been proposed to investigate the relation between the geometry of the loss landscape and the generalization performance of the models (Keskar et al., 2017; Smith & Le, 2018; Dziugaite & Roy, 2017; Chaudhari et al., 2017; Jiang et al., 2020; Moosavi-Dezfooli et al., 2019; Liu et al., 2020). Recently, Sharpness-Aware Minimization (SAM) (Foret et al., 2021) seeks to find parameters that lie in a region with uniformly low loss value and shows promising performance across various architectures and benchmark datasets. Concurrent works have also been proposed to introduce adversarial weight perturbations to improve the robustness against adversarial examples (Wu et al., 2020) or generalization performance (Zheng et al., 2021). However, the computational overhead of these methods is roughly doubled compared with those using conventional optimizers (*e.g.*, SGD). To address this issue, ESAM (Du et al., 2022a), LookSAM (Liu et al., 2022a) and SAF (Du et al., 2022b) have been proposed to accelerate the SAM optimization without performance drop. Apart from the efficiency issues, several methods including ASAM (Kwon et al., 2021), GSAM (Zhuang et al., 2022) and Fisher SAM (Kim et al., 2022) have been proposed to improve the performance of SAM. More recently, SAM has been applied to improve the performance of the pruned models (Na et al., 2022). While these methods target on full-precision models, our proposed SAQ focuses on improving the generalization performance of the quantized models, which is a pioneering one in the sense that we jointly perform model compression (*i.e.*, quantization) and loss landscape smoothing.

3 Preliminary

3.1 Network Quantization

In this paper, we use uniform quantization which is hardware-friendly (Zhou et al., 2016). Given an L-layer deep model, let w^l and x^l be the weight and input activation w.r.t. the l-th layer. For simplicity, we omit the layer index l afterwards. Before performing quantization, we first normalize weight w and input activation x into the scale of $[0, 1]$ by applying clipping as

$$\hat{w} = \begin{cases} \frac{1}{2}\left(\frac{w}{\alpha_w} + 1\right), & \text{if } -1 < \frac{w}{\alpha_w} < 1 \\ 0, & \text{if } \frac{w}{\alpha_w} \leq -1 \\ 1, & \text{if } \frac{w}{\alpha_w} \geq 1 \end{cases}, \quad \hat{x} = \begin{cases} \frac{x}{\alpha_x}, & \text{if } 0 < \frac{x}{\alpha_x} < 1 \\ 0, & \text{if } \frac{x}{\alpha_x} \leq 0 \\ 1, & \text{if } \frac{x}{\alpha_x} \geq 1 \end{cases}, \tag{1}$$

where α_w and α_x are layer-wise trainable clipping levels that limit the range of weight and activation, respectively. We then quantize the normalized value $\hat{z} \in \{\hat{w}, \hat{x}\}$ to the discrete one $\bar{z} \in \{\bar{w}, \bar{x}\}$ by $\bar{z} = D(\hat{z}, s) = s \cdot \lfloor \frac{\hat{z}}{s} \rceil$, where $\lfloor \cdot \rceil$ is a rounding operator that returns the nearest integer of a given value and $s = 1/(2^b - 1)$ is the normalized step size for b-bit quantization. Lastly, we obtain the

quantized w and x by

$$Q_w(w) = \alpha_w(2\bar{w} - 1),\ Q_x(x) = \alpha_x \bar{x}. \tag{2}$$

During training, the rounding operation $\lfloor \cdot \rceil$ is non-differentiable. To overcome this issue, following (Zhou et al., 2016; Hubara et al., 2016), we apply the straight-through estimation (STE) (Bengio et al., 2013) to approximate the gradient of the rounding operator by identity mapping for backpropagation, namely, $\partial \bar{z}/\partial \hat{z} \approx 1$.

3.2 Sharpness-aware Minimization

Without loss of generality, let $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the training data. The goal of model training is to minimize the empirical risk $\mathcal{L}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{w}, \mathbf{x}_i, y_i)$, where $\ell(\mathbf{w}, \mathbf{x}_i, y_i)$ is a loss function for the sample $(\mathbf{x}_i, y_i)$ with model weights $\mathbf{w}$. Instead of seeking a single place with a local minimal loss, Sharpness-Aware Minimization (Foret et al., 2021) (SAM) seeks a region that has uniformly low training loss (both low loss and low curvature). Specifically, the formulation of SAM is a min-max optimization problem which is defined as

$$\min_{\mathbf{w}} \max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} \mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon}), \tag{3}$$

where ρ is a pre-defined constant that constrains the radius of the neighborhood. In Eq. (3), the inner optimization problem attempts to find weight perturbations $\boldsymbol{\epsilon}$ in an ℓ_2 Euclidean ball with radius ρ that maximizes the perturbed loss $\mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon})$. To solve the inner problem, SAM approximates the optimal $\boldsymbol{\epsilon}$ to maximize $\mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon})$ using a first-order Taylor expansion as

$$\hat{\boldsymbol{\epsilon}} = \arg\max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} \mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon}) \approx \arg\max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} \mathcal{L}(\mathbf{w}) + \boldsymbol{\epsilon}^\top \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}) \approx \rho \frac{\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})}{\|\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})\|_2}. \tag{4}$$

By substituting Eq. (4) back into Eq. (3), we then have the following optimization problem:

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w} + \hat{\boldsymbol{\epsilon}}). \tag{5}$$

Lastly, SAM updates the model weights based on the gradient $\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})|_{\mathbf{w}+\hat{\boldsymbol{\epsilon}}}$.

