Self-Supervised Quantization-Aware Knowledge Distillation

Kaiqi Zhao
Arizona State University

Ming Zhao
Arizona State University

Abstract

Quantization-aware training (QAT) and Knowledge Distillation (KD) are combined to achieve competitive performance in creating low-bit deep learning models. However, existing works applying KD to QAT require tedious hyper-parameter tuning to balance the weights of different loss terms, assume the availability of labeled training data, and require complex, computationally intensive training procedures for good performance. To address these limitations, this paper proposes a novel Self-Supervised Quantization-Aware Knowledge Distillation (SQAKD) framework. SQAKD first unifies the forward and backward dynamics of various quantization functions, making it flexible for incorporating various QAT works. Then it formulates QAT as a co-optimization problem that simultaneously minimizes the KL-Loss between the full-precision and low-bit models for KD and the discretization error for quantization, without supervision from labels. A comprehensive evaluation shows that SQAKD substantially outperforms the state-of-the-art QAT and KD works for a variety of model architectures. Our code is at: `https://github.com/kaiqi123/SQAKD.git`.

1 Introduction

Deep neural networks (DNNs) have substantial computational and memory requirements. As the use of deep learning grows rapidly on a wide variety of Internet of Things and devices, the mismatch between resource-hungry DNNs and resource-constrained devices also becomes increasingly severe (Zhang et al., 2023; Zhao et al., 2023b). Quantization is one of the important model compression approaches to address this challenge by converting the full-precision model weights or activations to lower precision. In particular, Quantization-Aware Training (QAT) (Krishnamoorthi, 2018) has achieved promising results in creating low-bit models, which starts with a pre-trained model and performs quantization during retraining. However, most QAT works lead to considerable accuracy loss due to quantization, and no algorithm achieves consistent performance on every model architecture (e.g., VGG, ResNet, MobileNet) (Li et al., 2021b). Also, the various QAT works are motivated by different intuitions and lack a commonly agreed theory, which makes it challenging to generalize. Moreover, we empirically find that they do not work well on low-bit networks (1-3 bits). Therefore, we argue that there is a great need for a generalized, simple yet effective framework that is flexible to incorporate and improve various QAT algorithms for both low-bit and high-bit quantization.

Recent works apply Knowledge Distillation (KD) to QAT to mitigate the accuracy loss of the low-precision networks (termed "student") by transferring knowledge from full-precision or high-precision networks (termed "teacher") during training. However, these KD-applied QAT works 1) require tedious hyper-parameter tuning to balance the weights of different loss terms; 2) assume the availability of labeled training data which is in practice difficult and sometimes even infeasible to obtain; 3) require complex, computationally intensive training procedures for good performance (see Section 2 for details); and 4) focus narrowly on one specific KD approach alongside one particular quantizer which, as shown in our study, does not consistently perform well.

In this work, we propose a simple yet effective framework — Self-Supervised Quantization-Aware Knowledge Distillation (SQAKD). SQAKD first unifies the forward and backward dynamics of various quantization functions and shapes quantization-aware training as an optimization problem minimizing the discretization error between original weights/activations and their quantized counterparts. Then, we perform an in-depth analysis of the loss landscape of QAT where

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

the conventional training is replaced by KD. Our analysis reveals that CE-Loss (i.e., cross-entropy loss with labels) does not cooperate effectively with KL-Loss (i.e., the Kullback-Leibler divergence loss between the teacher's and student's penultimate outputs) and their combination may degrade the network performance. Thus, SQAKD proposes the novel formulation of QAT as a co-optimization problem that simultaneously minimizes the KL-Loss between the full-precision and low-bit models for KD and the discretization error for quantization without supervision from labels.

Compared to existing QAT methods and those that combine KD and QAT, the proposed SQAKD has several advantages. First, SQAKD is flexible for incorporating various QAT works by unifying the optimization of their forward and backward dynamics. Second, SQAKD improves the SOTA QAT works in both convergence speed and accuracy by using the full-precision teacher's help to guide gradient updates for low-bit weights, and it simultaneously minimizes the KL-Loss and the discretization error, making the estimated discrete gradients close to the continuous gradients. Third, SQAKD is hyperparameter-free, since it uses only the KL-Loss as the training loss and does not require the weight adjustment of multiple loss terms that appear in previous methods. Fourth, SQAKD operates in a self-supervised manner without labeled data, supporting a broad range of applications in practical scenarios. Finally, SQAKD offers simplicity in training procedures by requiring only one training phase to update the student, which reduces training cost and promotes usability and reproducibility.

Our comprehensive evaluation shows that SQAKD outperforms SOTA QAT and KD works significantly on various model architectures (VGG, ResNet, MobileNet-V2, ShuffleNet-V2, and SqueezeNet). First, compared to QAT, e.g., EWGS, PACT, LSQ, and DoReFa, SQAKD improves their convergence speed and top-1 accuracy (by up to 15.86%) for 1-8 bit quantization. Second, compared to 11 KD methods, SQAKD outperforms them by up to 17.09% on 1-bit VGG-13 with CIFAR-100. Third, compared to KD-integrated QAT, SQAKD achieves the smallest accuracy drop, outperforming the baselines by up to 3.06% on 2-bit ResNet-32 with CIFAR-100. Furthermore, on Jetson Nano hardware, SQAKD achieves an inference speedup of 3× for 8-bit quantization on Tiny-ImageNet.

In summary, our main contributions are summarized as: First, we are the first to 1) quantitatively investigate and benchmark 11 KD methods in the context of QAT, and 2) provide an in-depth analysis of the loss landscape of KD within QAT. Second, we propose a Self-Supervised Quantization-Aware Knowledge Distillation (SQAKD) method which operates in a self-supervised manner without labeled data and eliminates the need for hyper-parameter balancing. It is applicable to incorporate various quantizers, and it consistently outperforms state-of-the-art QAT, KD, and KD-integrated QAT methods on various models and datasets. Third, we open source all quantized networks, including those without any accuracy loss, such as 2-bit VGG-8, 4-bit ResNet-32, and 8-bit MobileNet-V2 that get 91.55%, 71.65%, and 58.13% in top-1 accuracy on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively. These low-precision networks are beneficial for diverse real-world applications.

2 Background and Related Works

Quantization. There are two main approaches to quantization. Post-Training Quantization (PTQ) (Li et al., 2021a) quantizes a pre-trained model without retraining and often leads to worse accuracy degradation than QAT (Krishnamoorthi, 2018), which performs quantization during retraining. This paper focuses on QAT.

Many QAT works emphasize designing the forward and backward propagation of the quantizer, the function that converts continuous weights or activations to discrete values. Early works such as BNN (Courbariaux et al., 2016) and XNOR-Net (Rastegari et al., 2016) employ channel-wise scaling in the forward pass, while DoReFa-Net (Zhou et al., 2016) introduces a universal scalar for all filters. Recent works utilize trainable parameters for the quantizer, improving control over facets such as clipping ranges (e.g., PACT (Choi et al., 2018), LSQ (Esser et al., 2019), APoT (Li et al., 2019), and DSQ (Gong et al., 2019)), i.e., the bounds where the input values are constrained, and quantization intervals (e.g., QIL (Jung et al., 2019) and EWGS (Lee et al., 2021)), i.e., the step size between adjacent quantization levels.

However, existing QAT methods lead to different levels of accuracy loss and are motivated by various heuristics, lacking a commonly agreed theory. Furthermore, MQbench (Li et al., 2021b) reveals that the difference between QAT algorithms is not as substantial as reported in their original papers; and no algorithm achieves the best performance on all architectures. Moreover, most QAT algorithms focus on high-bit networks (4 bits or more), underperforming on low-bit networks (see Section 4.1 for details). Therefore, this paper aims to close this gap, providing a generalized framework for integrating and improving various QAT works across both low-bit and high-bit precisions.

Knowledge Distillation (KD). KD transfers knowledge from large networks (termed "teacher") to im-

Table 1: Summary of related works that apply KD to QAT. The training cost is estimated by the sum of the initial training time for the pre-trained teachers ($N \times T_{pre}$), where N is the number of pre-trained teachers and T_{pre} is each teacher's training cost, and the training time during QAT with KD for both the teacher ($M_t \times T_t$) and the student ($M_s \times T_s$). Here, M_t and M_s denote the number of training phases for the teacher and student, respectively, while T_t and T_s represent their respective training costs per phase.

