SQuaT: Self-Supervised Knowledge Distillation via Student-Aware Quantized Teacher Features

HyeonJun Lee* **Hyeonsik Jo*** **Jinwoo Chung** **Jangho Kim†**
Kookmin University
{lbghj522, hsjo, imaboybut, jangho.kim}@kookmin.ac.kr

Abstract

Quantization-Aware Training (QAT) enables the deployment of quantized models with minimal accuracy degradation. However, in practical scenarios, training labels are often unavailable due to privacy, copyright, or cost constraints. Knowledge Distillation (KD) is a common approach to address this challenge, but we observe that prior work combining QAT with KD suffers from a fundamental limitation: during distillation, the range mismatch between the teacher and the quantized student model induces an unattainable residual, resulting in an irreducible lower bound on the distillation loss. Motivated by this observation, we propose **SQuaT** (**S**tudent-Aware **Qua**ntized **T**eacher Features), a label-free QAT framework with KD that theoretically eliminates this lower bound by applying the student's quantization parameters to quantize the teacher's features during distillation. Through comprehensive experiments across diverse settings, we demonstrate that SQuaT consistently outperforms strong baselines, with particularly pronounced gains in extreme low-bit (e.g., 1- and 2-bit) settings. Furthermore, extensive evaluations across various model design choices show that our approach does not rely on specific architectural assumptions, making it broadly applicable across diverse architectures and quantization settings. The source code is available at `https://github.com/lcdbsa522/SQuaT`.

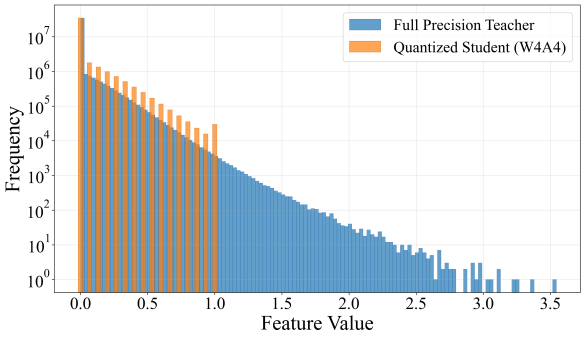


Figure 1: Distribution of feature maps for the FP teacher and 4-bit student in ResNet-20 on CIFAR-10.

1 INTRODUCTION

Over the past decade, Deep Neural Networks (DNNs) have emerged as a dominant modeling paradigm in artificial intelligence, achieving remarkable performance across a wide range of domains, including computer vision and natural language processing. However, these advances have increasingly relied on larger model sizes, which incur substantial memory and computational costs at inference time, thereby posing practical challenges for deploying models in resource-constrained environments such as mobile and edge devices (Han et al., 2015; Kim et al., 2023; Back et al., 2025). Quantization is a widely adopted model compression technique that significantly reduces inference-time computation and memory usage by mapping high-precision floating-point weights and activations to low-precision integer values. In particular, Quantization-Aware Training (QAT) has been widely adopted as an effective strategy to mitigate the accuracy degradation of full-precision (FP) models by simulating quantization effects during training (Jacob et al., 2018; Nagel et al., 2021). Nevertheless, existing QAT methods typically assume access to both the weights of an FP model and a sufficiently large amount of labeled training data. In practice, while the weights of pretrained FP models are often readily available, training labels

*Equal contribution, †Corresponding author. Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

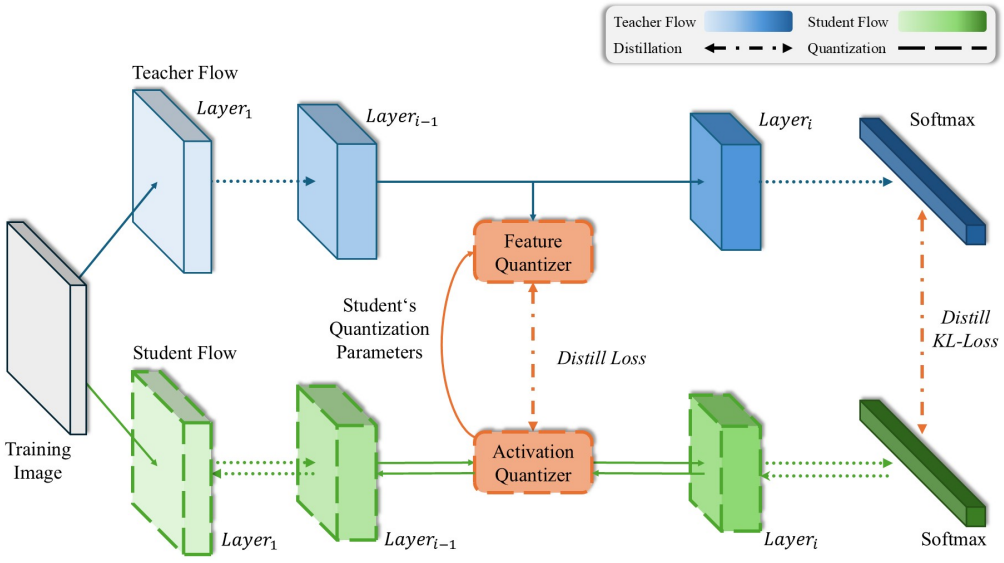


Figure 2: Overall framework of SQuaT. By projecting teacher features onto the student's quantization lattice via a student-aware projection using the student's quantization parameters, SQuaT reduces the distributional discrepancy between the teacher's full-precision features and the student's low-precision features, thereby enabling effective quantization-aware training without labels.

are frequently inaccessible due to privacy, copyright, and cost constraints. These practical limitations naturally motivate Knowledge Distillation (KD)-based approaches (Hinton et al., 2015), in which a pretrained FP model serves as a teacher model to guide a quantized student model. Accordingly, the central question addressed in this work is as follows: *Can we perform effective and stable QAT relying solely on the teacher's knowledge, without access to labeled data?*

Self-Supervised Quantization-Aware Knowledge Distillation (SQAKD) (Zhao and Zhao, 2024) demonstrated the feasibility of label-free QAT by relying exclusively on logit-level distillation during QAT training. However, this approach does not fully leverage the teacher's intermediate representations, which contain substantially richer information than logits alone. Indeed, Self-Supervised Learning methods such as SimCLR (Chen et al., 2020) and BYOL (Grill et al., 2020) show that feature-level learning enables powerful representation learning.