4 Proposed Method

As shown in Figure 1, the low-precision model shows a much sharper loss landscape compared with the full-precision one. Thus, small perturbations on the full-precision weights may incur large changes in the quantized weights, which leads to severe loss oscillation. As a result, the gradients are unstable during training, which might mislead weight update and the resulting quantized model might converge to poor local minima.

4.1 Unified View for Sharpness-aware Quantization

Motivated by SAM, we propose Sharpness-Aware Quantization (SAQ) to smooth the loss landscape and improve the generalization performance of the quantized models. Specifically, we consider quantization and SAM as introducing quantization noises $\boldsymbol{\epsilon}_q$ and adversarial perturbations $\boldsymbol{\epsilon}_s$ to the model weights $\mathbf{w}$, respectively, which provides a unified view for the loss landscape smoothing of the quantized models. Specifically, the optimization problem can be defined as

$$\min_{\mathbf{w}, \boldsymbol{\alpha}_w, \boldsymbol{\alpha}_x} \mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon}_q + \hat{\boldsymbol{\epsilon}}_s) \text{ where } \hat{\boldsymbol{\epsilon}}_s = \arg\max_{\|\boldsymbol{\epsilon}_s\|_2 \le \rho} \mathcal{L}_p(\mathbf{w}), \tag{6}$$

where $\mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon}_q + \hat{\boldsymbol{\epsilon}}_s)$ is a perturbed quantization loss and $\mathcal{L}_p(\mathbf{w})$ is a perturbed loss. Note that $\boldsymbol{\epsilon}_q$ and $\hat{\boldsymbol{\epsilon}}_s$ are the outputs of some functions and will be discussed in Section 4.2.

4.2 Learning for Sharpness-aware Quantization

To solve the optimization problem in Eq. (6), we need to obtain $\boldsymbol{\epsilon}_q$ as well as $\hat{\boldsymbol{\epsilon}}_s$. According to whether $\boldsymbol{\epsilon}_q$ and $\hat{\boldsymbol{\epsilon}}_s$ are dependent on each other, we can transform the loss function in Eq. (6) to different objectives, as shown in Table 1. For convenience, we define the quantization error function $\boldsymbol{\epsilon}_q(\mathbf{w})$ and the perturbation function $\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})$ as

$$\boldsymbol{\epsilon}_q(\mathbf{w}) = Q_w(\mathbf{w}) - \mathbf{w},\ \hat{\boldsymbol{\epsilon}}_s(\mathbf{w}) = \rho \frac{\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})}{\|\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})\|_2}. \tag{7}$$

Case 1: We calculate the quantization noises $\boldsymbol{\epsilon}_q$ and optimal perturbations $\hat{\boldsymbol{\epsilon}}_s$ independently. In this case, the perturbed loss is defined as $\mathcal{L}_p(\mathbf{w}) = \mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_s)$. By maximizing the perturbed loss with ℓ_2-norm constraint, the optimal perturbations can be approximated by $\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})$. In this way, the optimization problem can be transformed to

Table 1: Objectives for different cases of SAQ.

Name	Objective function
Unified	$\mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_q+\hat{\boldsymbol{\epsilon}}_s)$
Case 1	$\mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w})+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}))$
Case 2	$\mathcal{L}((\mathbf{w}+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}))+\boldsymbol{\epsilon}_q(\mathbf{w}+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})))$
Case 3	$\mathcal{L}((\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w}))+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w})))$

$$\min_{\mathbf{w},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_x} \mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w})+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})). \tag{8}$$

With Eq. (7), we have $\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w}) = Q_w(\mathbf{w})$. Then, the above problem can be rewritten as

$$\min_{\mathbf{w},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_x} \mathcal{L}(Q_w(\mathbf{w})+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})). \tag{9}$$

In this case, the optimal perturbations introduced to the quantized weights $Q_w(\mathbf{w})$ depend on the gradient of the full-precision weights $\mathbf{w}$. Using the chain rule, the full-precision weights' gradient can be computed by

$$\frac{\partial \mathcal{L}_p(\mathbf{w})}{\partial w_i} = \frac{\partial \mathcal{L}_p(\mathbf{w})}{\partial Q_w(w_i)}\frac{\partial Q_w(w_i)}{\partial w_i} = \begin{cases} \frac{\partial \mathcal{L}_p(\mathbf{w})}{\partial Q_w(w_i)} & \text{if } -1 \le \frac{w_i}{\alpha_w^l} \le 1 \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

where w_i is the i-th element of $\mathbf{w}$ for layer l and α_w^l is the corresponding clipping level. Due to the clipping operation, the difference between the full-precision weights' gradient $\partial\mathcal{L}_p(\mathbf{w})/\partial w_i$ and the quantized weights' gradient $\partial\mathcal{L}_p(\mathbf{w})/\partial Q_w(w_i)$ results in a perturbation mismatch problem, which makes the training process noisy and degrades the quantization performance.

Besides, Case 1 assumes that $\boldsymbol{\epsilon}_q$ and $\hat{\boldsymbol{\epsilon}}_s$ are computed independently, which ignores the dependency between them. To address this issue, we introduce another two cases of SAQ in the following.