	Self-Supervised	Number of hyper-parameters for balancing loss terms	Number of training phases	Training cost	Initial teacher type	Train jointly / Train the student only
AP-SCHEME-A	No	3	1	$T_s + T_t$	One random initialized full-precision	Train jointly
AP-SCHEME-B	No	3	1	$T_{pre} + T_s$	One pre-trained full-precision	Train the student only
AP-SCHEME-C	No	3	2	$T_{pre} + 2T_s$	One pre-trained full-precision	Train the student only
QKD	No	2	3	$T_{pre} + 3T_s + T_t$	One pre-trained full-precision	Train jointly
CMT-KD	No	3	1	$NT_{pre} + T_s + T_t$	Multiple different bit-widths	Train jointly
SPEQ	No	1	1	$T_{pre} + T_s + T_t$	One quantized with target bit-width	Train jointly
PTG	No	2	4	$T_{pre} + 4T_s + T_t$	One pre-trained full-precision	Train jointly
SQAKD (Ours)	**Yes**	**0**	**1**	$\mathbf{T_{pre} + T_s}$	One pre-trained full-precision	Train the student only

prove the performance of small networks (termed "student"). Hinton et al. first proposed to transfer soft logits by minimizing the KL divergence between the teacher's and student's softmax outputs and the cross-entropy loss with data labels (Hinton et al., 2015). Later, some works (Tung and Mori, 2019; Peng et al., 2019; Zhao et al., 2023a) proposed to transfer various forms of intermediate representations, such as FSP matrix (Yim et al., 2017) and attention maps (Zagoruyko et al., 2016). However, despite the success of KD for image classification, the understanding of KD in model quantization is limited.

Knowledge Distillation + Quantization. Several recent works employ KD to mitigate accuracy loss from quantization, with the low-precision network as the student and the high- or full-precision network as the teacher. Mishra et al. proposed three schemes in Apprentice (AP) to improve the performance of ternary-precision or 4-bit networks. QKD (Kim et al., 2019) coordinates quantization and KD through three phases, including self-studying, co-studying, and tutoring. SPEQ (Boo et al., 2021) constructs a teacher using the student's parameters and applies stochastic bit precision to the teacher's activations. PTG (Zhuang et al., 2018) proposed a four-stage training strategy, emphasizing sequential optimization of quantized weights and activations, progressive reduction of bit width, and concurrent training of the teacher and the student. CMT-KD (Pham et al., 2023) encourages collaborative learning among multiple quantized teachers and mutual learning between teachers and the student.

We summarize the aforementioned works in Table 1. Despite their contributions, they have several limitations. First, they all require a delicate balance for the weights of different loss terms, which leads to time-consuming and error-prone hyperparameter tuning. For example, QKD requires three hyperparameters to balance the loss terms and their influence on performance is substantial. Second, they all assume that labels of the training data are always available during training, whereas in real-world scenarios access to labeled data is often limited and/or costly. Third, they require a complex, computationally expensive training process; for example, 3 training phases and 4 optimization strategies are required in QKD and PTG, respectively, costing training time and usability. Finally, these works focus on one specific KD approach alongside one particular quantizer, which cannot work well consistently for different applications on different hardware (Li et al., 2021b). This paper proposes a novel quantization-aware KD framework that eliminates hyperparameter balancing, operates in a self-supervised manner, supports diverse quantizers, and offers simplicity in training procedures with reduced training costs, as discussed in the rest of the paper.

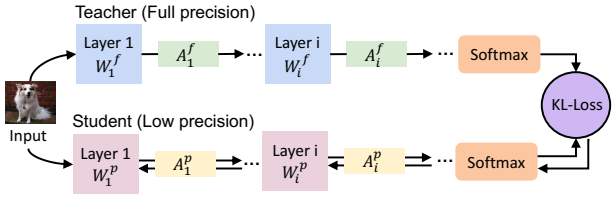


Figure 1: Workflow of SQAKD.

3 Methodology

Figure 1 illustrates the workflow of the proposed framework — Self-Supervised Quantization-Aware Knowledge Distillation (SQAKD).

3.1 QAT as Constrained Optimization

To provide a generalized theoretic framework, SQAKD first unifies the forward and backward dynamics of various quantizers and formulates QAT as an optimization problem.

Forward Propagation. Let us define $Quant(\cdot)$ as a

uniform quantizer that converts a full-precision input x to a quantized output $x_q = Quant(x)$. x can be the activations or weights of the network. First, the quantizer $Quant(\cdot)$ applies a clipping function $Clip(\cdot)$, which normalizes and restricts the full-precision input x to a limited range, producing a full-precision latent presentation x_c, as follows:

$$x_c = Clip(x, \{p_i\}_{i=1}^{i=K_c}, v, m), \tag{1}$$

where v and m are the lower and upper bounds of the range, respectively, $\{p_i\}_{i=1}^{i=K_c}$ denotes the set of trainable parameters needed for quantization, and K_c denotes the number of parameters. Note that different quantizers have different schemes for $Clip(\cdot)$. For example, in PACT, the lower bound v is set to 0 and the upper bound m is a trainable parameter optimized during training. The quantizer requires only one parameter, that is, $\{p_1|p_1 = m, K_c = 1\}$, and the clipping function is described as: $x_c = Clip(x, \{p_1|p_1 = m\}, 0, m) = 0.5(|x| - |x - m| + m)$. In EWGS, v and m are set to 0 and 1, respectively, and every quantized layer uses separate parameters (i.e. p_1 and p_2) for quantization intervals: $x_c = Clip(x, \{p_1, p_2\}, 0, 1) = clip(\frac{x-p_1}{p_1-p_2}, 0, 1)$.

Then, the quantizer $Quant(\cdot)$ converts the clipped value x_c to a discrete quantization point x_q using the function $R(\cdot)$ that contains a round function:

$$x_q = R(x_c, b, \{q_i\}_{i=1}^{i=K_r}), \tag{2}$$

where b is the bit width and $\{q_i\}_{i=1}^{i=K_r}$ denotes the set of trainable parameters. Note that $\{q_i\}_{i=1}^{i=K_r}$ is not necessary for some quantizers. For example, in EWGS, if activations are the input, $x_q = R(x_c, b) = \frac{round((2^b-1)\cdot x_c)}{2^b-1}$; and if weights are the input, $x_q = R(x_c, b) = 2(\frac{round((2^b-1)\cdot x_c)}{2^b-1} - 0.5)$. In some quantizers, like PACT and LSQ, the trainable parameters in the function $R(\cdot)$ are the same as those in the clipping function $Clip(\cdot)$, that is, $\{q_i|q_i = p_i\}_{i=1}^{i=K_r}$ and $K_r = K_c$.

In summary, the quantizer $Quant(\cdot)$ is described as: $x_q = Quant(x, \alpha, b, v, m)$, where α denotes a shorthand for the set of all the parameters in the functions $R(\cdot)$ and $Clip(\cdot)$: $\alpha = \{\{p_i\}_{i=1}^{i=K_c}, \{q_i\}_{i=1}^{i=K_r}\}$.

Backward Propagation. Directly training a quantized network using back-propagation is impossible since the quantizer $Q(\cdot)$ is non-differentiable. This issue arises due to the round function in Eq. 2, which produces near-zero derivatives almost everywhere. To solve this problem, most QAT works use Straight-Through Estimator (STE) (Bengio et al., 2013) to approximate the gradients: $\frac{\partial L}{\partial x_c} = \frac{\partial L}{\partial x_q}$. Instead of the commonly used STE for backpropagation, we propose a novel formula to integrate the discretization error $(x_c - x_q)$, representing the deviation between full-precision and its quantized weights/activations:

$$\frac{\partial L}{\partial x_c} = \frac{\partial L}{\partial x_q} + \mu \cdot (x_c - x_q), \tag{3}$$

with μ being a non-negative value. STE is represented by setting μ to zero, while EWGS is obtained when μ is the product of δ (a non-negative value), $sign(\frac{\partial L}{\partial x_q})$, and $\frac{\partial L}{\partial x_q}$. Note that μ can also be updated by other schemes, like Curriculum Learning driven strategy (Bengio et al., 2009; Zhao et al., 2022).