However, directly applying feature-level knowledge distillation to a quantized student model in this manner suffers from a fundamental limitation. As illustrated in Figure 1, the intermediate feature maps of a FP teacher model and a low-bit quantized student model have markedly different value ranges. In other words, the representation manifolds of the FP teacher and the low-bit quantized student are inherently mismatched. While the student's outputs are constrained to lie on a quantization lattice defined by a clipping range and step size determined by the target bit-level, the teacher produces unconstrained full-precision features. Consequently, using FP features as distillation targets for the student induces an unattainable residual between the teacher and the student, resulting in a theoretically irreducible lower bound on the distillation loss. This lower bound forces the student to continually pursue targets that it cannot attain, leading to distorted gradient signals and ultimately unstable optimization.

In a related line of work, Quantized Feature Distillation (QFD) (Zhu et al., 2023), proposed in a supervised learning setting, attempts to perform feature-level knowledge distillation tailored to quantized models. QFD introduces a feature quantizer into the full-precision teacher model prior to QAT and calibrates it through a few epochs of pretraining—typically corresponding to a small fraction of the total QAT training—in order to preserve the teacher's performance. During subsequent QAT, the calibrated quantizer is used to quantize the teacher's features to the same target bit-level as the student, enabling feature distillation. However, this approach introduces additional overhead, as it requires additional pretraining of a teacher-side quantizer. More importantly, because the teacher is calibrated independently of the student's quantization characteristics, the quantization lattice of the teacher does not align with that of the student, even at the same bit-level. As a result, the semantic

meaning of quantized values may differ between the teacher and the student.

To address this fundamental limitation, we propose SQuaT (Student-Aware Quantized Teacher Features). The core idea of SQuaT is to quantize the intermediate features of the teacher model using the quantization parameters of the student model during distillation, thereby projecting the teacher's features directly into the student's quantized representation space. This student-aware projection aligns the teacher and student features on the same quantization lattice and enables feature-level knowledge distillation using only targets that are attainable by the student. As a result, the unattainable residual induced by FP targets that the student cannot represent is eliminated, thereby removing the theoretical lower bound on the distillation loss. We theoretically analyze and formally prove the removal of this lower bound in Section 3.

Furthermore, by directly reusing the student model's quantization parameters, SQuaT ensures that the teacher and student features are interpreted consistently within the student's quantized representation space. In other words, the semantic meaning of values represented on the quantization lattice is preserved across the two models from the student's perspective. Figure 2 provides an overview of the complete SQuaT distillation framework.

While prior feature-level knowledge distillation approaches have relied on teacher-aware projection, which maps the student toward the teacher's representation space, we instead introduce a paradigm shift to student-aware projection, in which the teacher is projected into the student's representation space. Through this shift, SQuaT enables stable and effective feature-level knowledge distillation in label-free QAT settings.

Our main contributions are summarized as follows:

- We propose SQuaT, which introduces a paradigm shift from teacher-aware to student-aware projection by leveraging the student model's quantization parameters.
- Through theoretical and empirical analyses, we show that SQuaT effectively removes the lower bound caused by unattainable residuals in prior teacher-aware projection-based methods.
- We conduct extensive experiments to validate the proposed approach, demonstrating that SQuaT is agnostic to model design choices and broadly applicable, with particularly pronounced gains in extreme low-bit settings.

2 RELATED WORK

2.1 Quantization-Aware Training

Quantization-Aware Training (QAT) simulates quantization-induced errors during training, enabling models to adapt to low-precision computation environments (Jacob et al., 2018; Nagel et al., 2021). In the forward pass, weights and activations are quantized for computation, while gradients are computed with respect to the underlying full-precision (FP) weights in the backward pass. To enable gradient propagation, approximation techniques such as the Straight-Through Estimator (STE) (Bengio et al., 2013) have been widely adopted; however, their limitations have also been extensively discussed in prior work (Bai et al., 2018; Chen et al., 2019; Gong et al., 2019). To improve the stability and accuracy of QAT, various methods have been proposed (Choi et al., 2018; Li et al., 2019; Uhlich et al., 2019). Among them, Learned Step Size Quantization (LSQ) introduces trainable step sizes (Esser et al., 2019), while Element-Wise Gradient Scaling (EWGS) leverages gradient scaling to alleviate STE-induced instability (Lee et al., 2021). These advances have enabled QAT to be extended to extreme low-bit settings with improved stability and accuracy.

2.2 Knowledge Distillation for QAT

Knowledge distillation (KD) is a widely adopted approach for mitigating performance degradation in low-bit QAT settings. Quantized Feature Distillation (QFD) (Zhu et al., 2023) performs feature-level KD in supervised QAT environments. Specifically, prior to QAT, feature quantizers are inserted into intermediate layers of the teacher network and trained to compensate for quantization loss. During the subsequent QAT stage, these calibrated quantizers convert teacher features to the target bit-width, and distillation is performed on the quantized representations. Although this design attempts to align the distillation signal with the low-bit representation space, it introduces additional training overhead due to the calibration of teacher-side quantizers. Moreover, since the teacher's quantization parameters are optimized independently of the student's, their quantization lattices may not coincide even at the same bit-width, potentially resulting in misalignment between teacher and student representations.

2.3 Self-Supervised Learning for QAT

As labeled data become increasingly difficult to obtain due to privacy and cost constraints, self-supervised learning has attracted increasing attention. Self-

Supervised Quantization-Aware Knowledge Distillation (SQAKD) (Zhao and Zhao, 2024) demonstrated the feasibility of label-free QAT through logit-level distillation. However, it relies solely on output probability distributions and does not fully exploit the rich intermediate representations of the teacher model. Self-supervised learning methods such as SimCLR (Chen et al., 2020) and BYOL (Grill et al., 2020) have shown that feature-level learning plays a crucial role in learning discriminative representations. This suggests that incorporating feature-level guidance, in addition to logit-level supervision, may help mitigate representation distortion caused by quantization in QAT settings.

3 PROPOSED FRAMEWORK

In this section, we first review the formulation of QAT, and then introduce SQuaT, a self-supervised feature distillation framework. The central idea of SQuaT is to project teacher features onto the student's quantization lattice, thereby enabling joint alignment of feature- and logit-level signals within a unified representation space.

3.1 Quantization-Aware Training

We adopt the EWGS method (Lee et al., 2021), which directly learns quantization parameters rather than relying on the simple STE-based QAT formulation (Bengio et al., 2013). We now describe the forward and backward quantization procedures used in SQuaT.

