

# Teacher Intervention: Improving Convergence of Quantization Aware Training for Ultra-Low Precision Transformers

Minsoo Kim<sup>1</sup>, Kyuhong Shim<sup>2</sup>, Seongmin Park<sup>1</sup>, Wonyong Sung<sup>2</sup> and Jungwook Choi<sup>1\*</sup>

<sup>1</sup>Department of Electronic Engineering, Hanyang University

<sup>2</sup>Department of Electrical and Computer Engineering, Seoul National University

{minsoo2333, skstjdals, choij}@hanyang.ac.kr

{skhu20, wysung}@snu.ac.kr

## Abstract

Pre-trained Transformer models such as BERT have shown great success in a wide range of applications, but at the cost of substantial increases in model complexity. Quantization-aware training (QAT) is a promising method to lower the implementation cost and energy consumption. However, aggressive quantization below 2-bit causes considerable accuracy degradation due to unstable convergence, especially when the downstream dataset is not abundant. This work proposes a proactive knowledge distillation method called *Teacher Intervention* (TI) for fast converging QAT of ultra-low precision pre-trained Transformers. TI intervenes layer-wise signal propagation with the intact signal from the teacher to remove the interference of propagated quantization errors, smoothing loss surface of QAT and expediting the convergence. Furthermore, we propose a *gradual* intervention mechanism to stabilize the recovery of subsections of Transformer layers from quantization. The proposed schemes enable fast convergence of QAT and improve the model accuracy regardless of the diverse characteristics of downstream fine-tuning tasks. We demonstrate that TI consistently achieves superior accuracy with significantly lower fine-tuning iterations on well-known Transformers of natural language processing as well as computer vision compared to the state-of-the-art QAT methods.

## 1 Introduction

The Transformer-based pre-trained neural networks have significantly improved the performance of various applications of artificial intelligence, including natural language processing (NLP) (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020) and computer vision (CV) (Dosovitskiy et al., 2020; Touvron et al., 2021; Liu et al., 2021a). The self-attention mechanism represents these models

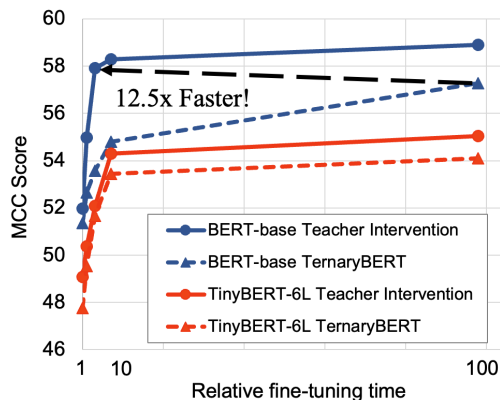


Figure 1: Comparison of fine-tuning time and accuracy of ternary weight quantized Transformer models between Teacher Intervention (TI) and TernaryBERT (BERT-base and TinyBERT-6L for CoLA task). TI achieves higher accuracy within  $12.5\times$  shorter fine-tuning time.

(Vaswani et al., 2017), which links different symbols within a sequence to obtain a relational representation. Thanks to the exceptional performance of the pre-trained Transformer models, there have been increasing needs for their efficient deployment. However, the gigantic size of the pre-trained Transformer models hinders straightforward implementation. Even relatively small models like BERT-base (Devlin et al., 2019) contain a few hundred million parameters, incurring profound memory and computation overhead for resource-constrained devices with limited memory and computing fabric. Therefore, seminal research efforts attempted to reduce this burden via model compression. (Behnke and Heafield, 2020) and (Gordon et al., 2020) pruned unimportant weights to reduce the number of parameters, while (Mao et al., 2020) further employed low-rank matrix factorization. In addition, Knowledge Distillation (KD) (Hinton et al., 2015) was employed in (Sanh et al., 2019; Sun et al., 2019, 2020; Wang et al., 2020) to transfer knowledge of the original model (*teacher*) to the compressed one (*student*) by mimicking the

\*Corresponding Author

Teacher’s behavior.

Among many model compression techniques, *quantization-aware training* (QAT) stands out for its recent success in reducing computational complexity and memory requirements of Transformer models (Bhandare et al., 2019; Zafrir et al., 2019; Kim et al., 2021). QAT reflects quantization errors during the forward pass computation of stochastic gradient descent to train a more accurate quantized model. However, quantizing weight parameters of Transformers to a precision lower than 2-bits degrades the accuracy, especially when the dataset size for the target downstream tasks is not large enough (Zhang et al., 2020b; Bai et al., 2021). Although few-sample fine-tuning of Transformer models has been reported to be highly unstable (Grießhaber et al., 2020; Yu et al., 2021; Zhang et al., 2020a; Dodge et al., 2020; Mosbach et al., 2020), efforts on understanding and improving QAT on small dataset tasks are limited. XTC (Wu et al., 2022) recently revealed that a simple expansion in fine-tuning iterations could heal the QAT accuracy. Still, it caused an order-of-magnitude increase in QAT time, hindering the broad deployment of quantized Transformers.

This work proposes a proactive KD method called *Teacher Intervention* (TI) for fast converging QAT of ultra-low precision pre-trained Transformers. We reveal that the difficulty of quantization on few-sample fine-tuning originates from disruption of loss surface due to quantization error propagation. To mitigate this undesirable phenomenon, we propose TI to intervene layer-wise signal propagation with the intact signal from the teacher. TI removes the interference of propagated quantization errors to smooth out the loss surface and expedite the convergence. We further discover that subsections of Transformer layers exhibit different susceptibility to quantization error for diverse downstream tasks. Thus, we propose a gradual intervention mechanism that first intervenes at the attention output for stable tuning of the feed-forward network, followed by self-attention map intervention for its recovery from quantization. The proposed gradual intervention along the subsections of Transformer layers enables fast convergence of QAT and improves the model accuracy regardless of the diverse characteristics of downstream fine-tuning tasks. We perform an extensive evaluation on various fine-tuned Transformers (BERT-base/large (Devlin et al., 2019), TinyBERT-4L/6L (Jiao et al.,

2020), and SkipBERT-6L (Wu et al., 2022) for NLP, and ViT (Dosovitskiy et al., 2020) for CV) and demonstrate that TI consistently achieves superior accuracy with lower fine-tuning iterations compared to the state-of-the-art QAT methods. In particular, TI outperforms TernaryBERT (Zhang et al., 2020b) on GLUE tasks with  $12.5\times$  savings in fine-tuning hours, as shown in Fig.1.

## 2 Related Work

### 2.1 Knowledge Distillation for BERT Compression

Knowledge distillation (KD) (Hinton et al., 2015) is a transfer learning framework that passes on knowledge of a large model (*teacher*) to a smaller one (*student*). Since KD provides extra guidance on how the student should behave, it can help mitigate accuracy degradation for model compression. Therefore, KD has been widely employed in training smaller BERT models for various application domains.

The most common distillation approach is to match the probability distribution from the final output softmax between the teacher and student for the same input, as shown in DistilBERT (Sanh et al., 2019). In addition to the distillation loss at the model output, PKD (Sun et al., 2019) suggested loss on intermediate output that matched the normalized output of the teacher and the student at each Transformer layer. MobileBERT (Sun et al., 2020) also employed per-head attention map transfer along with the customized network structure for constructing efficient Transformers. MiniLM (Wang et al., 2020) further transferred knowledge from the self-attention map as well as the value-relation. Considering the structural mismatch between the Teacher and Student models, MiniLM performed distillation only at the last Transformer layer.