Optimization Objective. We frame QAT as an optimization problem minimizing both the discretization error as well as the discrepancy between model predictions and true labels. The goal is precise quantization without sacrificing predictive accuracy. Thus, the optimization objection is defined as follows:

$$\begin{aligned} &\min_{W_f, \alpha_W, \alpha_A} L(W_f) \\ s.t. \quad &W_q = Quant_W(W_f, \alpha_W, b_W, v_W, m_W) \\ &A_q = Quant_A(A_f, \alpha_A, b_A, v_A, m_A), \end{aligned} \tag{4}$$

where $Quant_W(\cdot)$ and $Quant_A(\cdot)$ are the quantizers for weights and activations, and W_f/A_f and W_q/A_q are the model's full-precision and quantized weights/activations. Note that $L(\cdot)$ can be the cross entropy loss with labels or other loss functions, e.g., distillation loss (see Sections 3.2 and 3.3 for details).

To the best of our knowledge, we are the first to generalize diverse quantizers, encompassing forward and backward propagations into a unified formulation of an optimization problem. This generalization and formulation enable our framework to be flexible in integrating various SOTA quantizers and effectively improve their performance (see Section 4.1 for the results).

3.2 Analysis of KD in QAT

Does KD perform well in QAT? Despite the wide use of KD, its effectiveness in addressing the QAT problem lacks a thorough study. To apply KD in QAT, we let a pre-trained full-precision network act as the teacher, guiding a low-bit student with the same architecture. Then, the training loss in Eq. 4 is defined as a linear combination of the cross-entropy loss with labels and the distillation loss between the teacher's and student's output distributions, controlled by a hyper-parameter λ:

$$L = (1 - \lambda)L_{CE} + \lambda L_{Distill}. \tag{5}$$

Note that the distillation loss $L_{Distill}$ can be one term, e.g., the KL divergence loss (Hinton et al., 2015), or multiple terms, e.g., the intermediate-presentation-based contrastive losses (Zhao et al., 2023a).

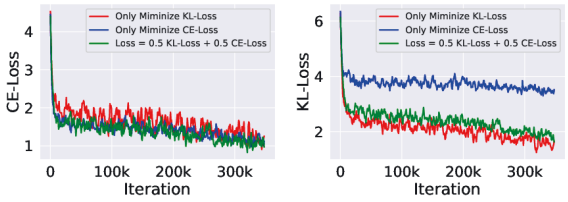
We hypothesize that existing KD methods, while effective in normal training scenarios, may not achieve satisfactory performance in QAT. This is because quantized networks exhibit lower representational

capacity compared to their full-precision counterparts (Nahshan et al., 2021), making it difficult to optimize multiple loss terms effectively. In addition, quantization introduces additional noise to the network weights or activations due to discretization. This noise can degrade the performance of the KD methods that rely on fine-grained information or target matching specific outputs. To validate this hypothesis, we comprehensively evaluate 11 KD methods in the context of quantization (see Section 4.2 for evaluation results). Notably, this evaluation represents the first attempt, to the best of our knowledge, to provide a thorough assessment of the performance of KD in addressing QAT problems.

Are both the cross-entropy loss and distillation loss necessary? To address the issue of KD methods underperforming in QAT, we analyze the significance of two loss components, shown in Eq. 5: the cross-entropy loss (termed CE-Loss) with labels and the KL divergence loss (termed KL-Loss) between the teacher's and student's penultimate outputs (before logits). We focus specifically on KL-Loss rather than other forms of distillation loss, as the conventional KD (Hinton et al., 2015), using KL-Loss, performs the best among existing KD works (Tian et al., 2019). CE-Loss quantifies the alignment between the low-bit network's predictions and the true labels, while KL-Loss measures the similarity between the low-bit network and the pre-trained full-precision network.

We consider three scenarios: 1) only minimizing KL-Loss by setting $\lambda = 1$ in Eq. 5, 2) only minimizing CE-Loss ($\lambda = 0$), and 3) jointly minimizing CE-Loss and KL-Loss with equal weights ($\lambda = 0.5$). Figure 2 illustrates the evolution of CE-Loss and KL-Loss in each iteration during the training of 1-bit VGG-13 on CIFAR-100. We observe Figure 2a that solely minimizing KL-Loss effectively leads to the concurrent minimization of both CE-Loss and KL-Loss. This indicates that the low-bit student can produce predictions close to the ground truth even without label supervision. In comparison, as shown in Figure 2b, solely minimizing CE-Loss or jointly minimizing CE-Loss and KL-Loss cannot achieves good minimization of KL-Loss. This indicates that integrating CE-Loss into the loss function, either partially ($\alpha = 0.5$) or entirely ($\alpha = 0$), negatively impacts the performance.

Therefore, we propose that solely minimizing KL-Loss is sufficient to achieve optimal gradient updates in the quantized network; CE-Loss does not cooperate effectively with KL-Loss, and their combination may actually degrade the network performance. Also, by excluding CE-Loss and considering only KL-Loss, the loss function becomes hyperparameter-free and thus eliminates the need for balancing the loss terms.



(a) CE-Loss (b) KL-Loss

Figure 2: Illustration of the evolution of (a) CE-Loss and (b) KL-Loss in each iteration during the training of 1-bit VGG-13 on CIFAR-100.

Table 2: Models and datasets.

Model	**ResNet** (He et al., 2016), **VGG** (Eigen and Fergus, 2015), **MobileNet** (Sandler et al., 2018), **ShuffleNet** (Zhang et al., 2018b), **SqueezeNet** (Iandola et al., 2016), **AlexNet** (Krizhevsky et al., 2012)
Dataset	**CIFAR-10** (Krizhevsky et al., 2009), **CIFAR-100** (Krizhevsky et al., 2009), **Tiny-ImageNet** (Le and Yang, 2015)

3.3 Optimization via Self-Supervised KD

As we drop the CE-Loss and keep only the KL-Loss in SQAKD, the optimization objection in Eq. 4 becomes:

$$\begin{aligned} \min_{W_f^S, \alpha_W, \alpha_A} \quad & KL(S(h^T/\rho)||S(h^S/\rho)) \\ s.t. \quad & W_q^S = Quant_W(W_f^S, \alpha_W, b_W, v_W, m_W) \\ & A_q^S = Quant_A(A_f^S, \alpha_A, b_A, v_A, m_A), \end{aligned} \tag{6}$$

where ρ is the temperature, which makes distribution softer for using the dark knowledge, Y is the ground-truth labels, and h^T and h^S are the penultimate layer outputs of the teacher and student, respectively. $Quant_W(\cdot)$ and $Quant_A(\cdot)$ are the quantization function for the student's weights and activations, and W_f^S/A_f^S and W_q^S/A_q^S are the student's full-precision weights/activities and quantized weights/activities.

During training, the teacher's weights are frozen and it performs the forward propagation only. In the forward pass, the student's parameters are quantized, while the corresponding full-precision values are preserved internally. During the backward propagation, the gradients are applied to the student's preserved full-precision values. After convergence, the student retains its full-precision weights and the parameters utilized in the quantizer. The student's quantized weights are obtained by applying the quantizer to the full-precision weights. The overall procedure of SQAKD is summarized in Algorithm 1.