3.1.1 Forward pass

Given an input tensor x (representing either weights or activations), a fixed global bit-width b, and learnable normalization range parameters (a^-, a^+), the forward quantization proceeds as follows. First, normalize to $[0, 1]$:

$$x_{\text{norm}} = \text{clip}\left(\frac{x - a^-}{a^+ - a^-},\, 0,\, 1\right). \tag{1}$$

Project onto the b-bit uniform grid:

$$x_{\text{quant}} = \frac{\text{round}\big(x_{\text{norm}} \cdot (2^b - 1)\big)}{2^b - 1} \;\in\; \Big\{0, \frac{1}{2^b - 1}, \ldots, 1\Big\}. \tag{2}$$

Apply a type-specific linear mapping (as in EWGS) to obtain the quantized output:

$$\widehat{x}_{\phi_\ell} = \begin{cases} x_{\text{quant}}, & \text{activations}, \\ 2(x_{\text{quant}} - 0.5)\,, & \text{weights}. \end{cases} \tag{3}$$

Here, ϕ_ℓ denotes the quantization parameters for layer ℓ, including the range (a_ℓ^-, a_ℓ^+) and the bit-width b_ℓ, i.e., $\phi_\ell = \{a_\ell^-, a_\ell^+, b_\ell\}$. These parameters can be defined either per layer or per channel.

3.1.2 Backward pass

Quantization involves non-differentiable operations (e.g., `round`, `clip`). Following EWGS, gradients are rescaled based on the quantization error and upstream signals. Let $x_{\text{norm},\ell}$ be the normalized value, $x_{\text{quant},\ell}$ the discretized value at layer ℓ, $e_\ell = x_{\text{norm},\ell} - x_{\text{quant},\ell}$ the quantization error, and $g_\ell = \partial\mathcal{L}/\partial\widehat{x}_{\phi_\ell}$ the upstream gradient. Then

$$\begin{aligned} g_{q,\ell} = \frac{\partial \mathcal{L}}{\partial x_{\text{quant},\ell}} &= \frac{\partial \mathcal{L}}{\partial \widehat{x}_{\phi_\ell}} \cdot \frac{\partial \widehat{x}_{\phi_\ell}}{\partial x_{\text{quant},\ell}} \\ &= \begin{cases} g_\ell, & \text{activations}, \\ 2g_\ell, & \text{weights}, \end{cases} \end{aligned} \tag{4}$$

$$\frac{\partial \mathcal{L}}{\partial x_{\text{norm},\ell}} \;\approx\; g_{q,\ell}(1 + \eta \cdot \text{sign}(g_{q,\ell}) \cdot e_\ell)\,, \tag{5}$$

where η denotes a small scaling hyperparameter.

3.2 SQuaT

For each layer ℓ, let $f_{S,\ell}(\cdot; \theta_S, \phi_{S,\ell})$ and $f_{T,\ell}(\cdot)$ denote the student and teacher feature extractors at layer ℓ, respectively, where $\phi_{S,\ell} = \{a_\ell^-, a_\ell^+, b_\ell\}$ are the student's quantization parameters for that layer.

3.2.1 Student-Aware Projection

We define the student-aware projection at layer ℓ by

$$\Pi_{\phi_{S,\ell}}(z) \;\triangleq\; \widehat{z}_{\phi_{S,\ell}}, \tag{6}$$

where $\widehat{z}_{\phi_{S,\ell}}$ is obtained from the student's forward quantization path at layer ℓ (normalize $\rightarrow$ grid projection $\rightarrow$ type-specific mapping), as defined in Section 3.1.1.

We project teacher features onto the student's quantization lattice via a student-aware projection, yielding student-aware quantized teacher features. The student is then trained to match these projected targets. Let $\mathcal{K}$ be the set of distillation layers. Define $\tilde{f}_{T,\ell}(x) \triangleq \Pi_{\phi_{S,\ell}}(f_{T,\ell}(x))$ and optimize the following objective:

$$\min_{\theta_S, \{\phi_{S,\ell}\}} \; \mathbb{E}_{x\sim\mathcal{D}} \frac{1}{|\mathcal{K}|} \sum_{\ell\in\mathcal{K}} \big\| \widehat{f}_{S,\ell}(x; \theta_S, \phi_{S,\ell}) - \tilde{f}_{T,\ell}(x) \big\|_2^2, \tag{7}$$

where the student-side feature corresponds to the actual (already quantized) student output:

$$\widehat{f}_{S,\ell}(x; \theta_S, \phi_{S,\ell}) \;\triangleq\; f_{S,\ell}(x; \theta_S, \phi_{S,\ell}). \tag{8}$$

3.2.2 Self-Supervised Knowledge Distillation

SQuaT performs label-free joint alignment at both the feature and logit levels. These objectives are complementary: the former preserves spatial/channel patterns, while the latter maintains class-level semantics.

Feature-level alignment. For a set of distillation layers $\mathcal{K}$, we align quantized student features $\widehat{f}_{S,\ell}$ with SQuaT targets $\tilde{f}_{T,\ell}$:

$$\mathcal{L}_{\text{feat}} \;=\; \frac{1}{|\mathcal{K}|}\sum_{\ell\in\mathcal{K}}\Big\|\widehat{f}_{S,\ell}(x;\theta_S,\phi_{S,\ell}) - \tilde{f}_{T,\ell}(x)\Big\|_2^2, \quad (9)$$

Logit-level alignment. We smooth predictions with temperature $T_{kd}>0$ and apply KL divergence:

$$p_T = \text{softmax}\Big(\frac{z_T}{T_{kd}}\Big), \quad p_S = \text{softmax}\Big(\frac{z_S}{T_{kd}}\Big), \quad (10)$$

$$\mathcal{L}_{\text{logit}} \;=\; T_{kd}^2 \cdot \text{KL}(p_T \,\|\, p_S)\,, \quad (11)$$

where p_T serves as a pseudo-label distribution that preserves inter-class relationships and mitigates semantic drift that may arise from feature-only alignment.

Joint objective. By replacing the unattainable target $f_{T,\ell}$ with the attainable $\tilde{f}_{T,\ell}$, SQuaT minimizes

$$\mathcal{L}_{\text{SQuaT}} \;=\; \lambda_{\text{feat}}\,\mathcal{L}_{\text{feat}} \;+\; \lambda_{\text{logit}}\,\mathcal{L}_{\text{logit}}, \quad (12)$$

where $\lambda_{\text{feat}}, \lambda_{\text{logit}} \geq 0$ balance feature- and logit-level objectives, and the feature targets are constructed via the student-aware projection.

The overall process of SQuaT is depicted in Algorithm 1. During training, the teacher network T remains fixed, and only the student parameters θ_S and quantization parameters ϕ_S are updated.