While the above mentioned studies focused on the task-agnostic BERT, there have been several efforts (Tang et al., 2019; Aguilar et al., 2020) to train tiny task-specific students. In this line of study, the task-specific, downstream fine-tuned BERT is first prepared, and the Student is trained with KD by utilizing this fine-tuned model as a teacher. As a hybrid approach, TinyBERT (Jiao et al., 2020) proposed a two-step KD, the first step for general distillation, followed by task-specific distillation.

Although extensive research has been conducted to utilize KD for BERT compression, there are

limited efforts in investigating and developing KD techniques for model quantization, an orthogonal model compression method. In this work, we develop a new KD technique primarily focusing on improving QAT of pre-trained Transformers. In particular, we reveal that more aggressive intervention of the teacher on the subsections of each Transformer layer helps the ultra-low precision model regain the model accuracy.

## 2.2 Quantization for Ultra-Low Precision BERT

Quantization is a promising technique for reducing the high inference cost of large-scale models without changing the model structure. Instead of representing numbers with the 32-bit floating-point (FP32) format, employing fixed-point representation, such as 8-bit integer (INT8) quantization, has achieved significant speedup and storage savings for BERT (Zafir et al., 2019; Kim et al., 2021). However, direct quantization of weight parameters would suffer accuracy degradation of the original model accuracy when the quantization bit-precision is low. Therefore, quantization-aware training (QAT) is commonly applied for ultra-low precision model quantization.

Recently, QAT has been applied for compressing BERT with precision lower than 2-bit. TernaryBERT (Zhang et al., 2020b) represents each weight element into one of three values  $\{-1, 0, 1\}$ . TernaryBERT actively incorporates KD into QAT for improving accuracy degradation. Especially, KD with the MSE loss on the attention score (before taking Softmax) and the output of each Transformer layer is employed for QAT. To further reduce the bit-precision, BinaryBERT (Bai et al., 2021) suggested a modified QAT procedure that initializes the weights for binary quantization. However, ternarizing or binarizing weight parameters significantly degrades the model accuracy, especially when the dataset size for the target downstream tasks is not large enough.

In fact, it has been reported that finetuning BERT on downstream tasks with insufficient data is highly unstable (Grieblhaber et al., 2020; Yu et al., 2021). As a result, several works proposed modified finetuning procedures for improving the stability (Zhang et al., 2020a; Dodge et al., 2020; Mosbach et al., 2020). Still, the proposed approaches do not address the sensitivity of Transformer models on QAT for small datasets. XTC (Wu et al.,

2022) recently proposed a QAT method with significantly increased iterations and data augmentation to improve quantization accuracy of ultra-low bit precision Transformers. As illustrated in Fig. 1, however, this prolonged fine-tuning results in sizable deployment overhead, let alone costly data augmentation. In this work, we discover that quantization significantly disrupts the propagation of self-attention in Transformer layers hindering the optimization process of QAT. Therefore, we propose a new KD-based method that proactively intervene the error propagation to improve convergence of QAT methods.

## 3 Background and Motivation

### 3.1 Transformer Layer

The BERT model (Devlin et al., 2019) is built with Transformer layers (Vaswani et al., 2017), which include two main sub-modules: Multi-Head Attention (MHA) and Feed-Forward Network (FFN). Input to the  $l$ -th Transformer layer is  $\mathbf{X}_l \in \mathbb{R}^{n \times d}$  where  $n$  and  $d$  are the sequence length and hidden state size, respectively. Let  $H$  be the number of attention heads and  $d_h = d/H$ .  $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{d \times d_h}$  are the weight parameters converting  $\mathbf{X}_l$  into Query ( $\mathbf{Q} = \mathbf{X}_l \mathbf{W}_h^Q$ ), Key ( $\mathbf{K} = \mathbf{X}_l \mathbf{W}_h^K$ ), and Value ( $\mathbf{V} = \mathbf{X}_l \mathbf{W}_h^V$ ), respectively. Then, attention score ( $\mathbf{AS}_h = \mathbf{Q} \mathbf{K}^\top$ ), self-attention map ( $\mathbf{SA}_h = \text{Softmax}_h(\frac{\mathbf{AS}_h}{\sqrt{d}})$ ), and attention context ( $\mathbf{AC} = \mathbf{SA}_h \mathbf{V}$ ) can be defined.

The attention output ( $\mathbf{AO}$ ) is defined as  $\mathbf{AO} = \text{MHA}(\mathbf{X}_l) = \text{Concat}(\mathbf{AC}_1, \mathbf{AC}_2, \dots, \mathbf{AC}_{N_H}) \times \mathbf{W}^O$ . Motivated by (Kobayashi et al., 2020), the attention output can be re-written per each token  $i$ :

$$\text{MHA}(\mathbf{X}_l)(i) = \sum_{j=1}^n \alpha_{i,j} f(\mathbf{X}_l(j)), \quad (1)$$

where  $f(x) := (x \mathbf{W}^V + b^V) \mathbf{W}^O$  and  $\alpha_{i,j}$  is  $j$ 'th value of  $i$ 'th token in  $\mathbf{SA}_h$ . Therefore, the computation of MHA consists of two parts: self-attention generation (SA-GEN) corresponding to the attention map ( $\alpha$ ) and self-attention propagation (SA-PROP) corresponding to  $f(x)$ . Fig. 3 illustrates the structure of a Transformer layer indicating regions corresponding to SA-GEN and SA-PROP. FFN consists of two fully-connected layers with weight parameters  $\mathbf{W}^1$  and  $\mathbf{W}^2$ :

$$\text{FFN}(\mathbf{Y}_l) = \text{GeLU}(\mathbf{X}_l \mathbf{W}^1 + b^1) \mathbf{W}^2 + b^2. \quad (2)$$

Therefore, an output of Transformer layer  $X_{l+1}$  is defined as:

$$\begin{aligned} \mathbf{Y}_l &= \text{LayerNorm}(\mathbf{X}_l + \text{MHA}(\mathbf{X}_l)), \\ \mathbf{X}_{l+1} &= \text{LayerNorm}(\mathbf{Y}_l + \text{FFN}(\mathbf{Y}_l)). \end{aligned} \quad (3)$$

### 3.2 Quantization-Aware Training

Quantization-aware training (QAT) emulates inference-time quantization during fine-tuning of the full-precision model to adjust the parameters to be robust to the quantization error (Hwang and Sung, 2014) (Hubara et al., 2016). In particular, ternary quantization represents all the weight parameters ( $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V, \mathbf{W}^O, \mathbf{W}^1, \mathbf{W}^2$ ) into ternary values  $t \in \{+1, 0, -1\}$  along with a scale factor  $s$  for sub-2bit inference. In this work, we follow the approach of TWN (Zhu et al., 2016) that analytically estimates the optimal scale factor  $s$  and the ternary weight  $\mathbf{W}_t = \{t\}^{|\mathbf{W}|}$  to minimize  $\|\mathbf{W} - s \cdot \mathbf{W}_t\|$ .

Due to aggressive bit-reduction, ternary quantization causes significant accuracy loss. KD can help compensate for accuracy degradation, where the original full-precision model works as a teacher to guide the training of the quantized model as a student. In case of Transformer models, TernaryBERT (Zhang et al., 2020b) applied KD on every output activation  $X_{l+1}$  as well as the attention score (AS) with mean squared error (MSE) loss:

$$L_{trm} = \sum_{l=1}^{L+1} \text{MSE}(\mathbf{X}_l^S, \mathbf{X}_l^T) + \sum_{l=1}^L \text{MSE}(\mathbf{AS}_l^S, \mathbf{AS}_l^T), \quad (4)$$

where  $S$  and  $T$  represent the student and teacher models, respectively. Also, the output logits of the student ( $P^S$ ) and the teacher ( $P^T$ ) are used in TernaryBERT to compute the cross-entropy (CE) loss:

$$L_{pred} = \text{CE}(P^S, P^T). \quad (5)$$

We follow the settings of TernaryBERT as our baseline QAT method.