Case 2: We first combine model weights with the optimal perturbations $\hat{\boldsymbol{\epsilon}}_s$ and then introduce the quantization noises $\boldsymbol{\epsilon}_q$ to the perturbed model weights. In this way, the optimization problem is transformed to

$$\min_{\mathbf{w},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_x} \mathcal{L}((\mathbf{w}+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}))+\boldsymbol{\epsilon}_q(\mathbf{w}+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}))). \tag{11}$$

Same as Case 1, the perturbed loss is $\mathcal{L}_p(\mathbf{w}) = \mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_s)$ and the optimal perturbations can be obtained by $\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})$. In this case, the quantization noises $\boldsymbol{\epsilon}_q(\mathbf{w}+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}))$ is represented as a function of the optimal perturbations $\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})$. Using Eq. (7), we reformulate the problem as

$$\min_{\mathbf{w},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_x} \mathcal{L}(Q_w(\mathbf{w}+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}))). \tag{12}$$

Nevertheless, the introduced small perturbations may not change the resulting quantized weights due to the discretization process, *i.e.*, $Q_w(\mathbf{w}+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})) = Q_w(\mathbf{w})$. As a result, $\mathcal{L}(Q_w(\mathbf{w}+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w})))$ might be reduced to $\mathcal{L}(Q_w(\mathbf{w}))$, which degenerates to the conventional quantization.

Case 3: We first combine model weights with the quantization noises $\boldsymbol{\epsilon}_q$ and then introduce the optimal perturbations $\hat{\boldsymbol{\epsilon}}_s$. In this way, the optimization problem becomes

$$\min_{\mathbf{w},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_x} \mathcal{L}((\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w}))+\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w}))). \tag{13}$$

In this case, we define the perturbed loss as $\mathcal{L}_p(\mathbf{w}) = \mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w})+\boldsymbol{\epsilon}_s)$ and obtain the optimal perturbations by $\hat{\boldsymbol{\epsilon}}_s(\mathbf{w}+\boldsymbol{\epsilon}_q(\mathbf{w}))$ which is expressed as a function of the quantization noises $\boldsymbol{\epsilon}_q(\mathbf{w})$. With Eq. (7), the optimization problem can be rewritten as

$$\min_{\mathbf{w},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_x} \mathcal{L}(Q_w(\mathbf{w})+\hat{\boldsymbol{\epsilon}}_s(Q_w(\mathbf{w}))), \tag{14}$$

where we introduce perturbations to the quantized weights $Q_w(\mathbf{w})$ rather than the full-precision weights $\mathbf{w}$ as in Case 2. In this way, the introduced perturbations will not be diminished by the quantization operation. Moreover, compared with Case 1, Case 3 does not suffer from the perturbation mismatch issue since the optimal perturbations depend on the gradient of the quantized weights instead of the full-precision ones. In summary, Case 3 is the best suited to smooth the loss landscape of the quantized models.

Final optimization problem. Note that for all cases, we seek a solution on the surface of $\mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_q+\hat{\boldsymbol{\epsilon}}_s)$ instead of the vanilla quantization loss $\mathcal{L}(Q_w(\mathbf{w}))$ as indicated in Eq. (6). Due to the discrete nature of network quantization, the loss gap between $\mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_q+\hat{\boldsymbol{\epsilon}}_s)$ and $\mathcal{L}(Q_w(\mathbf{w}))$ could be amplified with the decrease of bitwidth, which makes optimization challenging and thus leads to

a performance drop, especially for extreme low-bit cases. To reduce the loss gap, we introduce an additional vanilla quantization loss into the objective and reformulate the optimization problem as

$$\min_{\mathbf{w},\boldsymbol{\alpha}_w,\boldsymbol{\alpha}_x} \mathcal{L}(\mathbf{w}+\boldsymbol{\epsilon}_q+\hat{\boldsymbol{\epsilon}}_s)+\mathcal{L}(Q_w(\mathbf{w})) \text{ where } \hat{\boldsymbol{\epsilon}}_s = \arg\max_{\|\boldsymbol{\epsilon}\|_2\leq\rho}\mathcal{L}_p(\mathbf{w}). \tag{15}$$

Similar to Eq. (4), the gradient of $\mathcal{L}(Q_w(\mathbf{w}))$ has been computed during the backpropagation when solving the inner optimization problem. Therefore, we can reuse them while solving the outer optimization problem, which is computationally efficient. By solving the problem in Eq. (15), we enforce the quantized models to find flatter minima with both low loss and low curvature.

5 Experiments

Datasets and evaluation metrics. We evaluate our method on ImageNet (Deng et al., 2009) which is a large-scale dataset containing 1.28 million training images and 50k validation samples with 1k classes. We measure the performance of different methods using the Top-1 and Top-5 accuracy.

Compared methods. To investigate the effectiveness of the proposed method, we apply SAQ to both CNNs and vision Transformers, including ResNet-18 (He et al., 2016), ResNet-34, ResNet-50, MobileNetV2 (Sandler et al., 2018) and ViT (Dosovitskiy et al., 2021). We compare with enormous fixed-point quantization methods, including DoReFa-Net (Zhou et al., 2016), PACT (Choi et al., 2018), LQ-Nets (Zhang et al., 2018), DSQ (Gong et al., 2019), FAQ (McKinstry et al., 2019), QIL (Jung et al., 2019), Auxi (Zhuang et al., 2020), LSQ (Esser et al., 2020), APOT (Li et al., 2020), LSQ+ (Bhalgat et al., 2020), LLSQ (Zhao et al., 2020), DAQ (Kim et al., 2021a), BRECQ (Li et al., 2021a), EWGS (Lee et al., 2021), BR (Han et al., 2021) and LLT (Wang et al., 2022).

Implementation details. Our implementations are based on PyTorch (Paszke et al., 2019). We first train the full-precision models and use them to initialize the low-precision ones. Following LSQ (Esser et al., 2020), we quantize both weights and activations for all matrix multiplication layers, including convolutional layers, fully-connected layers, and self-attention layers. For the first and last layers, we quantize both weights and activations to 8-bit to preserve the performance. We do not apply advanced training strategies such as knowledge distillation in our method.