Algorithm 1 SQuaT: Self-Supervised Knowledge Distillation via Student-Aware Quantized Teacher Features

Require: Frozen teacher $f_T(\cdot;\theta_T)$, quantized student $f_S(\cdot;\theta_S,\{\phi_{S,\ell}\}_{\ell\in L})$, unlabeled data D, distillation layers $\mathcal{K}$, temperature T_{kd}, loss weights $(\lambda_{\text{feat}},\lambda_{\text{logit}})$, step size η

Ensure: Updated student parameters $(\theta_S,\{\phi_{S,\ell}\}_{\ell\in L})$

Notation: $\tilde{f}_{T,\ell}(x) \triangleq \Pi_{\phi_{S,\ell}}(f_{T,\ell}(x))$, where $\Pi_{\phi_{S,\ell}}$ projects the teacher features onto the student's quantization lattice via a student-aware projection.

for each training step $t = 1, 2, \ldots$ **do**
  Sample a mini-batch $x \sim D$
  for each $\ell \in \mathcal{K}$ **do**
    $\tilde{f}_{T,\ell}(x) \leftarrow \Pi_{\phi_{S,\ell}}\big(f_{T,\ell}(x)\big)$
  end for
  $\mathcal{L}_{\text{feat}} \leftarrow \frac{1}{|\mathcal{K}|}\sum_{\ell\in\mathcal{K}}\Big\|\widehat{f}_{S,\ell}(x;\theta_S,\phi_{S,\ell}) - \tilde{f}_{T,\ell}(x)\Big\|_2^2$
  $\mathcal{L}_{\text{logit}} \leftarrow T_{\text{kd}}^2 \cdot \text{KL}\big(p_T \,\|\, p_S\big)$
  $\mathcal{L} \leftarrow \lambda_{\text{feat}}\,\mathcal{L}_{\text{feat}} + \lambda_{\text{logit}}\,\mathcal{L}_{\text{logit}}$
  $\theta_S \leftarrow \theta_S - \eta\,\nabla_{\theta_S}\mathcal{L}$
  $\phi_{S,\ell} \leftarrow \phi_{S,\ell} - \eta\,\nabla_{\phi_{S,\ell}}\mathcal{L} \quad \forall \ell \in L$
end for

3.3 Analysis of Student-Aware Projection

Theorem 1. *Fix a layer ℓ. Let $\Pi_{\phi_{S,\ell}} : \mathbb{R}^{d_\ell} \to \mathbb{R}^{d_\ell}$ denote the student-aware projection (the student's forward quantization path at layer ℓ), and let $\mathcal{G}_{\phi_{S,\ell}} \triangleq \text{Im}(\Pi_{\phi_{S,\ell}})$ be the set of outputs attainable by the student at this layer. For the teacher feature $f_{T,\ell}$ (random vector induced by $x\sim\mathcal{D}$), the following hold:*

$$\text{(a)} \quad \inf_{h\in\mathcal{G}_{\phi_{S,\ell}}} \mathbb{E}\big\|h - f_{T,\ell}\big\|_2^2 \;=\; \mathbb{E}\big\|\Pi_{\phi_{S,\ell}}(f_{T,\ell}) - f_{T,\ell}\big\|_2^2, \quad (13)$$

$$\text{(b)} \quad \inf_{h\in\mathcal{G}_{\phi_{S,\ell}}} \mathbb{E}\big\|h - \Pi_{\phi_{S,\ell}}(f_{T,\ell})\big\|_2^2 \;=\; 0. \quad (14)$$

Proof. We first establish the result pointwise and subsequently take expectations.

(a) For any realization of $f_{T,\ell}$, consider the least-squares problem $\min_{h\in\mathcal{G}_{\phi_{S,\ell}}} \|h - f_{T,\ell}\|_2^2$. By definition, $\mathcal{G}_{\phi_{S,\ell}}$ is precisely the range of the student's forward quantization map $\Pi_{\phi_{S,\ell}}$; therefore, the nearest attainable point to $f_{T,\ell}$ is its image under this mapping, $\Pi_{\phi_{S,\ell}}(f_{T,\ell})$. Hence

$$\inf_{h\in\mathcal{G}_{\phi_{S,\ell}}} \|h - f_{T,\ell}\|_2^2 = \big\|\Pi_{\phi_{S,\ell}}(f_{T,\ell}) - f_{T,\ell}\big\|_2^2.$$

Taking expectations over $x\sim\mathcal{D}$ yields (13).

(b) Let the target be $\Pi_{\phi_{S,\ell}}(f_{T,\ell})$. Choosing $h = \Pi_{\phi_{S,\ell}}(f_{T,\ell}) \in \mathcal{G}_{\phi_{S,\ell}}$ achieves zero error pointwise, which implies $\inf_{h\in\mathcal{G}_{\phi_{S,\ell}}} \|h - \Pi_{\phi_{S,\ell}}(f_{T,\ell})\|_2^2 = 0$. Taking expectations gives (14). □

Consequently, supervising a quantized student with the FP target $f_{T,\ell}$ cannot drive the expected squared error below the quantization mismatch $\mathbb{E}\big\|\Pi_{\phi_{S,\ell}}(f_{T,\ell}) - f_{T,\ell}\big\|_2^2$, whereas using the student-quantized target $\tilde{f}_{T,\ell} = \Pi_{\phi_{S,\ell}}(f_{T,\ell})$ removes this lower bound. Moreover, letting $\text{dist}(u,S)$ denote the Euclidean distance, we have

$$\mathbb{E}\Big[\big\|\Pi_{\phi_{S,\ell}}(f_{T,\ell}) - f_{T,\ell}\big\|_2^2\Big] \;=\; \mathbb{E}\Big[\text{dist}\big(f_{T,\ell},\mathcal{G}_{\phi_{S,\ell}}\big)^2\Big],$$

so whenever any component of $f_{T,\ell}$ lies outside the learned activation range $[a_\ell^-, a_\ell^+]$ (or outside the weight mapping range), this lower bound *increases monotonically* with the distance beyond the range. Projecting the teacher feature via $\Pi_{\phi_{S,\ell}}$ maps such components back onto the student's attainable set, thereby removing the unattainable residual and stabilizing optimization.