### 3.3 Challenges

Despite attempts to bridge the accuracy gap, prior works on ultra-low precision Transformers (Bai et al., 2021; Zhang et al., 2020b) still suffer noticeable accuracy degradation, especially when the dataset size is small. Recall (Mosbach et al., 2020) that the few-sample fine-tuning is unstable. But we observed that QAT often fails even if it fine-tunes from the successfully trained model.

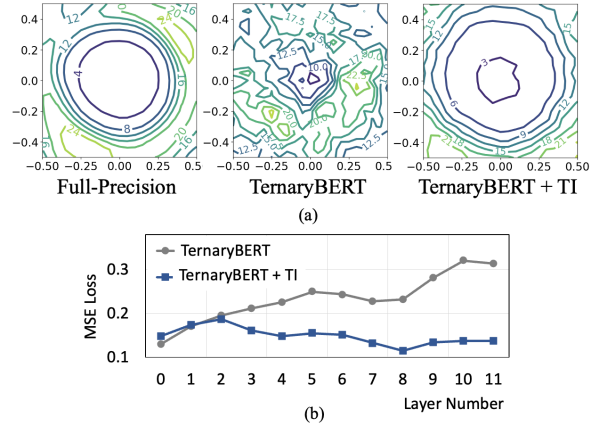


Figure 2: (a) Loss landscape visualization of a fine-tuned Transformer for a few-sample task (BERT-base, CoLA): A well-trained full-precision model and the quantized models without and with Teacher Intervention (TI). (b) MSE loss at the output of Transformer layers of TernaryBERT without and with TI

To gain intuition on this failure, we visualize the loss landscape of the quantized Transformers fine-tuned for a few-sample task (BERT-base, CoLA) in Fig. 2(a). In contrast to the smooth loss surface of the well-trained full-precision model, TernaryBERT exhibits sharp curvatures with many valleys, suffering unstable fine-tuning due to quantization. Investigating the internal behavior of Transformer layers under quantization elucidates the cause of unstable QAT. We measure the mean-square error (MSE) of the output of each Transformer layer between the full-precision baseline (teacher) and TernaryBERT (student). Fig. 2(b) reveals a prominent trend that MSE grows over the layers. (Similar trends can be observed in the other GLUE tasks.) This inflated error along the layers would degrade the model’s accuracy.

In this work, we focus on managing this aggravating impact of quantization errors on Transformers with a proactive knowledge distillation called *Teacher Intervention* (TI). Interestingly, as shown in Fig. 2(b), TI successfully suppresses the error propagation and flattens the loss surface for favorable convergence without precipitously increased fine-tuning iterations. We discuss the technical details of TI in the next section.

## 4 Method

### 4.1 Teacher Intervention

Teacher intervention (TI) is a KD method that aggressively intervenes in the student’s signal propagation along the Transformer layers to suppress propagation of quantization error. Fig. 3(b) illus-



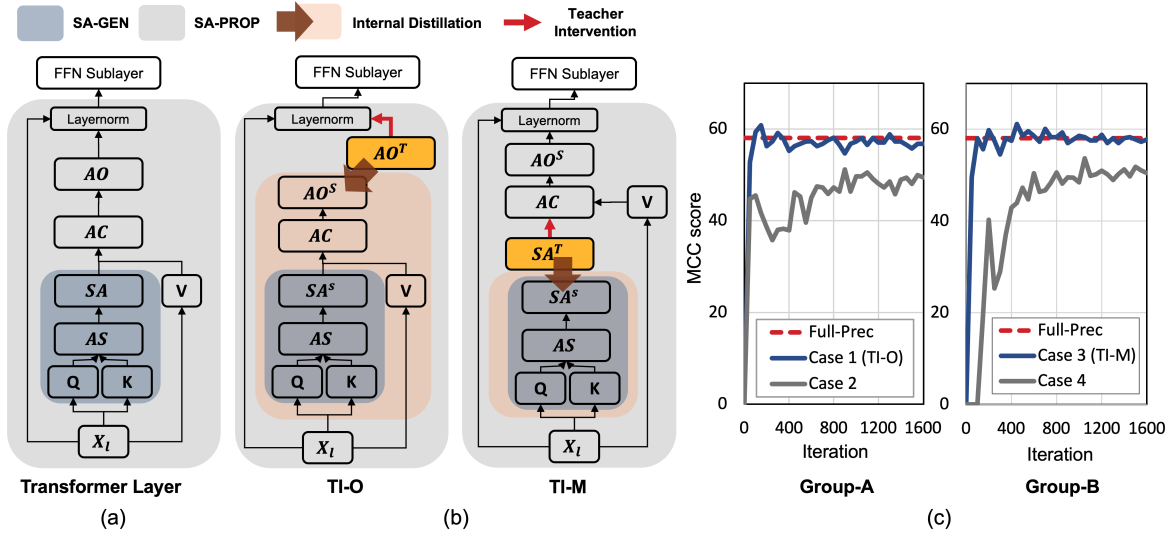


Figure 3: (a) Architecture of Transformer layer. (b) Locations of Teacher Intervention: output intervention (TI-O) and map intervention (TI-M).  $AO^T$  and  $SA^T$  are teacher’s attention outputs. (c) Accuracy curves of controlled experiments. Group-A: Quantize all sub-layers with TI-O (Case 1) or do not quantize MHA (Case 2), Group-B: Quantize all sub-layers with TI-M (Case 3) or do not quantize SA-GEN (Case 4). Note rapid convergence for the cases with TI.

trates the two options for teacher intervention. First, intervention on the attention output (a.k.a. *output intervention*, TI-O) replaces the student’s attention output ( $AO^S$  in Fig. 3(b)) in each Transformer layer with the teacher’s ( $AO^T$  in Fig. 3(b)). In this case, the FFN sub-layers are trained with ultra-low precision quantization without concerns of erroneous input from the preceding MHA. Meanwhile, the computation within the MHA sub-layer is quantized for internal distillation. Similarly, intervention on the self-attention map (a.k.a. *map intervention*, TI-M) replaces the student’s SA-GEN output ( $SA^S$  in Fig. 3(b)) with the teacher’s ( $SA^T$  in Fig. 3(b)).<sup>1</sup>

The development of TI is motivated by the previous observation of the aggravating impact of quantization error along the layers (cf. Fig. 2(b)). We conjecture that the root cause of this phenomenon is error propagation instead of the quantization error itself. To confirm our hypothesis, we conducted controlled experiments for TI-O and TI-M with two groups of quantization cases:

- Group-A: Quantize all sub-layers with TI-O (Case1) or do not quantize MHA (Case2).
- Group-B: Quantize all sub-layers with TI-M (Case3) or do not quantize SA-GEN (Case4).

The key difference between the two groups is that Case2/4 propagates the quantization error

<sup>1</sup>We empirically investigated other options for teacher intervention and found that TI-O and TI-M were most representative.

through the sub-layers while Case1/3 does not, thanks to TI. Fig. 3(c) shows the convergence curves of the four cases on CoLA. As shown in the figure, Case1/3 converges rapidly to full-precision accuracy despite the ultra-low bit quantization in all the sub-layers. Whereas Case2/4 converges slowly to the sub-optimal point with noticeable accuracy degradation. Although Case2/4’s MHA/SA-GEN computations are in full-precision, the error propagated from its preceding quantized FFN sub-layers still affects it, corrupting the attention output/self-attention map. On the other hand, TI-O and TI-M interrupt this error propagation to stabilize QAT on FFN sub-layers and MHA outputs, respectively. Therefore, Fig. 2(b) confirms that the error propagation disappears when TI is applied.