For CNNs, we use the uniform quantization method mentioned in Section 3.1. Relying on SGD with the momentum term of 0.9, we apply SAQ with Case 3 to train the quantized models unless otherwise specified. Following APOT (Li et al., 2020), we use weight normalization before quantization. We initialize the clipping levels to 1. We fine-tune 90 epochs for ResNet-18, ResNet-34, ResNet-50 and 150 epochs for MobileNetV2. The mini-batch size and weight decay are set to 512 and 1×10^{-4}, respectively. The learning rate is initialized to 0.02 and decreased to 0 following the cosine annealing (Loshchilov & Hutter, 2017). For ViTs, we use LSQ+ (Bhalgat et al., 2020) uniform quantization following Q-ViT (Li et al., 2022). We initialize the clipping levels by minimizing the quantization error following (Li et al., 2021b). Based on AdamW (Loshchilov & Hutter, 2019), we apply SAQ with Case 3 for optimization. The learning rate is initialized to 2×10^{-4} and decreased to 0 using the cosine annealing. The quantized model is trained for 150 epochs with a mini-batch size of 1,024. We do not apply the learning rate warmup and the automatic mixed-precision training strategy following Q-ViT. We put more implementation details in Section A and more ablation studies in Section B of the appendix.

5.1 Main Results

We apply SAQ to quantize ResNet-18, ResNet-34 and ResNet-50 on ImageNet. From Table 2, SAQ outperforms existing SOTA uniform quantization methods by a large margin. The improvement is more obvious with the increase of bitwidth. For example, for 2-bit ResNet-34, the Top-1 accuracy improvement of SAQ over LSQ is 0.2% while for the 4-bit one is 1.0%. We speculate that the loss landscape of the quantized models becomes sharper with the decrease of bitwidths due to the discretization in quantization as shown in Figure A in the appendix. As a result, smoothing the loss landscapes of the 2-bit quantized models is harder than the 4-bit counterparts. Moreover, for 2-bit quantization, deeper models show more obvious accuracy improvement over the SOTA methods. For instance, SAQ surpasses Auxi by 0.7% on 2-bit ResNet-50 while only bringing 0.3% Top-1 accuracy improvement over BR on 2-bit ResNet-18. Note that SAQ even outperforms APOT, a non-uniform quantization method, on different architectures. These results strongly show the promising

Table 2: Performance comparisons of different methods with ResNet-18, ResNet-34 and ResNet-50 on ImageNet. We obtain the results of DoReFa-Net from (Choi et al., 2018). "W/A" refers to the bitwidth of weights and activations, respectively. "FP" represents the Top-1 accuracy of the full-precision models. "-" denotes that the results are not reported.

Network	Method	Bitwidth (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5	Bitwidth (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5
ResNet-18 (FP: 70.7)	DoReFa-Net*	2/2	62.6	84.4	4/4	68.1	88.1
	PACT*	2/2	64.4	85.6	4/4	69.2	89.0
	LQ-Nets*	2/2	64.9	85.9	4/4	69.3	88.8
	DSQ	2/2	65.2	-	4/4	69.6	-
	BRECQ	2/2	-	-	4/4	69.6	-
	FAQ	2/2	-	-	4/4	69.8	89.1
	QIL*	2/2	65.7	-	4/4	70.1	-
	LLT*	2/2	66.0	86.2	4/4	70.4	89.6
	Auxi	2/2	66.7	87.0	4/4	-	-
	DAQ*	2/2	66.9	-	4/4	70.5	-
	LSQ†	2/2	66.9	-	4/4	71.1	90.0
	EWGS*	2/2	67.0	-	4/4	70.6	-
	BR	2/2	67.2	87.3	4/4	70.8	89.6
	APOT*	2/2	67.3	**87.5**	4/4	70.7	89.6
	SAQ (Ours)	2/2	**67.5**	**87.5**	4/4	**71.6**	**90.1**
ResNet-34 (FP: 74.1)	LQ-Nets*	2/2	69.8	89.1	4/4	-	-
	DSQ	2/2	70.0	-	4/4	72.8	-
	FAQ	2/2	-	-	4/4	73.3	91.3
	QIL*	2/2	70.6	-	4/4	73.7	-
	APOT*	2/2	70.9	89.7	4/4	73.8	91.6
	DAQ*	2/2	71.0	-	4/4	73.7	-
	Auxi	2/2	71.2	89.8	-	-	
	EWGS*	2/2	71.4	-	4/4	73.9	-
	LSQ	2/2	71.6	90.3	4/4	74.1	91.7
	SAQ (Ours)	2/2	**71.8**	**90.5**	4/4	**75.1**	**92.2**
ResNet-50 (FP: 76.8)	DoReFa-Net*	2/2	67.1	87.3	4/4	71.4	89.8
	LQ-Net*	2/2	71.5	90.3	4/4	75.1	92.4
	FAQ	2/2	-	-	4/4	76.3	93.0
	PACT*	2/2	72.2	90.5	4/4	76.5	93.2
	APOT*	2/2	73.4	91.4	4/4	76.6	93.1
	LSQ	2/2	73.7	91.5	4/4	76.7	93.2
	Auxi	2/2	73.8	91.4	4/4	-	-
	SAQ (Ours)	2/2	**74.5**	**91.9**	4/4	**77.6**	**93.6**

* denotes that the first and last layers are not quantized.
† represents that models are trained with the weight decay of 1×10^{-4} for fair comparisons.

performance of SAQ. Remarkably, our 4-bit ResNet-34 surpasses the full-precision model by 1.0% on the Top-1 accuracy. One possible reason is that performing quantization with SAQ helps to remove redundancy and regularize the networks. Similar phenomena can also be observed in LSQ.