4 EXPERIMENTS

In this section, we evaluate SQuaT across diverse datasets, model architectures, and quantization settings. We first compare SQuaT with the supervised QAT method EWGS (Lee et al., 2021) and the label-free QAT with KD baseline SQAKD (Zhao and Zhao, 2024) on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009). For the large-scale ImageNet-1K benchmark (Deng et al., 2009), we compare SQuaT with SQAKD. To evaluate architectural generalization in vision models, we additionally conduct experiments on the Vision Transformer–based DeiT-Tiny (Dosovitskiy et al., 2020; Touvron et al., 2021) for image classification on CIFAR datasets. For NLP tasks, we evaluate BERT (Devlin et al., 2019) on the GLUE benchmark (Wang et al., 2018). Furthermore, we compare projection strategies for feature distillation to empirically validate the student-aware projection principle described in Section 3.3. Finally, we measure on-device inference performance on edge hardware to assess practical deployability.

4.1 Experimental Setup

We describe the datasets, model architectures, and quantization configurations used in our experiments. For vision tasks, we conduct image classification on CIFAR-10, CIFAR-100, and ImageNet-1K. For NLP tasks, we use the GLUE benchmark, including RTE, SST-2, QNLI. All experiments are performed in a strictly label-free setting without access to ground-truth labels, and the pretrained full-precision teacher remains frozen throughout training.

For vision models, we use ResNet-20 and ResNet-32 for CIFAR and ResNet-18 (He et al., 2016) for ImageNet-1K. To assess architectural generalization, we additionally include the Vision Transformer–based DeiT-Tiny (Touvron et al., 2021). For NLP experiments, we use $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019).

We consider bit-width configurations ranging from 1-bit to 8-bit, and quantize both weights and activations in all experiments. Vision models are trained with an EWGS-based QAT framework (Lee et al., 2021), while Transformer-based models adopt an STE-based QAT framework (Bengio et al., 2013). Despite these differences, we apply the same student-aware projection strategy across all settings. Implementation details are provided in Appendix A.

4.2 Image Classification Results

Tables 1 to 3 report image classification results on CIFAR-10, CIFAR-100, and ImageNet-1K. For CIFAR, we evaluate 1-, 2-, and 4-bit configurations, while for ImageNet-1K we consider 1-, 3-, 4-, and 8-bit settings. We compare SQuaT against the supervised QAT method EWGS and the label-free QAT with KD baseline SQAKD.

On CIFAR-10, SQuaT consistently achieves the best performance across all bit-widths and model variants. Under 1-bit quantization with ResNet-20, SQuaT improves over SQAKD by +0.45 pp, indicating that the proposed student-aware projection remains effective even under severe quantization constraints. The improvements are particularly evident in extremely low-bit settings such as W1A1, where representational capacity is heavily restricted and mismatches between teacher and student representations become more pronounced.

A similar trend is observed on CIFAR-100: under 1-bit quantization with ResNet-32, SQuaT improves over SQAKD by +1.74 pp, with gains particularly pronounced in extreme low-bit settings. Across different model sizes and quantization configurations, SQuaT consistently achieves either the best or competitive performance, demonstrating stable improvements over the baseline.

On the large-scale ImageNet-1K benchmark, SQuaT again shows positive gains across all evaluated settings. Under 1-bit quantization, it improves over SQAKD by +0.40 pp. Although modest in absolute magnitude, such gains are non-trivial given the scale of the dataset and the difficulty of extreme quantization. These results indicate that the benefit of student-aware projection persists even in large-scale training settings where optimization becomes significantly more challenging.

Overall, SQuaT consistently improves performance across datasets of varying scale and complexity, as well as across different bit-widths and model sizes. The performance gap widens in extreme low-bit settings, highlighting the robustness of the proposed projection strategy under stringent quantization constraints. Notably, despite operating in a strictly label-free setting, SQuaT outperforms the supervised QAT baseline EWGS in several configurations. Furthermore, under 4-bit quantization on CIFAR-10 and CIFAR-100, SQuaT even surpasses the corresponding pretrained full-precision models. These results suggest that aligning teacher features to the student's quantization lattice yields systematic advantages over prior projection-based distillation baselines and can produce edge-friendly models with strong accuracy.

4.3 Generality Across Model Designs and Training Settings

We evaluate whether SQuaT consistently improves performance across different model architectures, do-

Table 1: Top-1 Test Accuracy (%) of ResNet-20 and ResNet-32 on CIFAR-10.

Method	ResNet-20 (FP: 92.63)			ResNet-32 (FP: 93.71)		
	W1A1	**W2A2**	**W4A4**	**W1A1**	**W2A2**	**W4A4**
EWGS	86.42±0.01	91.41±0.04	92.49±0.09	86.56±0.03	92.89±0.05	93.75±0.03
SQAKD(EWGS)	86.52±0.05	91.65±0.13	92.60±0.01	88.10±0.11	92.90±0.12	93.59±0.02
SQuaT(EWGS)	**86.97**±0.14	**91.90**±0.11	**92.67**±0.03	**88.17**±0.04	**93.16**±0.09	**93.72**±0.01
Δ (SQuaT - SQAKD)	+0.45	+0.35	+0.07	+0.07	+0.26	+0.13

Table 2: Top-1 Test Accuracy (%) of ResNet-20 and ResNet-32 on CIFAR-100.

Method	ResNet-20 (FP: 68.89)			ResNet-32 (FP: 71.40)		
	W1A1	**W2A2**	**W4A4**	**W1A1**	**W2A2**	**W4A4**
EWGS	56.86±0.24	66.68±0.16	68.49±0.03	59.76±0.44	69.12±0.35	70.60±0.14
SQAKD(EWGS)	56.38±0.03	66.76±0.29	69.13±0.01	59.38±0.04	70.00±0.01	71.65±0.01
SQuaT(EWGS)	**56.74**±0.12	**67.35**±0.01	**69.15**±0.02	**61.12**±0.19	**70.57**±0.04	**71.94**±0.05
Δ (SQuaT - SQAKD)	+0.36	+0.59	+0.02	+1.74	+0.57	+0.29

Table 3: Top-1 Test Accuracy (%) of ResNet-18 on ImageNet-1K.

Bit-width	ResNet-18 (FP: 69.76)		
	SQAKD	**SQuaT**	Δ
W1A1	49.46±0.01	**49.86**±0.25	+0.40
W3A3	67.62±0.11	**67.81**±0.06	+0.19
W4A4	68.56±0.13	**68.77**±0.07	+0.21
W8A8	69.25±0.01	**69.27**±0.01	+0.02

mains, loss functions, and quantization methods. Specifically, we examine whether the gains of SQuaT persist when varying (i) model architecture (CNN, Vision Transformer, and Transformer), (ii) task domain (vision and NLP), (iii) the feature distillation loss, and (iv) the quantization method and training framework.