Motivated by this insightful observation, we devise a new QAT method that employs TI for the step-by-step reconstruction of sub-layers of Transformers. In the first step, TI is used to fine-tune the quantized weights of either FFN (TI-O) or SA-PROP (TI-M) sub-layers. Note that the convergence in this step is quick, as shown in Fig. 3(c). Therefore, Step1 takes only a fraction of fine-tuning epoch. In the second step, quantization is applied to the entire weights of Transformer layers for QAT. Since part of the parameters is already trained in Step1 with the guidance of TI, Step2 converges faster to a superior local minimum. In Sec. 5, we demonstrate this improved convergence leads to boosts in accuracy for ultra-low bit Transformers.

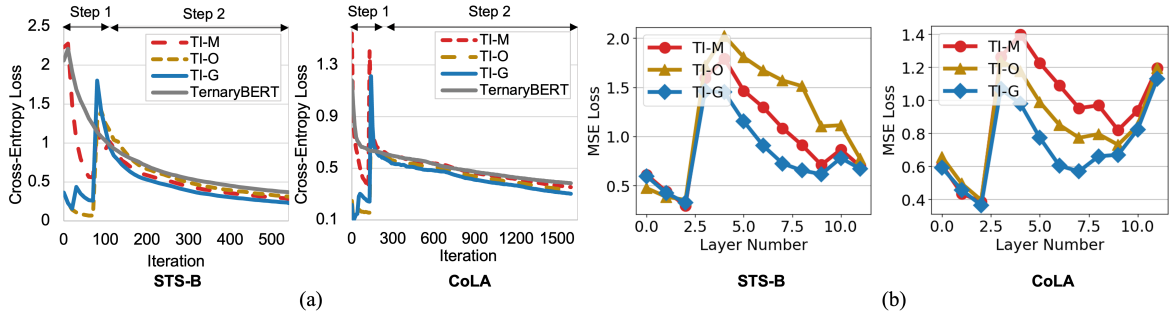


Figure 4: (a) Cross-entropy loss of QAT with different TI methods, and (b) Layer-wise MSE loss for different TI methods (Left: STS-B, Right: CoLA). Note that TI-G converges to lower loss than TI-M and TI-O on both tasks.

## 4.2 Gradual Teacher Intervention

Type	TI-M			TI-O			TI-G		
Sub-layer	GEN	PROP	FFN	GEN	PROP	FFN	GEN	PROP	FFN
Step1-Phase1	Q+TI	Q	Q	Q+TI	Q		Q+TI	Q	Q
Step1-Phase2							Q+TI	Q	Q
Step2		Q			Q			Q	

Table 1: Quantization settings for teacher intervention. Different from TI-M and TI-O, gradual teacher intervention (TI-G) applies TI to quantization (Q) from large (GEN+PROP) to small (GEN) scope in Step1.

Given multiple teacher intervention options, which would achieve the best performance? In this section, we propose a unified approach that gradually applies the output intervention followed by the map intervention. Note that the two TI options have strengths and weaknesses. For example, TI-O focuses on tuning the FFN sub-layers, but it lacks consideration of the self-attention map recovery. On the other hand, TI-M is best suited for recovering the self-attention map, but it does not protect signal propagation through SA-PROP. Interestingly, we empirically discover that different downstream tasks of the pre-trained Transformers have diverse preferences; e.g., BERT-base fine-tuned on STS-B is sensitive to disruption in the self-attention map while the model fine-tuned on CoLA prefers careful tuning of the FFN sub-layers. Therefore, developing a unified solution that utilizes teacher intervention in various scopes is beneficial. As a natural combination, we propose a gradual teacher intervention mechanism that applies TI-O first to tune the FFN sub-layers (Step1-Phase1), followed by TI-M to recover the self-attention map (Step1-Phase2). The proposed method, called *gradual intervention* (TI-G), has shown practical success in most ultra-low bit Transformers studied in this work. (We conducted an ablation study for the other possibilities of a unified solution for teacher intervention in A.3.) Table. 1 summarizes the TI

settings. Note that the two phases of TI-G in Step1 do not increase the total number of iterations thanks to fast convergence.

Fig. 4(a) shows the convergence curves of different TI options for STS-B and CoLA. As discussed earlier, TI-M shows better convergences than TI-O on STS-B, but the opposite trend is shown on CoLA. Nevertheless, TI-G always shows superior convergence compared to the other options, demonstrating its universal applicability. In particular, on STS-B, TI-G benefits from the output intervention in the first phase to favorably warm up the FFN sub-layers, and thus the map intervention in the next phase can reduce the loss more than TI-M. A similar situation happens in the case of CoLA.

Fig. 4(b) investigates the internal behavior of Transformers via the layer-wise MSE at the output of MHA. As discussed earlier, the fine-tuned Transformers exhibit distinct characteristics depending on the downstream tasks. For example, in the case of STS-B, TI-M is more effective in reducing MSE since the recovery of the self-attention map is essential. On the other hand, CoLA has preferred TI-O for its focus on tuning FFN sub-layers. In both cases, TI-G’s gradual intervention with decreasing scopes from the attention output to the self-attention map helps achieve the smallest MSE. Therefore, this investigation suggests that the proposed unified intervention mechanism can manage diverse characteristics of fine-tuned Transformers for various downstream tasks. We demonstrate the general applicability of TI-G in the next section.

## 5 Experiments

In this section, we evaluate TI for QAT of fine-tuned BERT and vision Transformer with ternary weight quantization. We demonstrate that TI significantly boosts the convergence of QAT to achieve higher model accuracy within shorter fine-tuning time compared to the state-of-the-art QAT methods

(TernaryBERT (Zhang et al., 2020b) and XTC (Wu et al., 2022)) on various Transformer tasks and datasets. More extensive evaluation results on a wide range of BERT models as well as the ablation studies on various aspects of TI can be found in Appendix A.3. Due to limited space, detailed experiment settings are also summarized in Appendix A.1.

Iterations	CoLA	RTE	MRPC	STS-B	Avg. Ratio
Budget-O	1,603	233	343	538	1
Budget-A	13,325	4,468	7,050	10,066	16
Budget-C	159,909	53,621	84,611	120,795	200

Table 2: Different budgets for fine-tuning iterations for GLUE tasks.

### 5.1 QAT Accuracy on Few-Sample BERT

First, we perform an extensive performance comparison of the proposed teacher intervention methods with the state-of-the-art QAT methods: TernaryBERT (Zhang et al., 2020b) and XTC (Wu et al., 2022). We evaluate TI on fine-tuned BERT<sub>base</sub> (12 layer), BERT<sub>large</sub> (24 layer) (Devlin et al., 2019), TinyBERT (Jiao et al., 2020) (6 layer), and SkipBERT (Wu et al., 2022) (6 layer) for GLUE tasks (Wang et al., 2018). We report the experimental results for the small dataset tasks (less than 10k of dataset size) here; the results on the other models and tasks are reported in Appendix A.2.

To investigate the convergence of these QAT methods, we consider the following fine-tuning budgets:

- Budget-O: The number of iterations reported in the original paper (Zhang et al., 2020b).
- Budget-O2:  $2\times$  iterations of Budget-O.
- Budget-O4:  $4\times$  iterations of Budget-O.
- Budget-A/C: prolonged fine-tuning budgets employed in XTC (Wu et al., 2022).