4 Evaluation

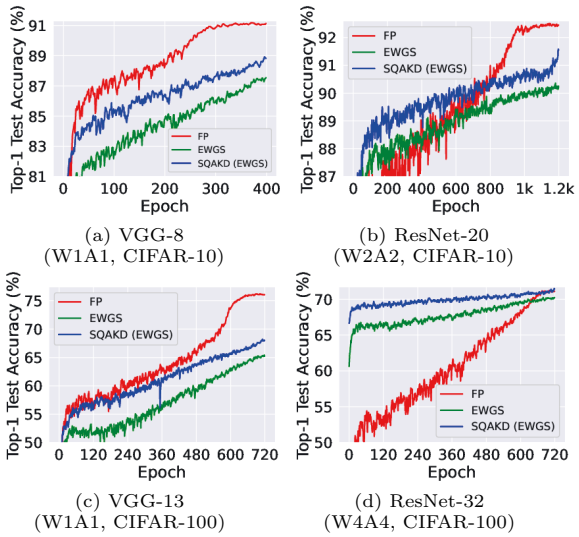
We conducted an extensive evaluation on diverse models and datasets, as listed in Table 2 (see Appendix for more details). We compared SQAKD with three groups of state-of-the-art (SOTA) methods: stan-

Algorithm 1 Self-Supervised Quantization-Aware Knowledge Distillation

Input: A pre-trained full-precision model F_f^T, and target bit-widths b_W for weights and b_A for activations.
Output: A quantized model F_q^S with specified bit-widths b_W for weights and b_A for activations.
Parameters: Lower and upper bounds v_W, m_W for weights, and v_A, m_A for activations; quantization function parameters $\alpha_W = \{\{p_{Wi}\}_{i=1}^{i=K_c}, \{q_{Wi}\}_{i=1}^{i=K_r}\}$ for weights and $\alpha_A = \{\{p_{Ai}\}_{i=1}^{i=K_c}, \{q_{Ai}\}_{i=1}^{i=K_r}\}$ for activations.
Hyperparameters: Temperature ρ
Initialization: Initialize the student with the pre-trained teacher: $F_f^S = F_f^T$
for each training iteration given every input X **do**
  Forward Propagation:
  Student: $h^S = F_f^S(X)$
  $s.t. \quad W_c = Clip_W(W_f^S, \{p_{Wi}\}_{i=1}^{i=K_c}, v_W, m_W) \quad (Eq.\ 1)$
  $W_q^S = R_W(W_c, b_W, \{q_{Wi}\}_{i=1}^{i=K_r}) \quad (Eq.\ 2)$
  $A_c = Clip_A(A_f^S, \{p_{Ai}\}_{i=1}^{i=K_c}, v_A, m_A) \quad (Eq.\ 1)$
  $A_q^S = R_A(A_c, b_A, \{q_{Ai}\}_{i=1}^{i=K_r}) \quad (Eq.\ 2)$
  Teacher: $h^T = F_f^T(X)$
  Minimize the loss function:
  $L = KL(S(h^T/\rho)||S(h^S/\rho))$ (Eq. 6)
  Backward Propagation:
  Calculate the gradient of discrete weights W_q via backpropagation: $\frac{\partial L}{\partial W_q}$.
  Calculate the gradient of latent values W_c: $\frac{\partial L}{\partial W_c} = \frac{\partial L}{\partial W_q} + \mu \cdot (W_c - W_q)$ (Eq. 3).
  Propagate the gradient to the input.
end for

Table 3: Baselines.

Group	Baselines
QAT	**PACT** (Choi et al., 2018), **LSQ** (Esser et al., 2019), **DoReFa** (Zhou et al., 2016), **EWGS** (Lee et al., 2021)
KD	**SP** (Tung and Mori, 2019), **AT** (Zagoruyko et al., 2016), **FitNet** (Romero et al., 2014), **CC** (Peng et al., 2019), **VID** (Ahn et al., 2019), **RKD** (Park et al., 2019), **AB** (Heo et al., 2019), **FT** (Kim et al., 2018), **FSP** (Yim et al., 2017), **NST** (Huang and Wang, 2017), **CRD** (Tian et al., 2019), **CKTF** (Zhao et al., 2023a)
KD+QAT	**SPEQ** (Boo et al., 2021), **PTG** (Zhuang et al., 2018) **QKD** (Kim et al., 2019), **CMT-KD** (Pham et al., 2023)



(a) VGG-8 (W1A1, CIFAR-10)
(b) ResNet-20 (W2A2, CIFAR-10)
(c) VGG-13 (W1A1, CIFAR-100)
(d) ResNet-32 (W4A4, CIFAR-100)

Figure 3: Top-1 test accuracy evolution of full-precision (FP) and quantized models using the standalone EWES and SQAKD integrating EWGS, during training.

dalone QAT, KD, and KD-integrated QAT, as listed in Table 3.

We implemented SQAKD on PyTorch version 1.10.0 and Python version 3.9.7. We used four Nvidia RTX 2080 GPUs for model training and conduct inference experiments on Jetson Nano using NVIDIA TensorRT. Please see Appendix for more implementation details.

4.1 Improvements on SOTA QAT Methods

Results on CIFAR-10 and CIFAR-100. Table 4 presents the top-1 test accuracy of ResNet and VGG models on CIFAR-10 and CIFAR-100, where activations and weights are quantized to 1, 2, and 4 bits using 1) the standalone QAT method, i.e., EWGS, and 2) SQAKD that incorporates EWGS, i.e., SQAKD (EWGS).

SQAKD significantly improves the accuracy of EWGS in all bit quantization scenarios. Specifically, on CIFAR-10, SQAKD improves EWGS by 0.36% to 1.28% on VGG-8 and 0.05% to 0.39% on ResNet-20; On CIFAR-100, the improvement is 1.26% to 3.01% on VGG-13 and 0.16% to 1.15% on ResNet-32. Furthermore, SQAKD produces quantized models with higher accuracy than full-precision models. For example, on ResNet-32 with CIFAR-100, a 4-bit model trained with SQAKD outperforms the full-precision model by 0.32%. Such small and accurate models are useful for many real-world applications for edge deployment. In comparison, the standalone EWGS leads to different levels of accuracy drop from 0.18% to 6.16% in all quantization scenarios.

Figure 3 illustrates the top-1 test accuracy evolution for full-precision and quantized models in each epoch during training on CIFAR-10 and CIFAR-100. Compared to the standalone EWGS, SQAKD significantly accelerates the convergence speed of 1-bit VGG-8 and 2-bit ResNet-20 with CIFAR-10, as well as 1-bit VGG-13 and 4-bit ResNet-32 with CIFAR-100. These results confirm that SQAKD improves EWGS in both converge speed and final accuracy.

Results on Tiny-ImageNet. As shown in Table 5, SQAKD consistently improves the top-1 accuracy of various QAT methods, including PACT, LSQ, and DoReFa, by a large margin for 3, 4, and 8-bit quan-

Table 4: Top-1 test accuracy (%) on CIFAR-10 and CIFAR-100. "FP" denotes the full-precision model's accuracy. "W∗A×" denotes that the weights and activations are quantized into ∗bit and ×bit, respectively. The number inside the parenthesis is the improvement compared to EWGS.

Dataset	CIFAR-10						CIFAR-100					
Model	VGG-8 (FP: 91.27)			ResNet-20 (FP: 92.58)			VGG-13 (FP: 76.36)			ResNet-32 (FP: 71.33)		
Bit-width	W1A1	W2A2	W4A4	W1A1	W2A2	W4A4	W1A1	W2A2	W4A4	W1A1	W2A2	W4A4
EWGS	87.77	90.84	90.95	86.42	91.41	92.40	65.55	73.31	73.41	59.25	69.37	70.50
SQAKD (EWGS)	**89.05** (+1.28)	**91.55** (+0.71)	**91.31** (+0.36)	**86.47** (+0.05)	**91.80** (+0.39)	**92.59** (+0.19)	**68.56** (+3.01)	**74.65** (+1.34)	**74.67** (+1.26)	**59.41** (+0.16)	**69.99** (+0.62)	**71.65** (+1.15)

Table 5: Top-1 test accuracy (%) of ResNet-18 and VGG-11 on Tiny-ImageNet.

Model	ResNet-18 (FP: 65.59)			VGG-11 (FP: 59.47)		
Bit-width	W3A3	W4A4	W8A8	W3A3	W4A4	W8A8
PACT	58.09	61.06	64.91	52.94	57.10	58.08
SQAKD (PACT)	**61.34** (+3.25)	**61.47** (+0.41)	**65.78** (+0.87)	**57.25** (+4.31)	**59.05** (+1.95)	**59.44** (+1.36)
LSQ	61.99	64.10	65.08	58.39	59.14	59.25
SQAKD (LSQ)	**65.21** (+3.22)	**65.34** (+1.24)	**65.96** (+0.88)	**58.43** (+0.04)	**59.19** (+0.05)	**59.42** (+0.17)
DoReFa	61.94	62.72	63.23	56.72	57.28	57.54
SQAKD (DoReFa)	**64.10** (+2.16)	**64.56** (+1.84)	**64.88** (+1.65)	**57.02** (+0.3)	**58.93** (+1.65)	**58.91** (+1.37)

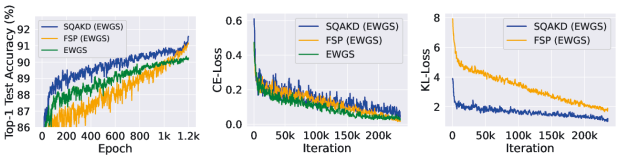
Table 6: Top-1 and top-5 test accuracy (%) of MobileNet-V2, ShuffleNet-V2, and SqueezeNet on Tiny-ImageNet.