Table 4 presents results on the Vision Transformer–based DeiT-Tiny. Unlike the CNN-based ResNet family, DeiT-Tiny adopts a self-attention architecture. Nevertheless, SQuaT consistently outperforms SQAKD across all bit-width settings, indicating that the proposed student-aware projection strategy is not tied to convolutional inductive biases and generalizes to structurally distinct architectures.

Table 5 reports results on $\text{BERT}_{\text{BASE}}$ evaluated on the GLUE benchmark (RTE, SST-2, QNLI). Across all tasks and bit-width configurations, SQuaT achieves higher performance than the baseline. These results demonstrate that the proposed approach extends beyond vision classification and remains effective for Transformer-based NLP models, suggesting that the benefits of student-aware projection are not limited to convolutional networks or vision tasks.

In Table 6, we investigate whether SQuaT depends on a specific feature distillation loss by comparing multiple loss functions, including L_1, L_2, KL divergence, and cosine similarity. Across all loss configurations, SQuaT consistently yields improvements over the baseline, indicating that the method is not specialized to a particular loss formulation.

These experiments are conducted under diverse quantization frameworks and quantizer designs. The CNN-based CIFAR/ImageNet results in Tables 1 to 3 use uniform quantization with EWGS-based QAT. In contrast, the DeiT-Tiny experiments adopt STATSQ (Liu et al., 2023) and LSQ (Esser et al., 2019), while the NLP experiments employ TwnQuantizer (Li et al., 2016) and SymQuantizer (Shen et al., 2020). All Transformer-based models are trained using STE-based QAT.

Despite these heterogeneous quantization settings, SQuaT consistently improves performance. These observations suggest that the effectiveness of SQuaT does not rely on a particular architectural assumption or quantization implementation, but rather stems from the general principle of aligning teacher features to the student's quantization lattice.

4.4 Empirical Analysis of Student-Aware Projection

In this section, we conduct both quantitative and qualitative analyses of the proposed student-aware projec-

Table 4: Top-1 Test Accuracy (%) of DeiT-Tiny on CIFAR-10 and CIFAR-100.

Method	CIFAR-10 (FP: 97.26)			CIFAR-100 (FP: 97.33)		
	W2A2	W3A3	W4A4	W2A2	W3A3	W4A4
SQAKD	85.78±0.11	85.81±0.09	85.88±0.01	86.39±0.01	86.21±0.16	86.78±0.15
SQuaT	**86.67**±0.23	**87.50**±0.02	**87.98**±0.01	**87.66**±0.23	**89.48**±0.51	**90.20**±0.23
Δ	+0.89	+1.69	+2.10	+1.27	+3.27	+3.42

Table 5: Performance (%) of $\text{BERT}_{\text{BASE}}$ on the GLUE Benchmark.

Bit-width	Method	RTE 2.5k (FP: 68.95)	SST-2 67k (FP: 93.23)	QNLI 108k (FP: 91.25)
W3A3	SQAKD	54.15±0.51	86.24±0.16	81.88±0.13
	SQuaT	**57.22**±0.25	**86.93**±0.16	**83.14**±0.06
	Δ	+3.07	+0.69	+1.26
W4A4	SQAKD	57.40±1.02	92.44±0.16	90.01±0.16
	SQuaT	**65.88**±0.25	**92.72**±0.08	**90.21**±0.03
	Δ	+8.48	+0.28	+0.20
W8A8	SQAKD	65.70±0.51	93.41±0.08	91.53±0.02
	SQuaT	**69.13**±0.25	**93.70**±0.16	**91.86**±0.01
	Δ	+3.43	+0.29	+0.33

Table 6: Top-1 Test Accuracy (%) of ResNet-20 on CIFAR-10 with various feature loss types.

Loss Func.	W1A1	W2A2	W4A4
SQAKD	86.47±0.28	91.62±0.03	92.59±0.02
SQuaT_{L_1}	86.78±0.14	91.77±0.09	92.64±0.03
SQuaT_{L_2}	**86.97**±0.14	**91.90**±0.11	92.67±0.03
SQuaT_{KL}	86.74±0.15	91.74±0.03	**92.70**±0.05
$\text{SQuaT}_{\text{Cos}}$	86.74±0.26	91.66±0.05	92.66±0.03

tion strategy. Table 7 compares different projection schemes used in feature distillation. Non-aware directly distills full-precision teacher features without any alignment. Teacher-aware, following QFD (Zhu et al., 2023), attaches a quantizer to the teacher, calibrates it over several epochs, and then projects the teacher features using the calibrated quantizer. In contrast, Student-aware projects teacher features onto the student's quantization lattice before performing distillation, as in SQuaT.

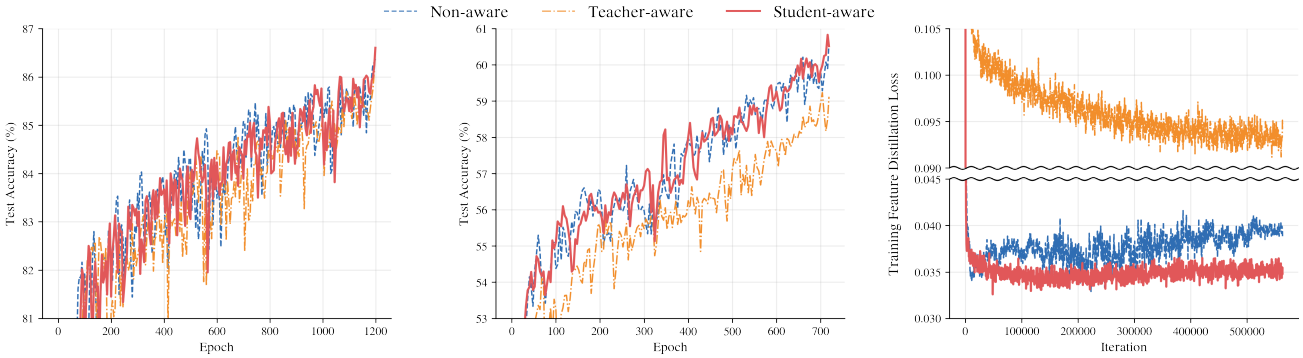
As shown in Table 7, the Student-aware scheme achieves higher accuracy than both Non-aware and Teacher-aware approaches across all bit-widths and model configurations. Notably, Teacher-aware projection often performs worse than the Non-aware variant. This suggests that a quantizer calibrated independently on the teacher side, without accounting for the student's representational constraints, may still leave a mismatch between teacher and student representations. In contrast, explicitly aligning teacher features to the student's representational space proves more effective than either directly using full-precision features or relying on teacher-side calibration.