A summary of fine-tuning budgets is shown in Table 2. Note that the number of fine-tuning iterations of Budget-A is roughly  $16\times$  larger than Budget-O. In the following subsections, we categorize the experiments to two scenarios: few-sample fine-tuning (Scenario-1: Budget O/O2/O4) and prolonged fine-tuning (Scenario-2: Budget A/C).

**Scenario-1: Few-Sample Fine-tuning:** Table 3 summarizes the experimental results of few-sample

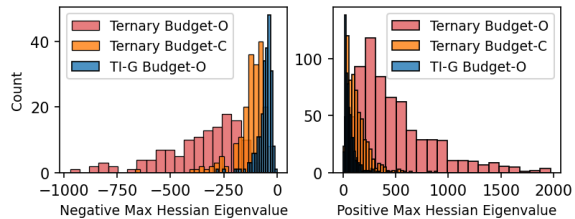


Figure 5: Convergence analysis of QAT methods with Hessian Max Eigenvalues.

fine-tuning. Consistent with the prior observations (Wu et al., 2022), the accuracy of TernaryBERT increased as the fine-tuning budget grew from Budget-O to Budget-O4. But all teacher intervention options have improved these baseline accuracies. Note that the preference between TI-M and TI-O varies across the tasks and models. For example, TI-M is mainly preferred on CoLA and RTE, while there is a marginally higher preference for TI-O on STS-B. For most cases, however, TI-G outperforms the other TI options. Although the accuracies of TI-M and TI-O are similar, TI-G’s accuracy significantly exceeds both. From these observations, we can conclude that the proposed gradual intervention significantly improves the QAT convergence for regaining model accuracy.

**Scenario-2: Prolonged Fine-tuning:** Table 4 summarizes the experimental results of prolonged fine-tuning. For fair comparisons, we followed the instructions of XTC to match the fine-tuning budgets, learning rates, and the model compression mechanism on both TinyBERT and SkipBERT. As expected, TI-G’s accuracy is significantly increased from Budget-O to Budget-A/C than Budget-O4. Interestingly, TI-G’s accuracy is higher than TernaryBERT and XTC for these prolonged fine-tuning budgets with noticeable margins. In fact, XTC’s average accuracy on Budget-A (for SkipBERT-6L) is lower than TI-G’s average accuracy on Budget-O4 (for TinyBERT-6L), highlighting the superior convergence of TI-G compared to XTC.

### 5.2 QAT Accuracy on Vision Transformers

We further evaluate the proposed teacher intervention method on ViT. Table 5 summarizes the QAT accuracies of ViT fine-tuned for CIFAR10, CIFAR100, and ImageNet with the fine-tuning budgets following the original paper (Dosovitskiy et al., 2020). As shown in the Table. 5, TI-G outperforms TernaryBERT on all the QAT cases for ternary weight quantization. It is noteworthy that TI-G is exceptionally effective for fine-tuned ImageNet.

QAT Type	Iterations	BERT <sub>base</sub>				BERT <sub>large</sub>			
		RTE	CoLA	STS-B	MRPC	RTE	CoLA	STS-B	MRPC
Full-Prec	Budget-O	73.28	58.04	89.24	87.77	70.39	60.31	89.83	88.43
TernaryBERT	Budget-O	67.44 $\pm 1.30$	49.44 $\pm 1.11$	87.58 $\pm 0.09$	85.58 $\pm 0.58$	63.36 $\pm 1.01$	53.25 $\pm 1.20$	88.65 $\pm 0.16$	88.31 $\pm 0.20$
TI-Map	Budget-O	69.60 $\pm 0.92$	51.37 $\pm 1.23$	87.75 $\pm 0.12$	86.25 $\pm 1.03$	66.13 $\pm 1.12$	52.40 $\pm 1.65$	88.61 $\pm 0.16$	88.67 $\pm 0.32$
TI-Output	Budget-O	69.31 $\pm 0.57$	50.91 $\pm 0.94$	87.76 $\pm 0.22$	86.04 $\pm 0.61$	65.20 $\pm 0.94$	52.66 $\pm 1.27$	88.56 $\pm 0.16$	88.68 $\pm 0.51$
TI-G	Budget-O	<b>70.32</b> $\pm 0.72$	<b>51.98</b> $\pm 1.35$	<b>87.77</b> $\pm 0.29$	<b>86.44</b> $\pm 0.49$	<b>66.27</b> $\pm 0.79$	<b>54.12</b> $\pm 1.13$	<b>88.66</b> $\pm 0.05$	<b>88.80</b> $\pm 0.41$
TernaryBERT	Budget-O2	70.51 $\pm 0.41$	52.65 $\pm 0.77$	88.04 $\pm 0.14$	86.00 $\pm 0.38$	66.42 $\pm 0.62$	55.72 $\pm 1.26$	89.00 $\pm 0.09$	88.22 $\pm 0.82$
TI-G	Budget-O2	<b>71.48</b> $\pm 0.36$	<b>54.98</b> $\pm 0.66$	<b>88.04</b> $\pm 0.18$	<b>88.63</b> $\pm 0.55$	<b>68.11</b> $\pm 0.75$	<b>57.55</b> $\pm 1.69$	<b>89.12</b> $\pm 0.04$	<b>88.25</b> $\pm 0.46$
TernaryBERT	Budget-O4	71.23 $\pm 0.42$	53.57 $\pm 0.80$	88.22 $\pm 0.04$	86.58 $\pm 0.32$	67.50 $\pm 0.95$	57.70 $\pm 0.64$	89.12 $\pm 0.01$	<b>89.12</b> $\pm 0.84$
TI-G	Budget-O4	<b>73.16</b> $\pm 0.36$	<b>57.92</b> $\pm 1.19$	<b>88.48</b> $\pm 0.61$	<b>89.56</b> $\pm 0.52$	<b>69.67</b> $\pm 0.72$	<b>59.89</b> $\pm 1.07$	<b>89.33</b> $\pm 0.10$	88.74 $\pm 0.76$
QAT Type	Iterations	TinyBERT-4L				TinyBERT-6L			
		RTE	CoLA	STS-B	MRPC	RTE	CoLA	STS-B	MRPC
Full-Prec	Budget-O	68.23	43.06	87.07	87.76	74.00	57.78	88.74	87.35
TernaryBERT	Budget-O	63.15 $\pm 0.50$	32.15 $\pm 1.43$	83.33 $\pm 0.36$	84.90 $\pm 0.38$	68.74 $\pm 1.42$	47.77 $\pm 0.35$	87.29 $\pm 0.12$	84.89 $\pm 0.53$
TI-G	Budget-O	<b>64.29</b> $\pm 0.72$	<b>35.17</b> $\pm 1.35$	<b>83.58</b> $\pm 0.29$	<b>85.48</b> $\pm 0.49$	<b>69.38</b> $\pm 0.78$	<b>49.08</b> $\pm 1.24$	<b>87.31</b> $\pm 0.11$	<b>86.30</b> $\pm 0.63$
TernaryBERT	Budget-O2	64.74 $\pm 0.76$	34.49 $\pm 1.89$	84.10 $\pm 0.34$	85.73 $\pm 0.05$	69.67 $\pm 0.95$	49.54 $\pm 0.22$	87.51 $\pm 0.04$	86.61 $\pm 0.41$
TI-G	Budget-O2	<b>64.98</b> $\pm 1.45$	<b>36.30</b> $\pm 0.24$	<b>84.27</b> $\pm 0.22$	<b>86.44</b> $\pm 0.31$	<b>71.12</b> $\pm 0.63$	<b>50.38</b> $\pm 0.56$	<b>87.56</b> $\pm 0.09$	<b>86.80</b> $\pm 0.42$
TernaryBERT	Budget-O4	65.10 $\pm 0.55$	35.91 $\pm 0.31$	84.36 $\pm 0.26$	86.70 $\pm 0.13$	70.75 $\pm 0.36$	51.66 $\pm 0.51$	87.75 $\pm 0.04$	86.76 $\pm 0.23$
TI-G	Budget-O4	<b>65.22</b> $\pm 0.55$	<b>37.40</b> $\pm 1.30$	<b>84.39</b> $\pm 0.27$	<b>87.22</b> $\pm 0.13$	<b>72.13</b> $\pm 0.12$	<b>52.08</b> $\pm 0.01$	<b>87.90</b> $\pm 0.14$	<b>87.15</b> $\pm 0.39$

Table 3: Accuracy comparison of QAT methods on BERT family (few-sample GLUE tasks) without and with TI for regular fine-tuning iterations (Budget-O/O2/O4). Each experiment is repeated 10 times.