Model	Bit-width	Method	Top-1 Acc.		Top-5 Acc.	
Mobile Net-V2	FP	-	58.07		80.97	
	W3A3	PACT	47.77		73.44	
		SQAKD(PACT)	**52.73**	(+4.96)	**77.68**	(+4.24)
	W4A4	PACT	50.33		75.08	
		SQAKD(PACT)	**57.14**	(+6.81)	**80.61**	(+5.53)
	W8A8	DoReFa	56.26		79.64	
		SQAKD(DoReFa)	**58.13**	(+1.87)	**81.3**	(+1.66)
Shuffle Net-V2	FP	-	49.91		76.05	
	W4A4	PACT	27.09		52.54	
		SQAKD(PACT)	**41.11**	(+14.02)	**68.4**	(+15.86)
	W8A8	DoReFa	45.96		71.93	
		SQAKD(DoReFa)	**47.33**	(+1.37)	**73.85**	(+1.92)
Squee zeNet	FP	-	51.49		76.02	
	W4A4	LSQ	35.37		62.75	
		SQAKD(LSQ)	**47.40**	(+12.03)	**73.18**	(+10.43)
	W8A8	DoReFa	42.66		69.25	
		SQAKD(DoReFa)	**46.62**	(+3.96)	**73.02**	(+3.77)

tization on Tiny-ImageNet. For example, on 3-bit ResNet-18, SQAKD improves PACT by 4.31%. We observe that SQAKD achieves a more substantial improvement on low-bit quantization than on high-bit quantization. For example, on ResNet-18, the improvements achieved by SQAKD using LSQ exhibit an escalating trend, measured at 0.88%, 1.24%, and 3.22% respectively, as the bit width decreases from 8 to 4 and further to 3 bits. Similar trends are observed in DoReFa with ResNet-18 and PACT with VGG-11. This is because low-bit quantization causes more significant information loss, leading to reduced accuracy. The distillation process in SQAKD mitigates this by leveraging insights from the teacher to guide the low-bit weight gradients.

Table 7: Top-1 test accuracy of KD in QAT, using EWGS as the quantizer. Blue and red numbers inside the parenthesis denote the increase and decrease compared to the standalone EWGS, respectively.

Dataset	CIFAR-10	CIFAR-100
Model	ResNet-20 (FP: 92.58)	VGG-13 (FP: 76.36)
Bit-width	W2A2	W1A1
EWGS (Lee et al., 2021)	91.41	65.55
CRD (Tian et al., 2019)	91.36 (-0.05)	68.47 (+2.92)
AT (Zagoruyko et al., 2016)	89.99 (-1.42)	67.56 (+2.01)
NST (Huang and Wang, 2017)	89.07 (-2.34)	66.04 (+0.49)
SP (Tung and Mori, 2019)	90.92 (-0.49)	51.47 (-14.08)
RKD (Park et al., 2019)	90.91 (-0.50)	67.23 (+1.68)
FitNet (Romero et al., 2014)	90.21 (-1.20)	64.25 (-1.30)
CC (Peng et al., 2019)	91.31 (-0.10)	65.60 (+0.05)
VID (Ahn et al., 2019)	88.40 (-3.01)	65.88 (+0.33)
FSP (Yim et al., 2017)	91.44 (+0.03)	65.28 (-0.27)
FT (Kim et al., 2018)	90.08 (-1.33)	67.33 (+1.78)
CKTF (Zhao et al., 2023a)	90.76 (-0.65)	67.92 (+2.37)
SQAKD	**91.80** (+0.39)	**68.56** (+3.01)



(a) Test Accuracy (b) CE-Loss (c) KL-Loss

Figure 4: (a) Top-1 test accuracy, (b) CE-Loss, and (c) KL-Loss of EWGS, FSP, and SQAKD in each epoch/iteration during training on 2-bit ResNet-20 with CIFAR-10.

SQAKD can also effectively quantize models (MobileNet, ShuffleNet, and SqueezeNet) that are already compact for use on resource-constrained edge devices, as shown in Table 6. For example, on 4-bit quantization, for the top-1 and top-5 test accuracy, SQAKD improves 1) PACT by 14.02% and 15.86%, respectively, on ShuffleNet-V2; and 2) LSQ by 12.03% and 10.43%, respectively, on SqueezeNet. On MobileNet-V2, for 8-bit quantization, SQAKD enables this already lightweight model to outperform the full-precision model by 0.06% in top-1 and 0.33% in top-5 accuracy; and with 4-bit quantization, the accuracy drop achieved by SQAKD is within 1%, whereas PACT causes a significant accuracy drop of 7.74%.

4.2 Comparison with SOTA KD Methods

Table 7 presents the top-1 test accuracy of the proposed SQAKD and 11 existing KD methods in the

context of quantization (EWGS) for 2-bit ResNet-20 on CIFAR-10 and 1-bit VGG-13 on CIFAR-100. We find out, in the context of quantization, the 11 existing KD methods cannot converge without the supervision from ground-truth labels. So we compare the *supervised* existing KD works for quantization with the proposed *unsupervised* SQAKD in Table 7. The results show that *unsupervised* SQAKD outperforms the *supervised* KD methods by 0.36% to 3.4% on CIFAR-10 and 0.09% to 17.09% on CIFAR-100.

Compared to the standalone EWGS, SQAKD is 0.39% better on CIFAR-10 and 3.01% better on CIFAR-100. In comparison, none of the existing KD works can consistently improve EWGS on both CIFAR-10 and CIFAR-100, and none can outperform SQAKD.

Also, as observed, compared to EWGS, SQAKD performs better on CIFAR-100 than on CIFAR-10 (3.01% v.s. 0.39%). This could be because CIFAR-100 is more complicated than CIFAR-10 (with more classes and data), resulting in more lost information and a higher accuracy drop during quantization. SQAKD is good at compensating for the lost information and recovering the accuracy by transferring knowledge from the full-precision teacher.

We provide a detailed comparison of SQAKD to FSP with EWGS as the quantizer, as FSP is the second-best in accuracy (shown in Table 7). Figure 4a shows that on 2-bit ResNet-20 with CIFAR-10, SQAKD converges much faster than the standalone EWGS and FSP, whereas FSP decreases the convergence speed of the standalone EWGS. Figures 4b and 4c show that, for CE-Loss, *unsupervised* SQAKD achieves a comparable performance to *supervised* EWGS and *supervised* FSP; for KL-Loss, SQAKD converges much faster and achieves a lower final value than FSP. This validates that SQAKD is sufficient for minimizing both CE-Loss and KL-Loss, leading to faster and better distillation.

The above results confirm that, in the context of quantization, *self-supervised* SQAKD outperforms existing *supervised* KD works in both convergence speed and final accuracy.

4.3 Comparison with SOTA Methods Applying both QAT and KD

Table 8 shows the comparison between QAKD and SOTA KD-integrated QAT methods that utilize both quantization and KD, on CIFAR-10 and CIFAR-100. We measure the accuracy drop by comparing the top-1 test accuracy of the low-bit model with that of its corresponding full-precision model. The results of the related KD-integrated QAT works are directly obtained from their original papers, and the unavailable data is marked with '-'.

Table 8: Comparison of SQAKD with KD-integrated QAT methods. Accuracy drop is measured by comparing the top-1 test accuracy of low-bit models with their full-precision counterparts. Negative values indicate an accuracy drop, while positive values denote an increase.

Dataset	CIFAR-10	CIFAR-100		
Model	ResNet-20	ResNet-32	AlexNet	
Bit-width	W2A2	W2A2	W2A2	W4A4
QKD	-1.10	-4.40	-	-
CMT-KD	-	-	-0.30	-
SPEQ	-0.70	-	-	-
PTG	-	-	+0.80	+0.40
SQAKD	**-0.66**	**-1.34**	**+1.04**	**+2.29**

Table 9: Inference on Jetson Nano with Tiny-ImageNet.