Figure 3 shows the learning curves of feature distillation loss and Top-1 test accuracy for the three projection schemes. The Student-aware approach converges to the lowest feature distillation loss throughout training, indicating stable optimization. In contrast, the Teacher-aware scheme converges to a higher loss than even the Non-aware scheme, revealing a clear limitation. The Non-aware scheme initially decreases faster but later exhibits an increase in feature distillation loss as training progresses, suggesting a mismatch between full-precision teacher features and quantized student representations. As a result, the Student-aware approach achieves the highest final accuracy, whereas the others oscillate at higher loss levels or converge more slowly, ultimately leading to inferior performance across training and evaluation.

These empirical observations are consistent with the theoretical analysis in Section 3.3. Theorem 1 shows that projecting teacher features onto the student's quantization lattice removes the irreducible mismatch. Accordingly, the superior convergence of the Student-aware scheme indicates effective removal of this lower bound, leading to more stable optimization and higher final performance.

Table 7: Analysis of Feature Distillation Method.

Method	CIFAR-10 / ResNet-20 (FP: 92.63)			CIFAR-100 / ResNet-32 (FP: 71.40)		
	W1A1	W2A2	W4A4	W1A1	W2A2	W4A4
Non-aware	86.30±0.12	91.73±0.13	92.64±0.02	60.77±0.16	70.49±0.04	71.69±0.07
Teacher-aware	86.44±0.18	91.59±0.09	92.44±0.05	59.59±0.24	69.50±0.15	71.14±0.13
Student-aware	**86.97**±0.14	**91.90**±0.11	**92.67**±0.03	**61.02**±0.19	**70.57**±0.04	**71.94**±0.05



(a) CIFAR-10 / ResNet-20 / W1A1 (b) CIFAR-100 / ResNet-32 / W1A1 (c) CIFAR-100 / ResNet-32 / W1A1

Figure 3: Training curves of feature distillation loss and Top-1 test accuracy under different bit-width settings.

4.5 Sensitivity to Loss Weight

We analyze the sensitivity of SQuaT to the feature distillation loss weight under 4-bit quantization on CIFAR-100 with ResNet-32 (see Table 8). Following the weighting scheme of SQAKD, we fix the logit loss weight to $\lambda_{\text{logit}} = 100$ and vary the feature loss weight λ_{feat}. Among the evaluated configurations, $\lambda_{\text{feat}} = 10$ achieves the best accuracy of 71.94%. Accordingly, we use $\lambda_{\text{logit}} = 100$ and $\lambda_{\text{feat}} = 10$ for the CIFAR-100 ResNet-32 4-bit setting, while all other experiments set $\lambda_{\text{logit}} = \lambda_{\text{feat}} = 1$.

Table 8: Sensitivity to the feature loss weight λ_{feat} on CIFAR-100 with ResNet-32 under the 4-bit setting.

λ_{feat}	Top-1 Acc.	λ_{feat}	Top-1 Acc.
10	**71.94** ± 0.05	**60**	71.77 ± 0.14
20	71.86 ± 0.06	**70**	71.76 ± 0.06
30	71.71 ± 0.03	**80**	71.78 ± 0.06
40	71.75 ± 0.05	**90**	71.79 ± 0.04
50	71.73 ± 0.03	**100**	71.78 ± 0.01

4.6 On-Device Evaluation

To evaluate practical efficiency, we measure inference performance on an NVIDIA Jetson Nano using ResNet-18 trained on ImageNet (see Table 9). Compared with FP32 inference, INT8 inference increases throughput from 294.32 FPS to 2046.03 FPS and reduces latency from 3.44 ms to 0.53 ms, corresponding to a 6.49× speedup. These results demonstrate that SQuaT-trained quantized models provide substantial efficiency gains on edge devices while maintaining competitive accuracy.

Table 9: Inference Performance on Jetson Nano.

Model	Bit width	Throughput (FPS)	Latency (ms)	Speedup
ResNet-18	FP32	294.32	3.442	-
	INT8	2046.03	0.53	6.49×

5 CONCLUSION

We identify a fundamental limitation in label-free QAT with KD: prior teacher-aware projection suffers from a distributional mismatch between full-precision teacher features and quantized student representations, inducing an unattainable residual and an irreducible lower bound on the distillation loss. To address this issue, we introduce SQuaT, which applies a student-aware projection to align teacher features with the student's quantization lattice, thereby eliminating this bound by restricting targets to the student's attainable set. Experiments demonstrate consistent improvements over prior label-free QAT with KD methods, particularly in extreme low-bit regimes, and the approach is agnostic to model components, making it broadly applicable.

Acknowledgements

This work was supported by Hyundai Motor Company and Kia, and partially supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (o.RS-2025-02219317, AI Star Fellowship(Kookmin University)), and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-23524783)

References

Back, J., Ahn, N., and Kim, J. (2025). Magnitude attention-based dynamic pruning. *Expert Systems with Applications*, 276:126957.

Bai, Y., Wang, Y.-X., and Liberty, E. (2018). Proxquant: Quantized neural networks via proximal operators. *arXiv preprint arXiv:1810.00861*.

Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Chen, S., Wang, W., and Pan, S. J. (2019). Metaquant: Learning to quantize by learning to penetrate non-differentiable quantization. *Advances in Neural Information Processing Systems*, 32.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Choi, J., Wang, Z., Venkataramani, S., Chuang, P. I.-J., Srinivasan, V., and Gopalakrishnan, K. (2018). Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. (2019). Learned step size quantization. *arXiv preprint arXiv:1902.08153*.

Gong, R., Liu, X., Jiang, S., Li, T., Hu, P., Lin, J., Yu, F., and Yan, J. (2019). 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.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

Han, S., Mao, H., and Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713.

Kim, J., Yoo, J., Song, Y., Yoo, K., and Kwak, N. (2023). Finding efficient pruned network via refined gradients for pruned weights. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9003–9011.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

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

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

Li, F., Liu, B., Wang, X., Zhang, B., and Yan, J. (2016). Ternary weight networks. *arXiv preprint arXiv:1605.04711*.

Li, Y., Dong, X., and Wang, W. (2019). Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. *arXiv preprint arXiv:1909.13144*.

Liu, S.-Y., Liu, Z., and Cheng, K.-T. (2023). Oscillation-free quantization for low-bit vision transformers. In *International conference on machine learning*, pages 21813–21824. PMLR.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., Van Baalen, M., and Blankevoort, T. (2021). A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. (2020). Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8815–8821.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR.