QAT Type	Iterations	TinyBERT-6L				SkipBERT-6L				
		RTE	CoLA	STS-B	MRPC	RTE	CoLA	STS-B	MRPC	
Full-Prec	Budget-O	74.00	57.78	88.74	87.35	74.72	55.37	89.27	86.11	
TernaryBERT	Budget-A	72.02 $\pm 0.21$	53.44 $\pm 1.11$	88.43 $\pm 0.07$	88.14 $\pm 0.31$	XTC	69.91 $\pm 0.41$	53.74 $\pm 0.77$	88.77 $\pm 0.03$	86.29 $\pm 0.57$
TI-G	Budget-A	<b>72.92</b> $\pm 0.72$	<b>54.29</b> $\pm 1.35$	<b>88.45</b> $\pm 0.29$	<b>88.36</b> $\pm 0.49$	TI-G	<b>70.87</b> $\pm 0.20$	<b>56.46</b> $\pm 0.68$	<b>88.94</b> $\pm 0.04$	<b>86.98</b> $\pm 0.44$
TernaryBERT	Budget-C	73.40 $\pm 1.30$	54.11 $\pm 1.11$	<b>88.60</b> $\pm 0.02$	88.43 $\pm 0.58$	XTC	73.76 $\pm 0.54$	56.30 $\pm 0.67$	88.91 $\pm 0.03$	<b>87.38</b> $\pm 0.19$
TI-G	Budget-C	<b>73.82</b> $\pm 0.41$	<b>55.05</b> $\pm 1.13$	<b>88.60</b> $\pm 0.01$	<b>88.62</b> $\pm 0.02$	TI-G	<b>74.48</b> $\pm 0.79$	<b>56.32</b> $\pm 1.13$	<b>88.92</b> $\pm 0.03$	87.34 $\pm 0.41$

Table 4: Accuracy comparison of QAT methods on compressed BERT family (few-sample GLUE tasks) without and with TI for prolonged fine-tuning iterations (Budget-A/C). Each experiment is repeated 10 times.

Iterations	Short (1K)		Regular (10K/20K)	
	CIFAR100	CIFAR10	CIFAR100	ImageNet
Full-Prec	92.78	99.1	92.78	82.65
TernaryBERT	84.61 $\pm 0.12$	97.32 $\pm 0.02$	89.57 $\pm 0.04$	75.40 $\pm 0.12$
TI-G	<b>85.28</b> $\pm 0.04$	<b>97.59</b> $\pm 0.05$	<b>90.07</b> $\pm 0.04$	<b>76.66</b> $\pm 0.04$

Table 5: Accuracy comparison of QAT methods on ViT-B for vision benchmarks.

### 5.3 Convergence Analysis

In Sec. 4, we discussed that TI impedes propagation of quantization errors and flattens the loss surface for favorable convergence as shown in Fig. 2. To quantitatively analyze the convergence of QAT, we conducted the Hessian eigenvalue analysis (Park and Kim, 2022) on BERT<sub>base</sub> over CoLA task. As shown in Fig. 5, TernaryBERT suffers large magnitude (positive and negative) eigenvalues. The large magnitude Hessian eigenvalues indicate sharp loss surfaces, unfriendly for quantization (Shen et al., 2020; Dong et al., 2019). A prolonged fine-tuning

would suppress the large Hessian eigenvalues at the expense of significantly increased fine-tuning computations. In contrast, TI significantly reduces the magnitude with an order of magnitude smaller fine-tuning iteration. From this analysis, we can conclude that TI helps flatten the loss surface to improve the convergence of QAT.

### 5.4 Further Analysis of TI-G: Self-Attention Map Recovery

In this section, we provide an analysis of TI-M’s effectiveness in the use of TI-G. Since the self-attention map encodes the relational representation of symbols within a sequence, quantization on SA-GEN would disrupt this relational encoding. Therefore, TI on the self-attention map (TI-M) is crucial for maintaining relative order within the self-attention map. As shown in Sec. 4.2, TI-O lacks consideration of the self-attention map recovery, requiring gradual teacher intervention mechanism



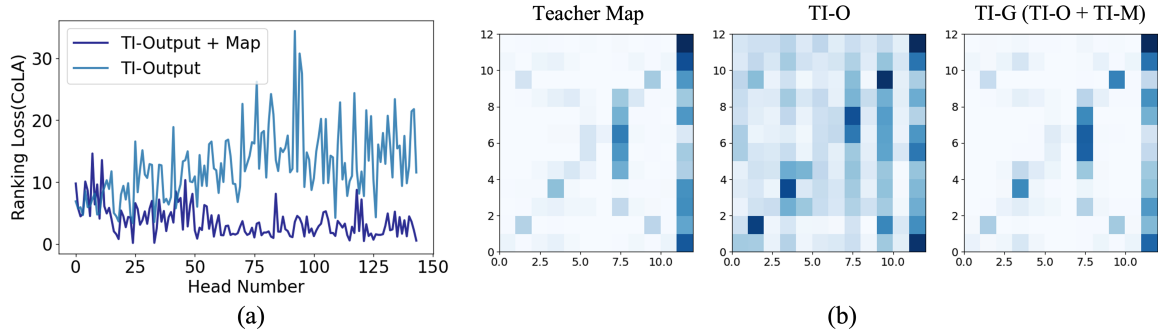


Figure 6: (a) Comparison of ranking loss between TI-O and TI-O with TI-M (TI-Output + Map) (b) Visualization of self-attention map with full-precision, TI-O and TI-G trained on CoLA. The input sentence is "John was lots more pleasure than Fred."

that applies TI-M after TI-O for recovering the self-attention map. Fig. 6 shows two empirical evidences that TI-M helps to recover the self-attention map with TI-O.

Fig. 6(a) shows the ranking loss computed on the self-attention map of  $BERT_{base}$  after QAT on CoLA. Note that the ranking loss quantitatively indicates how much the relative order in the quantized self-attention map has been skewed from the teacher’s self-attention map (Liu et al., 2021b). As can be seen, QAT without TI-M leads to significant disruption of the self-attention map. In contrast, QAT with TI-M maintains the original ranking of cross-symbol correlation, successfully suppressing the ranking loss. We further visualize each quantized model’s resulting self-attention maps in Fig. 6(b). Contrary to using TI-O only, TI-O with TI-M successfully recovers the teacher self-attention map’s distinctive feature. From this observation, we can conclude that TI-M approach used with TI-O is particularly beneficial for preserving the self-attention map under quantization.

## 6 Conclusion

In this work, we proposed a proactive knowledge distillation method for improving the convergence of QAT for ultra-low precision Transformers called teacher intervention. The proposed method intervenes in the propagation of quantization error to suppress accuracy degradation and improve QAT’s convergence speed. We demonstrate that the proposed method outperforms the state-of-the-art QAT methods in achieving higher accuracy on various fine-tuned Transformers with smaller fine-tuning budgets. Our code is available at <https://github.com/MarsJacobs/ti-kd-qat>.