Model	Bit-width	Throughput (fps)	Inference Time (s)	Speedup
ResNet-18	FP32	1.8688	0.5351	-
	INT8	5.7925	0.1726	3.10×
MobileNet-V2	FP32	3.0905	0.3236	-
	INT8	9.4004	0.1064	3.04×
ShuffleNet-V2	FP32	12.6402	0.0791	-
	INT8	36.7817	0.0272	2.91×
SqueezeNet1_0	FP32	2.8105	0.3558	-
	INT8	8.7907	0.1138	3.13×

SQAKD consistently outperforms all the baselines in all cases. Specifically, for 2-bit ResNet models (ResNet-20 on CIFAR-10 and ResNet-32 on CIFAR-100), all the methods cause an accuracy drop while SQAKD achieves the smallest drop, lower than baselines by 0.04% to 3.06%. For AlexNet on CIFAR-100, only SQAKD and PTG achieve higher accuracy than their full-precision model whereas the improvement achieved by SQAKD outperforms those of PTG, e.g., 1.04% v.s. 0.8% for 2-bit quantization and 2.29% v.s. 0.4% for 4-bit quantization. In contrast, CMT-KD results in a 0.3% accuracy degradation.

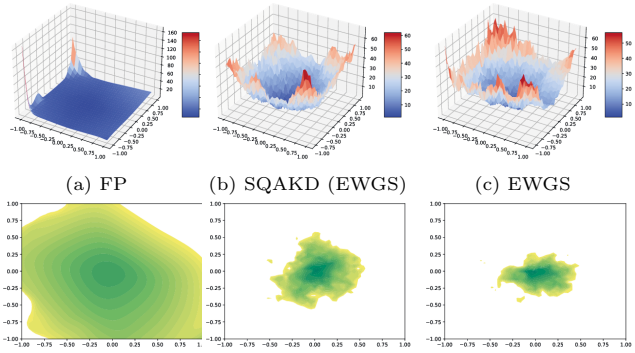
These baseline methods require a substantial amount of labeled data, intricate hyper-parameter tuning to balance the different loss terms, e.g., three parameters required in the loss function of CMT-KD, and complex training procedures, e.g., QKD with three-phase training. In contrast, SQAKD is self-supervised, does not require hyperparameter adjustments to balance the loss terms, and employs a simple-yet-effective single-phase training process. These characteristics make SQAKD a more practical and efficient approach compared to the SOTA works.

4.4 Inference Speedup

The reduction in model complexity in bit width that SQAKD achieves does translate to real speedup in model inference. We conduct inference experiments using the PyTorch framework and NVIDIA TensorRT on Jetson Nano, a widely used Internet-of-Things (IoT) platform. Table 9 shows that our method improves the inference speed by about 3× for 8-bit

Table 10: Top-1 test accuracy of different quantization forward and backward combinations.

Model and Dataset	Method	Forward	Backward	Acc.
ShuffleNet-V2 (W4A4, Tiny-ImageNet)	FP	-	-	49.91
	PACT	PACT	STE	27.09
	SQAKD (PACT)	PACT	STE	41.11
			EWGS	**41.88**
ResNet-20 (W2A2, CIFAR-10)	FP	-	-	92.58
	EWGS	EWGS	EWGS	91.41
	SQAKD (EWGS)	EWGS	STE	91.70
			EWGS	**91.80**



(a) FP (b) SQAKD (EWGS) (c) EWGS

(d) FP (e) SQAKD (EWGS) (f) EWGS

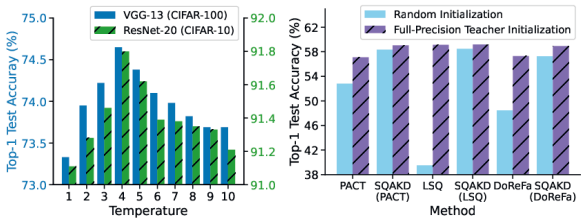
Figure 5: 3D loss surface (a, b, c) and 2D contours (d, e, f) for full-precision and 2-bit ResNet-20 using SQAKD and EWGS on CIFAR-10.

quantization using various model architectures, including ResNet-18, MobileNet-V2, ShuffleNet-V2, and SqueezeNet, on Tiny-ImageNet.

5 Ablation Study

Analysis of Loss Surface. Figure 10 compares the 3D loss surface and the corresponding 2D contour of the full-precision ResNet-20 and the 2-bit ResNet-20 trained using SQAKD and the standalone EWGS on CIFAR-10. SQAKD enables the quantized model to achieve a flatter and smoother loss surface compared to using the standalone EWGS. To the best of our knowledge, we are the first to visualize and analyze the loss surface of a low-precision model achieved using QAT.

Flexibility for various forward and backward combinations. To show the flexibility of our framework, we modularly integrate forward and backward techniques of SOTA QAT methods, like PACT and EWGS, into SQAKD. Table 10 shows that, on 4-bit ShuffleNet-V2 with Tiny-ImageNet, SQAKD enhances PACT by 14.02% using STE backward and further improves it by 0.77% using EWGS backward. The success of the EWGS backward results from its integration of discretization error into gradient approximation. On 2-bit ResNet-20 with CIFAR-10, SQAKD improves EWGS by 0.29% and 0.39% using STE and EWGS backward, respectively. We are the first to decouple QAT's forward and backward techniques and to provide a detailed performance analysis.



(a) Temperature (b) Initialization

Figure 6: Effect of (a) temperature (W2A2), and (b) initialization (W4A4, VGG-11, Tiny-ImageNet).

Effect of Temperature. Temperature ρ (in Eq. 6) softens the distribution, helping extract the teacher's dark knowledge. We conducted an experimental investigation over the range $\rho \in [1, 10]$ using VGG-13 on CIFAR-100 and ResNet-20 on CIFAR-10. Figure 6a shows that $\rho = 4$ yields the best performance. These findings align with the empirical insights in contrastive knowledge transfer (Tian et al., 2019; Zhao et al., 2023a).

Effect of Initialization. Figure 6b shows that, with the same setting, initialized either randomly or with the full-precision teacher, SQAKD increases the top-1 test accuracy of PACT, LSQ, and DoReFa by different levels (from 0.05% to 18.97%) for 4-bit VGG-11 on Tiny-ImageNet. Furthermore, the full-precision teacher initialization outperforms the random initialization in all cases.

6 Conclusion

This paper proposes a Self-Supervised Quantization-Aware Knowledge Distillation (SQAKD) framework for model quantization in image classification. SQAKD establishes stronger baselines and does not require labeled training data, potentially making QAT research more accessible. It also provides a generalized agreement to existing QAT methods by unifying the optimization of their forward and backward dynamics. SQAKD is hyperparameter-free and requires simple training procedures and low training costs, making it a highly practical and usable solution. We quantitatively investigate and benchmark 11 KD methods in the context of QAT, providing a novel perspective on KD beyond its typical applications. Our results confirm that SQAKD consistently outperforms the existing KD and QAT approaches by a large margin.

7 Acknowledgments

We thank the anonymous reviewers for their feedback. This work is supported by National Science Foundation awards CNS-2311026, SES-2231874, OAC-2126291 and CNS-1955593.

References

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

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

Yoonho Boo, Sungho Shin, Jungwook Choi, and Wonyong Sung. Stochastic precision ensemble: self-knowledge distillation for quantized deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6794–6802, 2021.

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.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4852–4861, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019.

Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4350–4359, 2019.

Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31, 2018.

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

Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

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

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

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6448–6457, 2021.

Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. *arXiv preprint arXiv:1909.13144*, 2019.

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. *arXiv preprint arXiv:2102.05426*, 2021a.

Yuhang Li, Mingzhu Shen, Jian Ma, Yan Ren, Mingxin Zhao, Qi Zhang, Ruihao Gong, Fengwei Yu, and Junjie Yan. Mqbench: Towards reproducible and deployable model quantization benchmark. *arXiv preprint arXiv:2111.03759*, 2021b.

Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, 110(11-12):3245–3262, 2021.

NVIDIA. Tensorrt: A c++ library for high performance inference on nvidia gpus and deep learning accelerators, 2021. URL `https://github.com/NVIDIA/TensorRT`. Accessed: 2021-04-27.

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*, pages 3967–3976, 2019.

Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5007–5016, 2019.

Cuong Pham, Tuan Hoang, and Thanh-Toan Do. Collaborative multi-teacher knowledge distillation for learning low bit-width deep neural networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6435–6443, 2023.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

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

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

Sergey Zagoruyko, Nikos Komodakis, and xxx. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

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)*, pages 365–382, 2018a.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018b.