Uhlich, S., Mauch, L., Yoshiyama, K., Cardinaux, F., Garcia, J. A., Tiedemann, S., Kemp, T., and Nakamura, A. (2019). Differentiable quantization of deep neural networks. *arXiv preprint arXiv:1905.11452*, 2(8).

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Zhao, K. and Zhao, M. (2024). Self-supervised quantization-aware knowledge distillation. In *International Conference on Artificial Intelligence and Statistics*, pages 4375–4383. PMLR.

Zhu, K., He, Y.-Y., and Wu, J. (2023). Quantized feature distillation for network quantization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11452–11460.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable]: Yes. See Section 3.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable]: No
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]: Yes

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]: Yes
   (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable]: Yes
   (c) Clear explanations of any assumptions. [Yes/No/Not Applicable]: Yes

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]: Yes
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable]: Yes. See Appendix A
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable]: Yes
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable]: Yes

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable]: Yes
   (b) The license information of the assets, if applicable. [Yes/No/Not Applicable]: No

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable]: Yes

(d) Information about consent from data providers/curators. [Yes/No/Not Applicable]: Not Applicable

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable]: Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable]: Not Applicable

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable]: Not Applicable

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable]: Not Applicable

Supplementary Materials: Self-Supervised Knowledge Distillation via Student-Aware Quantized Teacher Features

A IMPLEMENTATION DETAILS

On CIFAR-10 (Krizhevsky et al., 2009), ResNet-20 and ResNet-32 (He et al., 2016) were trained for 1200 epochs with a batch size of 256. The learning rates were set to 1×10^{-3} for model parameters and 1×10^{-5} for quantization parameters, and a weight decay of 1×10^{-4} was applied. On CIFAR-100 (Krizhevsky et al., 2009), the same architectures were trained for 720 epochs with a batch size of 64. The learning rates were set to 5×10^{-4} for model parameters and 5×10^{-6} for quantization parameters, and a weight decay of 5×10^{-4} was used.

DeiT-Tiny (Touvron et al., 2021) was also evaluated on the same datasets. The model was trained for 300 epochs with a batch size of 128. The learning rates for both the model parameters and quantization parameters were set to 1×10^{-3}, and a weight decay of 5×10^{-2} was applied.

On ImageNet-1K (Deng et al., 2009), ResNet-18 (He et al., 2016) was trained for 100 epochs with a batch size of 128. The model was initialized with pretrained weights provided by PyTorch (Paszke et al., 2019) . The learning rate was set to 1×10^{-2} for model parameters and 1×10^{-5} for quantization parameters, with a weight decay of 1×10^{-4}.

For $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019), experiments were conducted on the GLUE benchmark (Wang et al., 2018) for 3 epochs with a batch size of 32. The learning rate was set to 2×10^{-5} for both the model parameters and quantization parameters, and a weight decay of 1×10^{-2} was applied. The model was initialized with pretrained weights provided by HuggingFace.

ResNet-based models were optimized using the Adam optimizer (Kingma and Ba, 2014). However, when training ResNet-18 on ImageNet-1K, the model parameters were optimized using Stochastic Gradient Descent (SGD), while the quantization parameters were optimized using Adam. For DeiT-Tiny and $\text{BERT}_{\text{BASE}}$, AdamW (Loshchilov and Hutter, 2017) and BertAdam optimizers were used, respectively.

The feature locations where SQuaT is applied depend on the model architecture. For CNN-based models (ResNet), SQuaT is applied to the input feature of the last convolutional layer in the final residual block to perform distillation in the quantized space. For Transformer-based models (DeiT, BERT), SQuaT is applied to the input feature of the first linear layer in the FFN of the last Transformer block (i.e., the attention output), with distillation performed over all tokens.

The implementation was based on Python 3.11.11 and PyTorch 2.8.0, and all experiments were conducted on an NVIDIA RTX A6000 GPU.

Table 10: Implementation Details

Dataset	**Model**	**Epoch**	**Batch**	**Optimizer**	$\mathbf{LR}_m$	$\mathbf{LR}_q$	**Weight Decay**
CIFAR-10	ResNet-20	1200	256	Adam	1×10^{-3}	1×10^{-5}	1×10^{-4}
	ResNet-32	1200	256	Adam	1×10^{-3}	1×10^{-5}	1×10^{-4}
	DeiT-Tiny	300	128	AdamW	1×10^{-3}	1×10^{-3}	5×10^{-2}
CIFAR-100	ResNet-20	720	64	Adam	5×10^{-4}	5×10^{-6}	5×10^{-4}
	ResNet-32	720	64	Adam	5×10^{-4}	5×10^{-6}	5×10^{-4}
	DeiT-Tiny	300	128	AdamW	1×10^{-3}	1×10^{-3}	5×10^{-2}
ImageNet-1K	ResNet-18	100	128	Adam, SGD	1×10^{-2}	1×10^{-5}	1×10^{-4}
GLUE	$\text{BERT}_{\text{BASE}}$	3	32	BertAdam	2×10^{-5}	2×10^{-5}	1×10^{-2}

B ADDITIONAL EXPERIMENTS

Table 11 reports the image classification results on Tiny-ImageNet (Le and Yang, 2015). In this experiment, we evaluate 1-, 3-, and 4-bit quantization settings, and the models are trained using an EWGS-based QAT framework (Lee et al., 2021). Tiny-ImageNet contains 200 classes with large intra-class variation, making it more complex than CIFAR-100. Despite this increased complexity, SQuaT consistently outperforms SQAKD (Zhao and Zhao, 2024) across most bit-width settings.

In particular, under 1-bit quantization, SQuaT achieves a +0.34 pp improvement over SQAKD, which represents the largest performance gain among all bit-width configurations. This result demonstrates that the proposed method remains effective even under extreme low-bit quantization constraints.

These results indicate that SQuaT maintains stable performance improvements on larger-scale datasets, and a similar trend is consistently observed in large-scale ImageNet-1K experiments, as shown in Table 3.

Table 11: Top-1 Test Accuarcy (%) of ResNet-18 on Tiny-ImageNet.

Bit-width	**ResNet-18 (FP: 65.55)**		
	SQAKD	**SQuaT**	Δ
W1A1	59.18±0.04	**59.52**±0.03	+0.34
W3A3	65.79±0.15	**65.59**±0.02	-0.20
W4A4	65.73±0.11	**65.88**±0.03	+0.15
W8A8	65.88±0.01	**66.05**±0.05	+0.17