## 7 Limitations

This work proposes a proactive knowledge distillation method called *teacher intervention* for improving the convergence of QAT with small fine-tuning budgets. Although our analysis reveals that quantization error propagation is one of the main causes of sub-optimal convergence of QAT, a more in-depth investigation of the detailed recovery mechanism of QAT from quantization error would be interesting. Also, extending the discovery of proactive knowledge distillation to diverse Transformer architectures, including encoder-decoder and decoder-only models, would be a promising future research direction.

## Acknowledgements

This work was partly supported by Samsung Advanced Institute of Technology, Samsung Electronics Co., Ltd, Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government (MSIT) (2020-0-01297, Development of Ultra-Low Power Deep Learning Processor Technology using Advanced Data Reuse for Edge Applications) and National Research Foundation of Korea (NRF) grant funded by Korea government (MSIT) (No. 2021R1A2C1013513).

## References

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. 2020. [Knowledge distillation from internal representations](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY*,

- USA, February 7-12, 2020, pages 7350–7357. AAAI Press.
- Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. **BinaryBERT: Pushing the limit of BERT quantization**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4334–4348, Online. Association for Computational Linguistics.
- Maximiliana Behnke and Kenneth Heafield. 2020.  **Losing heads in the lottery: Pruning transformer attention in neural machine translation**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2664–2674, Online. Association for Computational Linguistics.
- Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model. *arXiv preprint arXiv:1906.00532*.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.
- Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2019. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 293–302.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. **Compressing BERT: Studying the effects of weight pruning on transfer learning**. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 143–155, Online. Association for Computational Linguistics.
- Daniel Griebhaber, Johannes Maucher, and Ngoc Thang Vu. 2020. **Fine-tuning BERT for low-resource natural language understanding via active learning**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1158–1171, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. **Binarized neural networks**. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Kyuyeon Hwang and Wonyong Sung. 2014. **Fixed-point feedforward deep neural network design using weights +1, 0, and -1**. In *2014 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–6.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. **TinyBERT: Distilling BERT for natural language understanding**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.
- Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. *International Conference on Machine Learning (Accepted)*.
- Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. **Attention is not only a weight: Analyzing transformers with vector norms**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021a. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022.
- Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. 2021b. **Post-training quantization for vision transformer**. In *Advances in Neural Information Processing Systems*.
- Yihuan Mao, Yujing Wang, Chufan Wu, Chen Zhang, Yang Wang, Quanlu Zhang, Yaming Yang, Yunhai Tong, and Jing Bai. 2020. **LadaBERT: Lightweight**

- adaptation of BERT through hybrid model compression. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3225–3234, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.
- Namuk Park and Songkuk Kim. 2022. [How do vision transformers work?](#) In *International Conference on Learning Representations*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. [Q-bert: Hessian based ultra low precision quantization of bert](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8815–8821.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. [Patient knowledge distillation for BERT model compression](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.
- Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. [MobileBERT: a compact task-agnostic BERT for resource-limited devices](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.
- Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 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, Brussels, Belgium. Association for Computational Linguistics.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20*, Red Hook, NY, USA. Curran Associates Inc.
- Xiaoxia Wu, Zhewei Yao, Minjia Zhang, Conglong Li, and Yuxiong He. 2022. [Extreme compression for pre-trained transformers made simple and efficient](#). In *Advances in Neural Information Processing Systems*.
- Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1063–1077.
- Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, pages 36–39.
- Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020a. Revisiting few-sample bert fine-tuning. In *International Conference on Learning Representations*.
- Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020b. [TernaryBERT: Distillation-aware ultra-low bit BERT](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 509–521, Online. Association for Computational Linguistics.
- Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. 2016. Trained ternary quantization. *arXiv preprint arXiv:1612.01064*.

## A Appendix

In this section, we provide experimental setup and TI implementation in Sec. A.1. We also provide evaluation results of GLUE (Wang et al., 2018) large datasets (SST-2, QNLI, MNLI, QQP) in Sec. A.2 and ablation studies in Sec. A.3. Further analysis of TI-G’s effectiveness is provided in 5.4.

### A.1 Experimental Settings

**Models Description** We use task-specific, fine-tuned BERT family models to evaluate our method. Further information on each model is as follows:

1. **BERT<sub>base</sub>**  $L=12, d=768 N_H=12$ , Contains about 110M parameters.
2. **BERT<sub>large</sub>**  $L=24, d=1024 N_H=16$ , Contains about 340M parameters.
3. **TinyBERT-4L**  $L=4, d=312 N_H=12$ , Contains about 14M parameters. This compressed model is fine-tuned using task-specific distillation method (Jiao et al., 2020) initialized from general pre-trained TinyBERT model file.<sup>2</sup>
4. **TinyBERT-6L**  $L=6, d=768 N_H=12$ , Contains about 67M parameters. This compressed model is fine-tuned using task-specific distillation method (Jiao et al., 2020) initialized from general pre-trained TinyBERT model file.<sup>3</sup>
5. **SkipBERT-6L**  $L=6, d=768 N_H=12$ , Contains about 67M parameters. This compressed model parameters are initialized from XTC (Wu et al., 2022) model initialization method: using every other layer of the fine-tuned BERT<sub>base</sub>’s parameter to initialize the layer-reduced model.

**Datasets and Settings** We evaluate our method on GLUE benchmark (Wang et al., 2018), which is a collection of resources for training, evaluating, and analyzing natural language understanding systems. For the training setting, we use the batch size of 16 for CoLA and 32 for other tasks. The learning rate starts from zero and gradually increases to  $2e-5$  during the warm-up stage and decays linearly to  $2e-9$  for total training epochs.

<sup>2</sup>[https://huggingface.co/huawei-noah/TinyBERT\\_4L\\_zh](https://huggingface.co/huawei-noah/TinyBERT_4L_zh)

<sup>3</sup>[https://huggingface.co/huawei-noah/TinyBERT\\_6L\\_zh](https://huggingface.co/huawei-noah/TinyBERT_6L_zh)

	(Zhang et al., 2020b)			(Wu et al., 2022)	
	Budget-O	Budget-O2	Budget-O4	Budget-A	Budget-C
DA	✗	✗	✗	✓	✓
Epoch	3	6	12	1	12

Table 6: Training budgets for the GLUE tasks. DA is data augmentation.

	CoLA	RTE	MRPC	STS-B
Train (noDA)	8,551	2,490	3,668	5,749
Train (DA)	213,212	142,991	225,630	322,121
DA / noDA	24.9	57.4	61.5	56.0
Budget-O Time	259	97	72	223
Budget-A Time	2,149	1,862	1,484	4,160
Budget-C Time	25,792	22,342	17,813	49,915

Table 7: Training details for the GLUE tasks. DA is data augmentation.

**Comparison of Training Cost** We performed an extensive performance comparison in Sec. 5.1 along the training budget to investigate the convergence of QAT methods. Table. 6 shows the training setting of each budget with the Data Augmentation (DA) option, which is first proposed by (Zhang et al., 2020b) (Jiao et al., 2020) artificially expanding the training sample for prolonged fine-tuning, causing a blow-up in fine-tuning costs. To investigate how much the DA magnifies the QAT overhead, we compare the number of training samples of GLUE small datasets and the number of data-augmented training samples in Table. 7. Furthermore, we measured the total QAT time for each Budget over GLUE small datasets with the A6000 single GPU, indicating that prolonged fine-tuning with DA would become profoundly painful, increasing model deployment costs and time.