Zhaofeng Zhang, Sheng Yue, and Junshan Zhang. Towards resource-efficient edge ai: From federated learning to semi-supervised model personalization. *IEEE Transactions on Mobile Computing*, 2023.

Kaiqi Zhao, Hieu Duy Nguyen, Animesh Jain, Nathan Susanj, Athanasios Mouchtaris, Lokesh Gupta, and Ming Zhao. Knowledge distillation via module replacing for automatic speech recognition with recurrent neural network transducer. 2022.

Kaiqi Zhao, Yitao Chen, and Ming Zhao. A contrastive knowledge transfer framework for model compression and transfer learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023a.

Kaiqi Zhao, Animesh Jain, and Ming Zhao. Automatic attention pruning: Improving and automat-

ing model pruning using attentions. In *International Conference on Artificial Intelligence and Statistics*, pages 10470–10486. PMLR, 2023b.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7920–7928, 2018.

Self-Supervised Quantization-Aware Knowledge Distillation: Supplementary Materials

1 Experiment Setup

1.1 Models and Datasets

We conducted extensive experiments to evaluate our proposed Self-Supervised Quantization-Aware Knowledge Distillation (SQAKD) framework on various image classification datasets, including 1) CIFAR-10 (Krizhevsky et al., 2009) which contains 50K (32×32) RGB training images, belonging to 10 classes, 2) CIFAR-100 (Krizhevsky et al., 2009) which consists of 50K 32×32 RGB training images with 100 classes, and 3) Tiny-ImageNet (Le and Yang, 2015) containing 100K 64×64 training images with 200 classes.

We utilized the widely used baseline architectures, ResNet (He et al., 2016) and VGG (Eigen and Fergus, 2015). We also select lightweight architectures, including MobileNet (Sandler et al., 2018), ShuffleNet (Zhang et al., 2018b), and SqueezeNet (Iandola et al., 2016). We initialize the model weights using its corresponding pre-trained, full-precision counterparts. Following the commonly used experimental settings (Li et al., 2021b; Lee et al., 2021; Rastegari et al., 2016; Zhang et al., 2018a), we do not quantize the first convolutional layer and the last fully-connected layer.

1.2 Implementation Details

We used four NVIDIA RTX 2080 GPUs to train the models. The implementation details are shown in Table 11. On Tiny-ImageNet, the learning rate rises up to 5e-4 linearly in the first 2500 iterations and then decays to 0.0 via cosine annealing; On CIFAR-10 and CIFAR-100, the learning rate starts with 1e-3 and 5e-4, respectively, and decays to 0.0 using a cosine annealing schedule. For all training images, we apply basic data augmentation, including random cropping, horizontal flipping, and normalization. Specifically, the training images are cropped to 224×224 for Tiny-ImageNet and 32×32 for CIFAR-10/CIFAR-100.

We conducted inference experiments on Jetson Nano using NVIDIA TensorRT (NVIDIA., 2021). We first converted the model from PyTorch to ONNX, then transformed the ONNX representation into a TensorRT engine file, and finally deployed it using the TensorRT runtime. The NVIDIA Jetson Nano is a compact computing platform designed for edge computing and artificial intelligence (AI) applications. Built on NVIDIA's advanced GPU architecture, the Jetson Nano provides computing capabilities in a small footprint, making it ideal for deploying AI models in embedded systems and low-power environments. NVIDIA TensorRT is a high-performance deep learning inference library, which supports deep learning applications in environments where computational resources are constrained.

Table 11: Implementation details.

Dataset	Model	Initial learning rate	Training epochs	Batch size	Optimizer	Weight decay
CIFAR-10	VGG-8	1.00E-03	400	256	Adam	1.00E-04
	ResNet-20	1.00E-03	1200	256	Adam	1.00E-04
CIFAR-100	VGG-13	5.00E-04	720	64	Adam	5.00E-04
	ResNet-32	5.00E-04	720	64	Adam	5.00E-04
Tiny-ImageNet	ResNet-18	5.00E-04	100	64	SGD	5.00E-04
	VGG-11	5.00E-04	100	32	SGD	5.00E-04
	MobileNet-V2	5.00E-04	100	32	SGD	5.00E-04
	ShuffleNet-V2	5.00E-04	100	64	SGD	5.00E-04
	SqueezeNet	5.00E-04	100	64	SGD	5.00E-04

1.3 Hyper-parameters of baselines

For a fair comparison of state-of-the-art (SOTA) QAT methods, we reran their original open-source code, achieving results consistent with those in the respective papers. Specifically, we used the original EWGS code (Lee et al., 2021) and the MQbench code (Li et al., 2021b) (a commonly used baseline code for QAT works) for PACT, LSQ, and DoReFa, as their original code is not available. For the related 11 KD baselines, we implemented them using the CRD code (Tian et al., 2019), which is a widely used baseline code and has been confirmed with the authors of those KD works. We also strictly used the exactly same hyperparameters as those reported in CRD (Tian et al., 2019) and the original papers of KD works. For example, we implemented SOTA KD methods in the context of QAT, and their loss functions are represented as: $L = L_{CE} + \lambda L_{Distill}$, where L_{CE} is the cross-entropy loss with labels, $L_{Distill}$ is the distillation loss for transferring knowledge between the teacher and the student, and λ controls the weights of the loss terms. Note that different KD methods have different forms of $L_{Distill}$, and $L_{Distill}$ can contain multiple loss terms. For example, CKTF contains two loss terms in $L_{Distill}$, while VID and FSP set the number of loss terms the same as the number of intermediate layer pairs between the teacher and student. The hyper-parameter settings λ are the same as those used in the original papers, shown in Table 12.

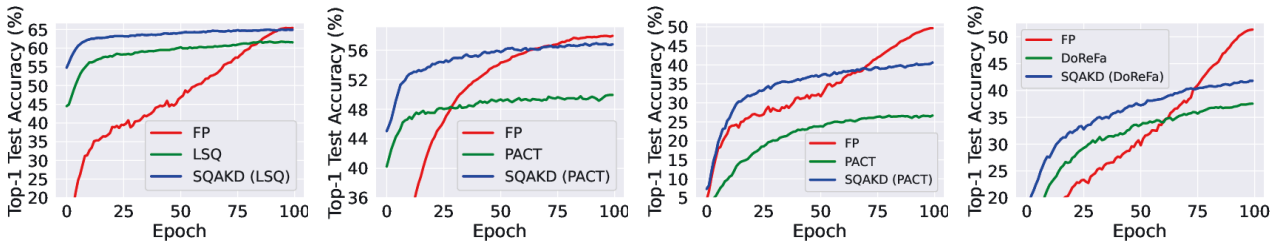
Table 12: Hyper-parameters of related KD methods

Method	CRD	AT	NST	SP	RKD	FitNet	CC	VID	FSP	FT	CKTF
λ	0.8	1000	50	3000	1.0	100	0.02	1.0	50	200	0.8

2 Evaluation

2.1 Improvements on SOTA QAT Methods

Results on Tiny-ImageNet. Figure 7 shows that SQAKD improves the convergence speed of all the baselines on ResNet-18, MobileNet-V2, ShuffleNet-V2 and SqueezeNet, with Tiny-ImageNet. Furthermore, on MobileNet-V2, SQAKD enables the quantized student to converge much faster than the full-precision teacher, whereas the standalone QAT methods cannot achieve that.



(a) ResNet-18 (W3A3) (b) MobileNet-V2 (W4A4) (c) ShuffleNet-V2 (W4A4) (d) SqueezeNet (W4A4)

Figure 7: Top-1 test accuracy evolution of full-precision models (FP), and quantized models using SQAKD and the standalone quantization methods, including LSQ, PACT, and DoReFa, in each epoch during training on Tiny-ImageNet.

Results on CIFAR-100. We observe that the accuracy improvement achieved by SQAKD is higher on VGG models than on ResNet models with the same level of quantization. For example, for 1-bit quantization, on CIFAR-100, the accuracy improvement on VGG-13 is 3.01%, which is much higher than 0.16% on ResNet-32; and on CIFAR-10, the accuracy improvement of 1.28% on VGG-8 outperforms that of 0.05% on ResNet-20. This might be because VGG models have a simpler, more explicit, and more structured hierarchy of features than ResNet models. The hierarchical structure in VGG models, with sequential convolutional layers, helps the student to learn representations at different levels of abstraction and to recover the accuracy loss caused by quantization. In comparison, the skip connections in ResNet architectures cause irregular, nonlinear pathways, complicating the knowledge transfer process.