To show the effectiveness of our method on lightweight models, we apply SAQ to quantize MobileNetV2. From Table 3, our SAQ yields better performance than the SOTA uniform quantization methods. For example, SAQ exceeds BR by 0.2% on the Top-1 accuracy. We also apply SAQ to ViT (Dosovitskiy et al., 2021). We implement LSQ+ following (Li et al., 2022) and compare our method with it. From Table 3, our SAQ shows consistently superior performance over the baseline LSQ+. For example, on ViT-S/16, SAQ obtains 0.8% improvement on the Top-1 accuracy.

5.2 Ablation Studies

Performance comparisons of different cases. To investigate the effectiveness of different cases introduced in Section 4.2, we apply different methods to quantize ResNet-18 and ResNet-50 on Im-

Table 3: Performance comparisons in terms of MobileNetV2, ViT-S/32, ViT-S/16 and ViT-B/32 on ImageNet. We obtain the results of PACT from (Wang et al., 2019b).

Network	Method	Bitwidth (W/A)	Top-1 Acc. (%)	Top-5 Acc. (%)
MobileNetV2 (FP: 71.9)	PACT	4/4	61.4	83.7
	DSQ*	4/4	64.8	-
	BRECQ	4/4	66.6	-
	LLSQ*	4/4	67.4	88.0
	EWGS	4/4	70.3	-
	BR	4/4	70.4	89.4
	SAQ (Ours)	4/4	**70.6**	**89.5**
ViT-S/32 (FP: 68.5)	LSQ+	4/4	68.0	88.1
	SAQ (Ours)	4/4	**68.6**	**88.4**
ViT-S/16 (FP: 77.2)	LSQ+	4/4	76.1	93.0
	SAQ (Ours)	4/4	**76.9**	**93.5**
ViT-B/32 (FP: 70.7)	LSQ+	4/4	72.1	90.4
	SAQ (Ours)	4/4	**72.7**	**90.7**

* denotes that the first and last layers are not quantized.

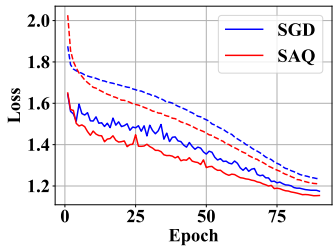
Table 4: Performance comparisons of different cases on ImageNet. λ_{max} denotes the largest eigenvalue of the Hessian of the converged quantized model.

Network	Method	Bitwidth (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5	$\lambda_{\max}$	Bitwidth (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5	$\lambda_{\max}$
ResNet-18	SGD	2/2	66.8	87.3	47.3	4/4	71.1	89.8	59.3
	Case 1	2/2	67.4	87.4	9.6	4/4	71.5	90.0	5.3
	Case 2	2/2	67.3	87.3	18.7	4/4	71.2	89.9	19.5
	Case 3	2/2	**67.5**	**87.5**	**6.5**	4/4	**71.6**	**90.1**	**5.0**
ResNet-50	SGD	2/2	73.9	91.6	60.1	4/4	76.5	93.1	71.8
	Case 1	2/2	74.3	91.8	12.6	4/4	77.3	93.5	6.6
	Case 2	2/2	74.2	91.8	24.4	4/4	77.0	93.3	14.0
	Case 3	2/2	**74.5**	**91.9**	**9.5**	4/4	**77.6**	**93.6**	**6.3**

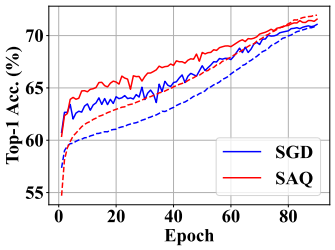
ageNet. We use "SGD" to represent training the quantized models with the vanilla SGD. To measure the loss curvature, we report the largest eigenvalue $\lambda_{\max}$ of the Hessian of the converged quantized models following (Chen et al., 2022; Foret et al., 2021). Here, lower $\lambda_{\max}$ indicates smoother loss landscapes. From Table 4, Case 1, Case 2 and Case 3 all yield significantly higher accuracy and lower $\lambda_{\max}$ than the SGD counterpart. This strongly shows that our method is able to smooth the loss landscape and improve the generalization performance of the quantized models. Among the three cases, Case 2 performs the worst with the lowest accuracy and the highest $\lambda_{\max}$, which suggests that the perturbations introduced by SAM might be diminished due to the discretization, leading to sub-optimal performance. Moreover, Case 3 consistently performs better than Case 1. For example, on 4-bit ResNet-50, Case 3 exceeds Case 1 by 0.3% on the Top-1 accuracy as well as achieving lower $\lambda_{\max}$. These results indicate that the perturbation mismatch issue in Case 1 might degrade the quantization performance.

Besides, we also show the loss and Top-1 accuracy curves of 4-bit ResNet-18 in Figure 2. At the beginning of training, the performance of SAQ is comparable to SGD. After the fourth epoch, we observe that SAQ yields much lower loss and higher Top-1 accuracy for both training and validation. For example, SAQ outperforms SGD by $\sim$2% on the Top-1 validation accuracy at the fifth epoch. These results justify that SAQ converges to a better and flatter local minimum.