**Teacher Intervention Implementation** Our experiments were performed on A6000 GPUs. We use Pytorch 1.10.2 for the implementation of TI and this implementation is based on the TernaryBERT Pytorch code base.<sup>4</sup> We would like to emphasize that TI has strong advantages of simplicity for implementation. To implement TI operation, we modified the teacher model forward operation, enforcing the teacher model to return layer-wise MHA outputs and SA-GEN outputs which will be used in TI-O and TI-M respectively. When the teacher model’s layer-wise required outputs are returned,

<sup>4</sup><https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/TernaryBERT>



Method	Cost	BERT <sub>base</sub>				BERT <sub>large</sub>			
		SST-2	QNLI	MNLI	QQP	SST-2	QNLI	MNLI	QQP
	Full-Prec	93.57	91.32	84.59	89.34	93.57	92.29	86.49	89.55
TernaryBERT	Budget-O	91.82 ±0.21	90.68 ±1.11	83.74 ±0.07	89.09 ±0.31	91.62 ±0.41	91.91 ±0.77	85.52 ±0.03	89.26 ±0.57
TI-Gradual	Budget-O	<b>92.16</b> ±0.72	<b>90.91</b> ±1.35	<b>84.13</b> ±0.29	<b>89.11</b> ±0.49	<b>92.12</b> ±0.20	<b>92.11</b> ±0.68	<b>86.11</b> ±0.04	<b>89.46</b> ±0.44

Method	Cost	TinyBERT-4L				TinyBERT-6L			
		SST-2	QNLI	MNLI	QQP	SST-2	QNLI	MNLI	QQP
	Full-Prec	92.86	91.32	84.59	89.34	93.57	92.29	86.49	89.55
TernaryBERT	Budget-O	91.43 ±0.72	84.18 ±1.35	80.93 ±0.29	85.02 ±0.49	91.12 ±0.20	89.71 ±0.68	83.51 ±0.04	89.43 ±0.44
TI-Gradual	Budget-O	<b>91.45</b> ±0.72	<b>84.51</b> ±1.35	<b>81.06</b> ±0.29	<b>85.09</b> ±0.49	<b>91.43</b> ±0.20	<b>89.81</b> ±0.68	<b>83.64</b> ±0.04	<b>89.44</b> ±0.44

Table 8: Accuracy comparison of QAT methods on BERT family (large-sample GLUE tasks) without and with TI for regular fine-tuning iterations (Budget-O). Each experiment is repeated 3 times.

Method	Cost	BERT <sub>base</sub>				BERT <sub>large</sub>			
		RTE	CoLA	STS-B	MRPC	RTE	CoLA	STS-B	MRPC
	Full-Prec	73.28	58.04	89.24	87.77	70.39	60.31	89.83	88.43
TernaryBERT	Budget-A	73.16 ±0.54	54.81 ±0.39	88.92 ±0.02	<b>87.77</b> ±0.32	70.39 ±0.21	58.65 ±0.31	89.77 ±0.01	<b>88.91</b> ±0.59
TI-G	Budget-A	<b>74.24</b> ±1.36	<b>58.29</b> ±0.50	<b>89.01</b> ±0.01	87.74 ±0.96	<b>71.48</b> ±0.95	<b>59.24</b> ±1.38	<b>89.89</b> ±0.07	88.52 ±0.22
TernaryBERT	Budget-C	73.40 ±0.41	58.25 ±0.84	89.23 ±0.01	88.43 ±0.31	71.48 ±0.06	59.11 ±1.47	89.85 ±0.01	<b>89.08</b> ±0.55
TI-G	Budget-C	<b>75.93</b> ±0.90	<b>58.91</b> ±0.54	<b>89.39</b> ±0.16	<b>87.85</b> ±0.26	<b>72.68</b> ±1.50	<b>59.17</b> ±0.61	<b>90.10</b> ±0.01	88.74 ±0.41

Table 9: Accuracy comparison of QAT methods on BERT family (few-sample GLUE tasks) without and with TI for prolonged fine-tuning iterations (Budget-A/C). Each experiment is repeated 10 times.

	BERT <sub>base</sub>	BERT <sub>large</sub>	TinyBERT-4L	TinyBERT-6L
Baseline	72.51	73.29	65.88	72.17
TI - output loss	73.92	73.79	66.85	72.77
TI - two step	73.95	74.26	66.32	72.59
TI	<b>74.13</b>	<b>74.46</b>	<b>67.13</b>	<b>73.02</b>

Table 10: Ablation study of Teacher Intervention. Each result is averaged score over GLUE small datasets (RTE, CoLA, STS-B, MRPC)

we passed these to the student model forward operation’s input. With these additional inputs, we replace the student’s attention map/output with the teacher’s in the attention sub-layer forward operation. More detailed implementation can be found at <https://github.com/MarsJacobs/ti-kd-qat>.

## A.2 Additional Experimental Results

**Large-Sample Fine-tuning** Table. 8 shows evaluation of TI in GLUE large datasets (SST-2, QNLI, MNLI, QQP). TI-G outperforms the baseline for all the cases as we discussed in Table. 3, showing that TI improve convergence of QAT in large sample fine-tuning.

**Additional Results of Prolonged Fine-Tuning** Table. 9 shows additional evaluation results of TI in prolonged fine-tuning (BERT<sub>base</sub> and BERT<sub>large</sub> with Budget-A/C). As can be seen, TI-G shows

	BERT <sub>base</sub>	BERT <sub>large</sub>	TinyBERT-4L	TinyBERT-6L
Inverted TI	74.27	74.37	68.28	72.98
Stochastic TI	74.32	<b>74.57</b>	68.17	72.93
TI-G	<b>74.37</b>	74.41	<b>68.32</b>	<b>73.06</b>

Table 11: Comparison of unified TI strategies. Each result is averaged score over GLUE small datasets (RTE, CoLA, STS-B, MRPC)

superior accuracy to TernaryBERT with a noticeable margin in Budget-A/C both. Furthermore, TI-G achieves close to full-precision accuracy with Budget-A, showing TI’s beneficial effect on the convergence of QAT.

## A.3 Ablation Study

**TI two step QAT and output loss** In this section, we delve deeper into evaluating the efficacy of the TI method. As TI-O employs one additive loss for internal distillation (Attention Output MSE loss, called output loss) and TI QAT is performed in two steps, conducting an ablation study on TI (exploiting output loss and two-step QAT separately) provides a deeper understanding of our method. Table. 10 demonstrates the effect of output loss and the use of a two-step method with TI independently with BERT family models over GLUE small datasets. As can be seen, both factors act positively on the performance of quantized

models. Interestingly, two-step QAT factor provides a powerful impact (Row 2) in compressed models (TinyBERT-4L and TinyBERT-6L), and output loss shows a distinct impact (Row 3) in the BERT<sub>large</sub> model. When both factors are implied together (Row 4), performance is boosted further.

**Exploration of unified teacher intervention methods** In this section, we investigate the best unified teacher intervention strategy. We proposed a unified intervention approach (TI-G) in Sec. 4.2, applying TI with output intervention (TI-O) followed by map intervention (TI-M). As more scheduling choices are left when using TI-O and TI-M together, we propose additional two scheduling methods, Inverted TI, and Stochastic TI. Inverted TI applies TI-O and TI-M in the opposite order of TI-G, and stochastic TI means choosing TI-O and TI-M options randomly in every training iteration, ensembling each TI option’s effect in a stochastic way. Table. 11 shows the comparison between unified TI scheduling methods with BERT family models over GLUE small datasets. TI-G outperforms other unified TI scheduling methods in almost every model, which shows our proposed teacher intervention unification approach’s efficacy.