2.2 Comparison with SOTA KD Methods

We observe that, with supervision by labels, the KD works that transfer structural knowledge of outputs perform better than those that transfer conditionally independent outputs. For example, on CIFAR-100, with EWGS as the quantizer, SP (Tung and Mori, 2019) and FitNet (Romero et al., 2014) underperform the standalone EWGS. In contrast, CRD (Tian et al., 2019), RKD (Park et al., 2019), and CKTF (Zhao et al., 2023a), which capture the structural relations of intermediate representations (CKTF) or penultimate outputs (CRD and RKD), perform well. This might be because, by capturing the correlations of high-order output dependencies from the full-precision model, it effectively restores the lost information resulting from quantization and guides the gradient updates of the low-bit model. Although KD has been studied in the existing works, to the best of our knowledge, we are the first to study the performance of various KD methods in the context of quantization.

3 Ablation study

3.1 Effect of Training Time

Motivated by the proposition that prolonged training increases test accuracy (He et al., 2019), we investigated its applicability to QAT. Figure 8 illustrates that for 2-bit VGG-13 and ResNet-20 on CIFAR-10, regardless of the approach — full precision (FP), EWGS, and SQAKD, a 1200-epoch training span consistently outperforms 400 epochs, with accuracy gains from 0.28% to 0.77%. Also, given equal training epochs, SQAKD yields consistent improvements over EWGS, from 0.2% to 0.61% for 400 epochs and 0.03% to 0.31% for 1200 epochs.

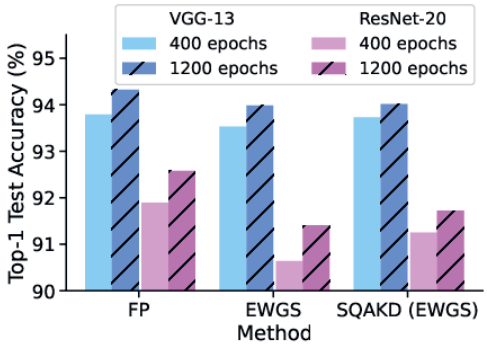


Figure 8: Effect of training Time (W2A2, CIFAR-10).

3.2 Analysis of CE-Loss and KL-Loss

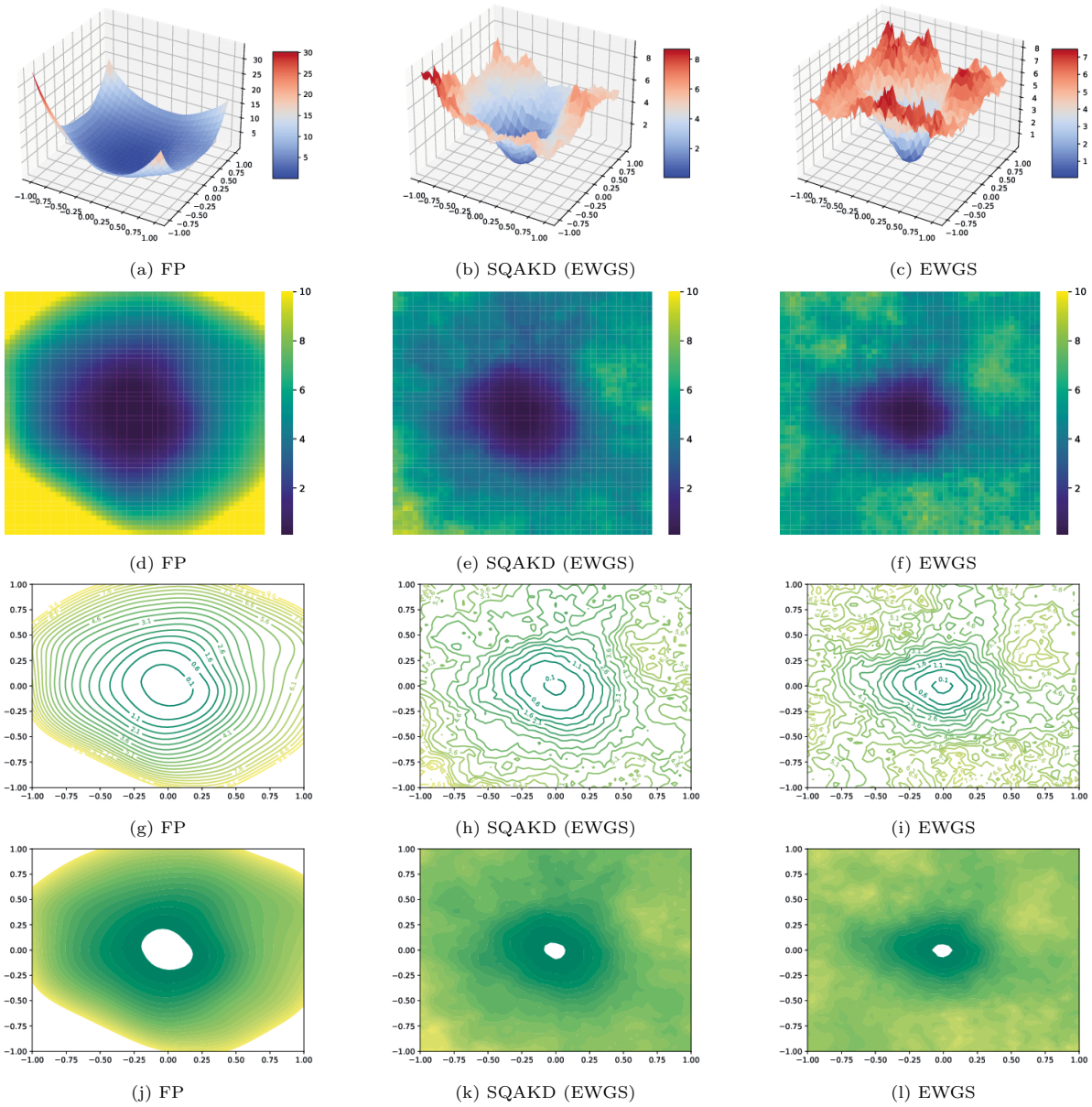
In addition to the three scenarios shown in Figure 2 where $\lambda \in \{0.0, 0.5, 1.0\}$ (Eq. 5), we extend our analysis by considering $\lambda \in \{0.0, 0.1, 0.2, 0.3, \ldots, 0.9, 1.0\}$ on 1-bit VGG-13 with CIFAR-100 to further investigate the relationship between the cross-entropy loss (CE-Loss) and the KL-divergence loss (KL-Loss). Figure 9 shows that solely minimizing KL-Loss ($\lambda = 1.0$) effectively leads to the concurrent minimization of both CE-Loss and KL-Loss. These results validate that solely minimizing KL-Loss is sufficient to achieve optimal gradient updates in the quantized network.

(a) CE-Loss (b) KL-Loss

Figure 9: The evolution of (a) CE-Loss and (b) KL-Loss during the training of 1-bit VGG-13 on CIFAR-100, with $\lambda \in \{0.0, 0.1, \ldots, 1.0\}$. When $\lambda = 0.0$, only CE-Loss is minimized; when $\lambda = 1.0$, only KL-Loss is minimized.

3.3 Analysis of Loss Surface

Figure 10 shows the 3D loss surface and the corresponding 2D heat, 2D contour, and 2D filled contour visualizations of the full-precision VGG-8 and the 2-bit VGG-8 trained using SQAKD and the standalone EWGS on CIFAR-10. SQAKD enables the quantized model to achieve a flatter and smoother loss surface compared to using the standalone EWGS.



(a) FP (b) SQAKD (EWGS) (c) EWGS

(d) FP (e) SQAKD (EWGS) (f) EWGS

(g) FP (h) SQAKD (EWGS) (i) EWGS

(j) FP (k) SQAKD (EWGS) (l) EWGS

Figure 10: 3D loss surface (a, b, c), and its corresponding 2D heat representations (d, e, f), 2D contour visualizations (g, h, i), and 2D filled contour visualizations (j, k, l) for full-precision (FP), the standalone EWSG, and SQAKD integrating EWSG, on 2-bit VGG-8 with CIFAR-10.