Effect of different losses in the objective function. To investigate the effect of different components in the objective in Eq. (15), we apply different methods to quantize ResNet-18. From Table 5, using the loss $\mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon}_q + \hat{\boldsymbol{\epsilon}}_s)$ surpasses the one equipped with the vanilla quantization loss $\mathcal{L}(Q_w(\mathbf{w}, b))$ by 0.3% on the Top-1 accuracy for 2-bit quantization. This result supports that smoothing the loss landscape improves the generalization performance of the quantized models. By combining $\mathcal{L}(Q_w(\mathbf{w}, b))$ and $\mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon}_q + \hat{\boldsymbol{\epsilon}}_s)$, we empirically observe Top-1 accuracy improvement



(a) Loss comparison



(b) Accuracy comparison

Figure 2: The training (dashed line) as well as validation (solid line) losses and accuracy comparisons between SGD and our SAQ with 4-bit ResNet-18 on ImageNet.

Table 5: Effect of different losses in the objective function on ImageNet.

Network	$\mathcal{L}(Q_w(\mathbf{w}, b))$	$\mathcal{L}(\mathbf{w} + \boldsymbol{\epsilon}_q + \hat{\boldsymbol{\epsilon}}_s)$	Bitwidth (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5	Bitwidth (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5
ResNet-18	✓		2/2	66.8	87.3	4/4	71.1	89.8
		✓	2/2	67.1	87.3	4/4	71.3	90.0
	✓	✓	2/2	**67.5**	**87.5**	4/4	**71.6**	**90.1**

of 0.4% and 0.3% for 2-bit and 4-bit quantization, respectively. This strongly justifies that introducing the vanilla quantization loss $\mathcal{L}(Q_w(\mathbf{w}, b))$ into the objective helps to mitigate the loss gap incurred by the introduced perturbations and boosts the performance of the low-precision models.

More results on transfer learning. To evaluate the transfer power of different quantized models, we conduct transfer learning experiments on new datasets, including CIFAR-10 (Krizhevsky & Hinton, 2009), CIFAR-100, Oxford-IIIT Pets (Parkhi et al., 2012), and Oxford Flowers-102 (Nilsback & Zisserman, 2008). We use the quantized models trained on ImageNet to initialize the model weights. We then fine-tune all layers using vanilla SGD. For the results on different transfer learning datasets, we repeat the experiments 5 times and report the mean as well as the standard deviation of the Top-1 accuracy. More implementation details can be found in Section A in the appendix. From Table 6, our SAQ leads to much better transfer performance. For example, on Oxford-IIIT Pets, SAQ quantized 4-bit ResNet-50 brings 1.0% Top-1 accuracy improvement over the SGD counterpart. These results justify that our SAQ is able to improve the generalization performance by smoothing the loss landscape of the quantized models.

Table 6: Transfer performance comparisons on downstream tasks. We measure the performance of different methods using the Top-1 accuracy (%).

Network	Method	CIFAR-10	CIFAR-100	Oxford Flowers-102	Oxford-IIIT Pets
4-bit ResNet-50	SGD	97.0±0.0	82.4±0.2	96.1±0.2	94.9±0.2
	SAQ (Ours)	**97.1±0.1**	**83.1±0.2**	**96.4±0.4**	**95.9±0.2**

6 Conclusion and Future Work

In this paper, we have devised a new training approach, called Sharpness-Aware Quantization (SAQ), to improve the generalization capability of the quantized models, which jointly performs compression (*i.e.*, quantization) and loss landscape smoothing for the first time. To this end, we have provided a unified view for the loss landscape smoothing of the quantized models by formulating quantization and SAM as introducing quantization noises and adversarial perturbations to the model weights. According to whether the quantization noises and adversarial perturbations are dependent on each other, we have split our SAQ into three cases. We then have fully studied and compared different cases. Extensive experiments on various datasets with different architectures including CNNs and Transformers have demonstrated that our proposed method consistently improves the performance of the quantized models and yields the SOTA uniform quantization results. In the future, we may extend our method to jointly perform pruning, quantization and loss landscape smoothing to obtain more compact models with better performance. We may also consider combining data selection or mixed-precision training to achieve better training efficiency.

REFERENCES

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael R Lyu, and Irwin King. Binarybert: Pushing the limit of bert quantization. In *ACL/IJCNLP*, 2021.

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 *CVPRW*, pp. 696–697, 2020.

Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *CVPR*, pp. 5918–5926, 2017.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pp. 213–229. Springer, 2020.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *ICLR*, 2017.

Peng Chen, Jing Liu, Bohan Zhuang, Mingkui Tan, and Chunhua Shen. Aqd: Towards accurate quantized object detection. In *CVPR*, pp. 104–113, 2021.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. In *ICLR*, 2022.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

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

Ruizhou Ding, Ting-Wu Chin, Zeye Liu, and Diana Marculescu. Regularizing activation distribution for training binarized deep networks. In *CVPR*, pp. 11408–11417, 2019.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *ICCV*, pp. 293–302, 2019.

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. In *ICLR*, 2021.

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Tan. Efficient sharpness-aware minimization for improved training of neural networks. In *ICLR*, 2022a.

Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent YF Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. *arXiv preprint arXiv:2205.14083*, 2022b.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *UAI*, 2017.

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

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021.

Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *ICCV*, pp. 4852–4861, 2019.

Tiantian Han, Dong Li, Ji Liu, Lu Tian, and Yi Shan. Improving low-precision network quantization via bin regularization. In *ICCV*, pp. 5261–5270, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. In *NeurIPS*, pp. 529–536, 1995.

Lu Hou and James T. Kwok. Loss-aware weight quantization of deep networks. In *ICLR*, 2018.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *NeurIPS*, 29, 2016.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *ICLR*, 2020.

Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *CVPR*, pp. 4350–4359, 2019.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

Dohyung Kim, Junghyup Lee, and Bumsub Ham. Distance-aware quantization. In *ICCV*, pp. 5271–5280, 2021a.

Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher sam: Information geometry and sharpness aware minimisation. In *ICML*, pp. 11148–11161, 2022.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. I-bert: Integer-only bert quantization. In *ICML*, pp. 5506–5518. PMLR, 2021b.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

Jungmin Kwon, Jeongseop Kim, Hyun-Seok Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *ICML*, 2021.

Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *CVPR*, pp. 6448–6457, 2021.

Hao Li, Zheng Xu, Gavin Taylor, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2018.

Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *ICLR*, 2020.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. In *ICLR*, 2021a.

Zhexin Li, Peisong Wang, Zhiyuan Wang, and Jian Cheng. Fixed-point quantization for vision transformer. In *CAC*, pp. 7282–7287. IEEE, 2021b.

Zhexin Li, Tong Yang, Peisong Wang, and Jian Cheng. Q-vit: Fully differentiable quantization for vision transformer. *arXiv preprint arXiv:2201.07703*, 2022.

Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *NeurIPS*, pp. 345–353, 2017.

Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. In *NeurIPS*, 2020.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *CVPR*, pp. 12360–12370, 2022a.

Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*, pp. 722–737, 2018.

Zechun Liu, Zhiqiang Shen, Shichao Li, Koen Helwegen, Dong Huang, and Kwang-Ting Cheng. How do adam and training strategies help bnns optimization. In *ICML*, volume 139, pp. 6936–6946, 2021a.

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 *CVPR*, pp. 4942–4952, 2022b.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *NeurIPS*, 34:28092–28103, 2021b.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *NeurIPS*, 2017.

Jeffrey L McKinstry, Steven K Esser, Rathinakumar Appuswamy, Deepika Bablani, John V Arthur, Izzet B Yildiz, and Dharmendra S Modha. Discovering low-precision networks close to full-precision networks for efficient inference. In *NIPSW*, pp. 6–9. IEEE, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *CVPR*, pp. 9078–9086, 2019.

Clara Na, Sanket Vaibhav Mehta, and Emma Strubell. Train flat, then compress: Sharpness-aware minimization learns more compressible models. *arXiv preprint arXiv:2205.12694*, 2022.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pp. 722–729. IEEE, 2008.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pp. 3498–3505. IEEE, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

Haotong Qin, Yifu Ding, Mingyuan Zhang, YAN Qinghua, Aishan Liu, Qingqing Dang, Ziwei Liu, and Xianglong Liu. Bibert: Accurate fully binarized bert. In *ICLR*, 2022.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, pp. 525–542, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pp. 4510–4520, 2018.

Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *ICLR*, 2018.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pp. 1–9, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019a.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *CVPR*, pp. 8612–8620, 2019b.

Longguang Wang, Xiaoyu Dong, Yingqian Wang, Li Liu, Wei An, and Yulan Guo. Learnable lookup table for neural network quantization. In *CVPR*, pp. 12423–12433, 2022.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *NeurIPS*, 33:2958–2969, 2020.

Kohei Yamamoto. Learnable companding quantization for accurate low-bit neural networks. In *CVPR*, pp. 5029–5038, 2021.

Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *CVPR*, 2019.

Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *ECCV*, pp. 365–382, 2018.

Xiandong Zhao, Ying Wang, Xuyi Cai, Cheng Liu, and Lei Zhang. Linear symmetric quantization of neural networks for low-precision integer hardware. In *ICLR*, 2020.

Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *CVPR*, pp. 8156–8165, 2021.

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.

Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Towards effective low-bitwidth convolutional neural networks. In *CVPR*, pp. 7920–7928, 2018.

Bohan Zhuang, Lingqiao Liu, Mingkui Tan, Chunhua Shen, and Ian Reid. Training quantized neural networks with a full-precision auxiliary module. In *CVPR*, pp. 1488–1497, 2020.

Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, James s Duncan, Ting Liu, et al. Surrogate gap minimization improves sharpness-aware training. In *ICLR*, 2022.

Appendix

A More implementation details

In this section, we provide more implementation details of SAQ. Following SAM (Foret et al., 2021) and GSAM (Zhuang et al., 2022), we apply m-sharpness strategy with $m = 128$. For both CNNs and ViTs, we use inception-style pre-processing (Szegedy et al., 2015) without strong data augmentation. Specifically, we randomly crop 224×224 patches from an image or its horizontal flip counterpart for training. At test time, a 224×224 centered crop is chosen. For the hyper-parameter ρ, we conduct grid search over $\{0.02, 0.05, 0.1, 0.15, 0.2, \ldots, 0.7\}$ to find appropriate values. We put the detailed settings of ρ in Table A. To compute the largest eigenvalue λ_{max} of the Hessian of different quantized models on ImageNet, we use the power iteration algorithm following (Dong et al., 2019). To reduce the computational cost, we randomly sample 10k training images for computation.

For the transfer learning experiments in Section 5.2, we train all models for 100 epochs. We use SGD with a momentum term of 0.9 for optimization. The learning rate is initialized to 0.01 and decreased to 0 using the cosine annealing. The mini-batch size and the weight decay are set to 64 and 0, respectively.

Table A: Hyper-parameter ρ for different quantized models on ImageNet.

Network	ResNet-18		ResNet-34		ResNet-50		MobileNetV2	ViT-S/32	ViT-S/16	ViT-B/32
Bitwidth	2	4	2	4	2	4	4	4	4	4
ρ	0.35	0.50	0.2	0.5	0.25	0.6	0.4	0.01	0.01	0.01

B SAQ vs. Train Flat and Then Quantize

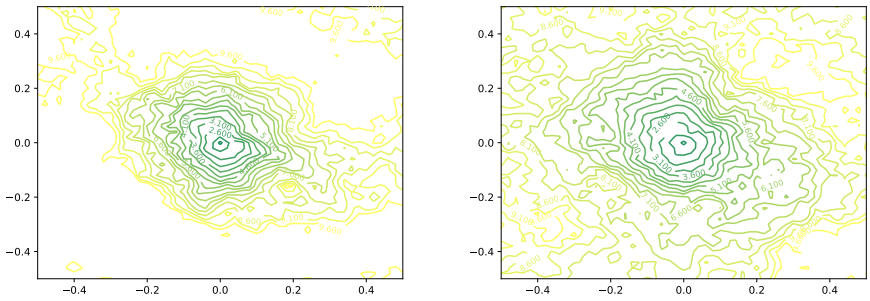
To further investigate the effectiveness of SAQ, we compare our method with "SAM $\rightarrow$ SGD" that first obtains a full-precision model with SAM and then trains a quantized model with SGD using full-precision model weights as initialization. We also include "SGD" that trains the quantized models with the vanilla SGD for comparisons. From Table B, SAM $\rightarrow$ SGD slightly improves the Top-1 accuracy (0.2%) over SGD at 2-bit quantization. We speculate that smoothing the loss landscape of the pre-trained models provides a better weight initialization for the quantized models. However, due to the large distribution gap between the quantized weights and full-precision weights, the performance gain of SAM $\rightarrow$ SGD over SGD is limited. Importantly, SAQ performs significantly better than SAM $\rightarrow$ SGD. For example, on ResNet-18, SAQ exceeds SAM $\rightarrow$ SGD by 0.5% on the Top-1 accuracy. These results show the superiority of jointly performing quantization and the loss landscape smoothing.

Table B: Performance comparisons of different methods on ImageNet. The Top-1 accuracy of the full-precision ResNet-18 and ResNet-34 with SAM are 70.9% and 74.4%.

Network	Method	Bitwidth (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5	Bitwidth (W/A)	Accuracy (%) Top-1	Accuracy (%) Top-5
ResNet-18 (FP: 70.7)	SGD	2/2	66.8	87.3	4/4	71.1	89.8
	SAM $\rightarrow$ SGD	2/2	67.0	87.5	4/4	71.1	89.9
	SAQ (Ours)	2/2	**67.5**	**87.5**	4/4	**71.6**	**90.1**
ResNet-34 (FP: 74.1)	SGD	2/2	71.4	90.2	4/4	74.4	91.9
	SAM $\rightarrow$ SGD	2/2	71.6	90.2	4/4	74.4	92.0
	SAQ (Ours)	2/2	**71.8**	**90.5**	4/4	**75.1**	**92.2**

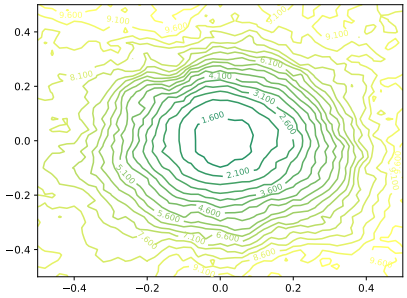
C Visualization of the loss landscapes

In this section, we show the loss landscape of different quantized models on ImageNet using the visualization method in (Li et al., 2018). We show the results in Figures A and B. The x- and y-

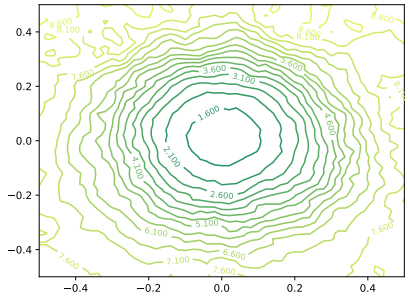


(a) 2-bit ResNet-18 obtained by SGD

(b) 2-bit ResNet-18 obtained by SAQ

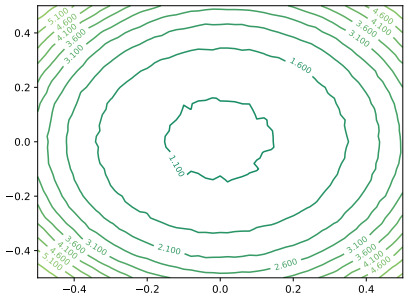


(c) 4-bit ResNet-18 obtained by SGD

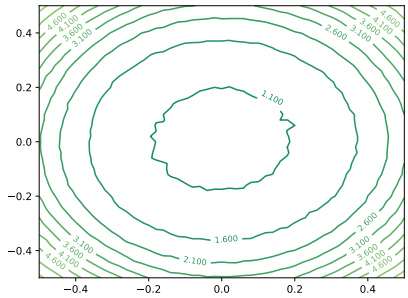


(d) 4-bit ResNet-18 obtained by SAQ

Figure A: The loss landscapes of the 2/4-bit ResNet-18 obtained by different methods on ImageNet.



(a) 4-bit ViT-B/32 obtained by SGD



(b) 4-bit ViT-B/32 obtained by SAQ

Figure B: The loss landscapes of the 4-bit ViT-B/32 obtained by different methods on ImageNet.

axes of the figures represent two randomly sampled orthogonal directions. From the results, the loss landscapes of the quantized models become smoother and flatter with the increase of bitwidth, suggesting that smoothing the loss landscapes of the 4-bit quantized models is easier than the 2-bit counterparts. Moreover, the loss landscapes of the quantized models obtained by SAQ are less chaotic and show larger contour interval compared with the SGD counterpart, indicating that SAQ is able to find flatter and smoother minima over SGD.